Operations Research Society of New Zealand

www.orsnz.org.nz

45th ANNUAL CONFERENCE

29-30 November 2010

University of Auckland
Auckland
New Zealand

Sponsored by

IBM

Fonterra

Derceto

Mighty River Power

Hoare Research Software Ltd

The Energy Centre, University of Auckland

Department of Engineering Science
University of Auckland

ISSN 0114-6602

The
Operations Research Society of New Zealand
gratefully acknowledges the
generous support of the
following conference
sponsors.

Principal Sponsor

Banquet Sponsor

Young Practitioner Prize Sponsor

Sponsor

THE UNIVERSITY OF AUCKLAND
BUSINESS SCHOOL
The Energy Centre
Sponsor

Hoare Research Software Ltd
Sponsor

THE UNIVERSITY OF AUCKLAND
FACULTY OF ENGINEERING
Department of Engineering Science
Sponsor

Operations Research Society of New Zealand (Inc.)
P.O. Box 6544, Wellesley Street, Auckland, New Zealand
http://www.orsnz.org.nz

Preface

The papers in this volume form the Proceedings of the 45th Annual Conference of the Operations Research Society of NZ (ORSNZ) held on 29 and 30 November 2010 at the University of Auckland, Auckland, New Zealand.

This conference could not have been organised without the invaluable assistance of staff from the Department of Engineering Science at the University of Auckland. The full conference committee is listed below. Thanks also to all the authors who contributed their papers.

We are very pleased to have distinguished plenary speakers in our programme. Professor Tava Olsen of the Business School, University of Auckland, has recently returned to New Zealand after nearly 20 years in the United States to take up the Ports of Auckland Chair in Logistics and Supply Chain Management. She is also the Academic Director of the New Zealand Centre for Supply Chain Management.

Professor Martin Savelsbergh is best known for his work at GeorgiaTech as an optimization and logistics specialist. He has over 20 years experience in operations research including optimization methods, algorithm design, performance analysis, logistics, supply chain management, and transportation systems. He has published over 100 research papers in many of the top optimization and logistics journals. Martin has recently moved to Australia where he is the Program Manager of Business and Services Analytics at the CSIRO and is a Conjoint Professor at the University of Newcastle.

We are most grateful for the support received from our Principal Sponsor, IBM ILOG, best known for their successful CPLEX optimization software. Derceto, developers of the Aquadapt software for real-time optimization of water distribution, are proud to sponsor the Young Practitioner Prize. We thank Fonterra for their generous contribution towards the banquet. We also acknowledge the support provided by Hoare Research Software Ltd (who also sponsor our newsletter), the Department of Engineering Science and the University of Auckland.

These proceedings are available in electronic form at http://conf45.orsnz.org.nz/

We hope you enjoy your time in Auckland, both at the conference and exploring Auckland's harbour and attractions.

Andrew Mason and Matthias Ehrgott
Proceedings Editors
November 2010

Table of Contents and Programme

Monday 29 November

08:45–10:30 Welcome and Young Practitioner Prize 1 (YPP1)
Session Chairs: Andrew Mason & Mike O'Sullivan
ClockTower 039

Dominic Fok
Maintenance Operation Centre Rostering Problem 1
Eduard Bulog
A Generic Nurse Rostering Algorithm for the INRC2010 Instances 8
Harriet Priddey† and Kane Harton
Comparing the Efficiency of Stores at New Zealand Post 18
Siamak Moradi
The Bi-Objective Multi-Commodity Minimum Cost Flow Problem 28
Anthony Downward
Risk Aversion and Retail Electricity Markets 39
Timothy Harton
Virtual Smelter Modelling for Metal Flow Management 50

11:00–12:30 Plenary Session
Session Chair: Andrew Mason
ClockTower 039

Tava Olsen
Demand Management: Effective, Evil, or Just Everyday? 60
Chun Zhi Wu
Achieve Smarter Decision Outcomes with Optimisation 61

14:00–15:30 Young Practitioner Prize 2 (YPP2)
Session Chair: Cameron Walker
ClockTower 039

Min Gih Choi
Collateralised Debt Obligation Portfolio Optimisation 62
Jason Undan
Optimisation of Demand-Side Bidding 72
Lawrence Li
Aircraft Route Guidance Through Convective Weather 82
Jason Drake
Design of Road Networks 90
Simon Bull† and Denis Helm†
Bi-objective Cycle Route Finding 100
Uttara Nataraj
Selecting a Portfolio of Cycling Projects 110

16:00–17:15 Young Practitioner Prize 3 (YPP3)
Session Chair: Golbon Zakeri
ClockTower 039

Sebastian Ruther
A Multi-commodity Flow Formulation for the Integrated Aircraft Routing, Crew Pairing, and Tail Assignment Problem *120*

Thomas Liddle
Kick Strength and Online Sampling for Iterated Local Search *130*

Albert Ang† and Michael McIvor†
Optimal Public-Transport Transfer Synchronization Using Operational Tactics *140*

Oddo Zhang
Optimisation of Small-Scale Ambulance Move-Up *150*

Kane Harton
Scale Advantage - Using Data Envelopment Analysis to Detect Economies of Scale in the Insurance Industry *160*

17:30 ORSNZ AGM and Council Meeting
All Welcome
ClockTower 039

19:30 Conference Banquet
Harbourside Restaurant: The Ferry Building, 99 Quay Street

† Presenting author

Tuesday 30 November

08:50–10:10 Parallel Session 1A
Session Chair: Alastair McNaughton
ClockTower 039

Geoff Leyland
Rima: Building Math Models for Reuse 170

Andrew Mason and Iain Dunning
OpenSolver -- Open Source Optimisation for Excel 181

Michael O'Sullivan,† Qi-Shan Lim, Cameron Walker and Stu Mitchell
Dippy -- A Simplified Interface to Advanced Integer Programming Techniques 191

Alastair McNaughton† and David Ryan
Constraint Branching Techniques 201

08:50–10:10 Parallel Session 1B
Session Chair: John Buchanan
ClockTower 029

Christopher Arasanmi
A User's Critical Characteristic (UCC) in ERP System Training Transfer: An HRD Perspective 209

Luisa Mariele Strauss and Denis Borenstein†
Analyzing the Brazilian Higher Education System Using System Dynamics 210

Robert Y. Cavana
Scenario Modelling for Managers: A System Dynamics Approach 219

Stuart Dillon, John Buchanan,† Jim Corner
Comparing Public and Private Sector Decision Making: Problem Structuring and Information Quality Issues 229

10:40–11:20 Parallel Session 2A
Session Chair: David Ryan
ClockTower 039

David Ryan
The Container Positioning Problem Revisited Yet Again! 238

Samsul Islam,† Tiru Arthanari and Tava Olsen
Empty Container-truck Movement Problem At Ports of Auckland 239

10:40–11:20 Parallel Session 2B
Session Chair: Matthias Ehrgott
ClockTower 029

Divya Jain† and K.K. Aggarwal
Optimal Ordering Policy With Inspection Errors and Learning Curve Consideration on Imperfect Quality Items 249

Sherry Hsu, Matthias Ehrgott† and Piaras Kelly
Optimisation of Mould Filling Parameters of the Compression Resin Transfer Moulding Process 259

11:20–12:30 Plenary Session
Session Chair: David Ryan
ClockTower 039

Mike Hewitt, George Nemhauser, Martin Savelsbergh† and Jin-Hwa Song
Branch-and-Price Guided Search *270*

Anver Sheha
Operations Research Opportunities at Fonterra.......... -

14:00–15:40 Parallel Session 3A
Session Chair: Hamish Waterer
ClockTower 039

Cameron Walker,† M.J. O'Sullivan, Y. Zhu, S. Hsu, M. L. MacKenzie, C. R. Donovan and V. Rajagopal
Regression Spline Fitting with Applications.......... *271*

Felipe Lillo† and Frederik Schmidt
Optimal Paths in Real Multimodal Transportation Networks: An Appraisal Using GIS Data from New Zealand and Europe *281*

Yuval Hadas† and Amir Laor
Network Design Model with Evacuation Constraints.......... *290*

Kenneth Kuhn† and Andrea Raith
Biobjective Air Traffic Flow Management *299*

Natashia Boland, Hamish Waterer† and Martin Savelsbergh
Shipping Stem Generation for the Hunter Valley Coal Chain *309*

14:00–15:40 Parallel Session 3B
Session Chair: Nicola Petty
ClockTower 029

Andrea Raith
Speed-up of Labelling Algorithms for Biobjective Shortest Path Problems.......... *313*

Yupei Xiong, Damon Gulczynski,† Bruce Golden, Daniel Kleitman and Edward Wasil
Worst-Case Analysis for the Split Delivery Vehicle Routing Problem with Minimum Delivery Amounts *323*

Laleh Haerian Ardekani, Tiru S. Arthanari† and Matthias Ehrgott
Performance of the Branch and Bound Algorithm on the Multistage Insertion Formulation of the Travelling Salesman Problem *326*

Shane Dye,† John Giffin, Nicola Petty
A Subset-Selection Prize-Collecting TSP with Uncertain Speed.......... *336*

Nicola Petty† and Shane Dye
Determining Degree of Difficulty in ROGO, a TSP-based Paper Puzzle *345*

† Presenting author

16:10–17:50 Parallel Session 4A
Session Chair: Mark Johnston
ClockTower 039

Yuval Hadas, Matan Shnaiderman† and Avishai (Avi) Ceder
Public-Transit Frequency Setting Using Minimum-Cost Approach with Stochastic Demand *353*

Matthew Baxter,† Mark Brown and Heng-Soon Gan
A Decision Support Tool for Equipment Replacement in Forestry Harvesting Operations *363*

Natashia Boland, Thomas Kalinowski,† Hamish Waterer and Lanbo Zheng
Maintenance Scheduling for the Hunter Valley Coal Chain *373*

Ross James† and Bernardo Almada-Lobo
Mathematical Programming-based Heuristics for Production Planning *376*

Mark Johnston† and Mike Wright
Scheduling Super Rugby *386*

16:10–17:30 Parallel Session 4B
Session Chair: John F Raffensperger
ClockTower 029

Guillermo Gallego and Masoud Talebian†
Demand Learning and Dynamic Pricing for Multi-Version Products *396*

Golbon Zakeri† and Anthony Downward
Financial Transmission Rights Auctions: Entry and Efficiency *400*

Stephen Starkey,† E.G. Read, S. Dye and B.J. Ring
A Natural Gas LP Formulation to Enhance Allocation with Market Pricing Mechanisms *409*

John F. Raffensperger
Outline of a Market for Ecological Connectivity *418*

† Presenting author

Maintenance Operation Centre Rostering Problem

Dominic Fok
Department of Engineering Science Part IV Project 2010
University of Auckland
In affiliation with Air New Zealand
New Zealand
cfok004@aucklanduni.ac.nz

Abstract

This report details the work done on the Maintenance Operations Centre (MOC) Rostering Solution Project. MOC looks after the short-term maintenance of Air New Zealand and other airlines. In order to provide sufficient service levels MOC has specific daily staff requirements. Each staff member has a set of skills that apply to specific aircraft systems. The objective of the project is to produce a method of generating high quality rosters for MOC that meet daily staffing requirements and maximize the skills available during any time period.

The current method MOC uses to generate their roster is by hand. The problem is characterized as a Rostering Problem. Information on how the objective and staffing requirements are placed into the Rostering Problem is given.

The modeling language used for this project is AMPL, which in conjunction with CPLEX is used to solve the problem. The solution rosters generated provide rosters that meet the daily staffing requirements and provide improved skill and service coverage. However, the rosters cannot be currently used in the operation of MOC, as they do not allocate the shifts fairly. It is expected that equal shift allocation will still allow high quality rosters to be generated.

There are also plenty of areas where additional improvements to the model can be made to improve solution quality. The most prominent improvement would be the addition of employee holiday inputs. This would allow the model to consider the effects of an employee going on leave and adjust solutions accordingly.

Once the equal allocation of shifts is achieved and employee vacation information is implemented, this model will be able to produce solutions that MOC can use. The solutions used will provide higher skill and shift coverage during MOC's operation.

Key words: MOC, skills, roster, shifts

1 Introduction

1.1 Background

MOC is operated by Air New Zealand. It provides short-term and response maintenance for Air New Zealand aircraft, as well as aircraft of other airlines which have contracts with Air New Zealand. An example where MOC would come into action would be: an inbound flight notices its wing has a slight malfunction. MOC would arrange for its repair before the aircraft would fly out again. MOC operates on a 24/7 basis to provide service.

MOC has three different staff tiers to facilitate its operation. The first tier is made up of work crews, who do the actual maintenance work required. The second tier is made up of sub-managers, who direct the work crews and provide aircraft specific technical expertise. The third tier consists of the managers, they ensure the daily operation of the MOC goes unhindered and also provide technical expertise when necessary. MOC has asked that only the manager and sub-manager tiers be investigated for this project.

1.2 Shifts

In order to provide adequate services, the daily operation of MOC has specific staffing requirements. To meet these requirements, MOC uses ten different shift types. In addition to the on-duty shifts available, a staff member can be placed on a call shift. While on a call shift, staff members are not required to be at the work place, but must be available for contact and able to work if asked to. On call employees are only called in when a currently or soon to be on-duty employee is unable to work. Managers and sub-managers work different shift types. Each shift type is required to have an employee assigned to that shift each day. The combination of shift allocations in a roster for an employee is called a work-line.

1.5 Skills

MOC services several different types of aircraft. Each of these aircraft is different and has its own technical systems. Because of this, employees need to be certified that they are able to work on a particular aircraft system. This certification means each employee has a skill set that allows them to work on specific aircraft systems. For example, if a Boeing 747 has mechanical issues, an employee with Boeing 747 Mechanical certification is needed to oversee its repair. Due to the operation requirements of MOC, an employee must be on location to provide a skill, this means employees on call shifts must be ignored when looking at the available skills for a time period.

1.6 Project Objective

The objective of this project is to produce a method to generate rosters for MOC employees that meet all daily shift requirements as well as any other requirements and maximize the skills available for each hour of the day.

2 Method

2.1 Restrictions

In generating a roster, there are rules on the shift allocations allowed for an individual. Some examples of these rules include; no more than 60 hours of work in a week, no more than 6 consecutive days with night shifts, shift allocation must obey Circadian Rhythm and others. The inclusion of these rules in roster generation reduces the number of variables in the problem. However, the inclusion of these rules alone is insufficient to limit the number of possibilities to a level where it can be solved. Thus, further steps are taken to reduce the size of the problem.

A set roster length of 27 days is introduced. Study of the current roster shows a prevalence of 6 days on-duty and 3 days off-duty for employees. Thus, the 27 days is made up of 3 9-day blocks. Each of these 9-day blocks may be chosen from a list of preset shift combinations. One example is 3 morning shifts, followed by 3 afternoon shifts and then 3days off. The exact composition of these 9-day blocks was chosen so that the number of 9-day blocks is as small as possible, while ensuring feasible solution can still be found. The composition of these 9-day blocks follows the rules specified above and not in this paper, as well as some introduced to reduce the number of combinations. As a result there are only 1500 possible choices for an employee when deciding their shift allocations over a 27 day period.

2.2 Cost of Skill Slack Variables

As these are the only variables with a coefficient in the objective function, the values for the cost of a lack of skill for a time period must be carefully scrutinized. The minimization of the presence of active slack variables is what drives the optimization problem. For the MOC problem, each skill corresponds to an aircraft. By looking at the fleet of aircraft Air New Zealand has, a weight can be associated with each skill, giving a cost. The air fleet compositions of airlines other than Air New Zealand has been ignored for this project, as Air New Zealand is the primary customer and there is insufficient information on other airlines. The skill weight is based on the number of aircraft that requires that skill.

2.4 History

As 27 days is an insufficient length of time for the roster it is necessary to include past information. By considering the last 6 days of a roster, this information can be used to generate constraints for the first 6 days of the next roster, so the connection of one roster to another is seamless. An implication of this is that a complete roster of any length can be generated. The reason only 6 days of data is required is that only 6 consecutive days need to be defined to choose a work line without breaking any rules.

2.5 Restrictions on Number of Shifts for an Employee

Because of the different skills each employee has, it is possible that an employee with less valuable skills may be repeatedly assigned call shifts and a employee with valuable skills on duty shifts. As well as being unequal in treatment, the annualization of hours means there must a method of ensuring employees do an equal amount of the different shift types. This is a requirement by MOC and must be added as a constraint. Currently, a limitation method is used to force equality. The limit is designated individually for each shift type and each employee. If an employee has more than their fair allocation of a shift, they will be limited in how many times that shift can appear during roster generation. This will force the shift to be allocated to other employees. The restriction method assigns each employee a value for each shift type.

The 27-day work line assignments may not assign shifts to employees that cause this value to be exceeded. The current method of deciding the restriction numbers is arbitrarily done by hand as an effective algorithm has not been investigated. In general, if an employee has had more than the average number of a particular shift type allocated to them, they will have a lower limit that will force them to take other shifts. After each 27 day step has been generated and the solution generated so far has been analyzed, the results of the analysis decide the restriction limits to be applied to the next 27 day step.

2.6 Formulation and Application

The MOC problem is formulated as a Rostering problem. The constraints have been expanded to include skills as well as staffing requirements. For each of these skill constraints a slack variable has been introduced, which will become active if that skill is not covered during that time period. Some of the constraints also have inequalities rather than equalities, this is because there is no penalty with the same skill being available multiple times during a time period. Other elements of the constraint vector may be larger than one, such as the requirement of two sub-managers on a night shift per day.

Positive costs are attached to the slack variables, so the solver will try to minimize the number of periods where a skill is not available. The cost for work lines is zero as this project currently ignores work line specific employee preferences. This is due to the annualization of hours, where employees receive an equal salary regardless of the exact number of shifts worked. To maintain equality of shifts the solver will try to assign shifts evenly to each person, this will be detailed later. An example of the MOC problem cost vector is shown in Figure 2.

Each column of the A matrix corresponds to either the skill and staffing requirement effects of an employee being assigned that roster, or to a slack variable for a skill during a specific day and time period. Where the roster would meet or help satisfy the staffing/skill requirement for that day and day period, the column has an entry of one. If it does not contribute to that day's staffing or skill requirement, the entry would be zero. The final part of the model is the binary variable vector x. The binary requirement is necessary, as multiples of the same employee is not possible. Furthermore, the constraints corresponding to the slack variables only require one instance to meet that constraint, so multiples of slack variables would not occur. Should a particular work line for an employee be chosen, the x variable will be 1 in the corresponding position and similarly for the slack variables. The combination of the c, b, x vectors and the A matrix allow the problem to be solved, and to be passed to a solver.

3 Results

Results for two different model states are to be shown. This is to compare how solution effectiveness changes according to whether or not there is fair allocation of shifts.

3.1 No Shift Restrictions

For this model state the slack variable weights are included, but not the shift restriction. This is to see the effect of adding weights to the skills. The objective function value is calculated by summing the coverage ratios for each skill over all periods and then multiplying that value by the skill weighting, these new values are then added together to get the objective function. A solution was obtained for the weighted roster problem. An analysis of the original roster and Solution Roster 1 was performed. The objective function for the original roster is 740.68, for Solution Roster 1 874.88.

3.2 Full Model

For this model state the full model is used by including shift restriction. This is to see the effect of restricting number of specific shift types an employee may do. The objective function value is calculated as before. A comparison is also made with rosters without the restriction constraints for both objective function and fairness of allocation.

A solution was obtained for the restricted roster problem. An analysis of the original roster, Solution Roster 1 and Solution Roster 2 was performed. The objective function for the original roster is 740.68, for Solution Roster 1 it is 874.88 and for Solution Roster 2 it is 869.58. A comparison of their average standard deviations and the average difference of the maximum and mean for all shift type allocations can be found in Figure 1.

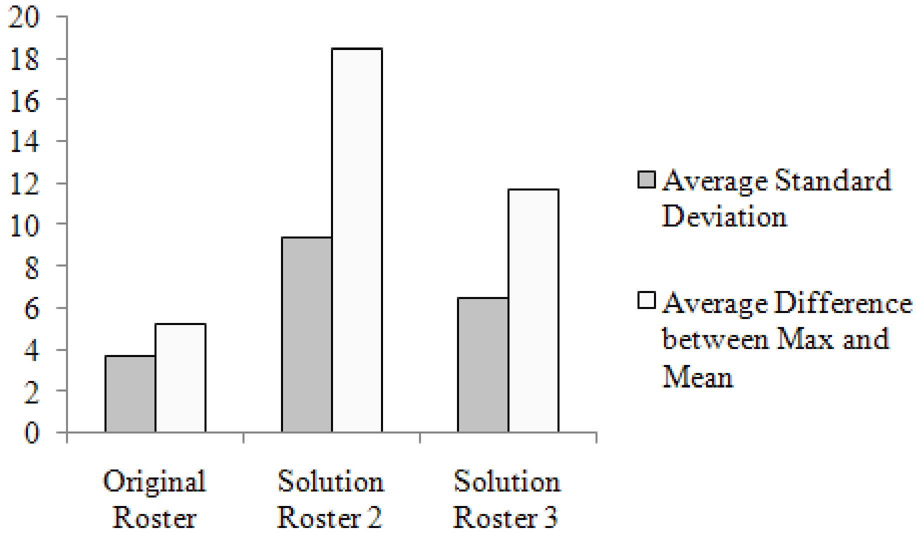


Figure 1 Comparison of Roster Occurrence Statistics

4 Discussion

It is clear that the use of a Rostering Problem optimization produces solutions that have a clear advantage over the original roster for the maximization of skills in a work period. However, the solutions produced so far have issues with the equality of shift assignments although, the shift restriction area of this problem has not been investigated in depth. It is expected that a solution that provides equality as well as high skill coverage is will be found. The issue of shift allocation equality needs to be addressed before the generation of solutions can be used in operation of MOC.

Another advantage of the solutions, is that one of the call shifts that was rarely filled in the original roster, is available single and even twice daily in the solution rosters. This extra coverage allows MOC some flexibility in dealing with the call shift, as they have an option allow a staff member to instead use the extra call shift when two people are assigned on the same day as an off day.

5 Limitations and Future Work

The current formulation has several limitations, one such is the lack of employee input. Employees may prefer have a say as to when they assigned certain shifts. Currently, it is possible to address this by explicitly adding employee requests as part of the history constraints, although if too many employees have requests this can result in an infeasible problem.

The formulation also ignores the length of each time period. This can be introduced such that each time period has a weighting equal to its length of time. Alternatively, data about the aircraft, or the peak arrival/departure times could be placed in the model to give the time periods weights.

The model currently does not have a method to consider aircraft flight information, such data could allow the model to prioritize skill coverage to certain time periods.

The optimization does not consider the possibility that MOC may prefer a skill be available at least twice in a time period. However, it is reasonably simple to adjust the constraint vector b to allow for such a requirement.

The model takes not take into account that employees may wish to take leave or require training. If training was only for a block of three days, it would be possible to force the solution generated to give that employee a day block where there is already another, but such a method would not be sufficient for employees going on three weeks holiday.

Currently as well as the problem of equalization of shifts, the issue of weekend shifts is also ignored. This should also be investigated and put into the model.

As well as the model limitations there are also the limitations of computing power and memory. During the first programming stages for the model, it was found that the problem could become too large to solve, due to the presence of numerous variables. The introduction of history allows the generation of smaller time steps, but requires human oversight to produce a complete solution. The production of solutions can also take a considerable amount of time. The time taken to produce an initial solution tends to be quite high, but also variable. Solutions have been known to take from as little as one minute to over twelve hours.

6 Conclusion

The application of the Rostering Problem to the operation of MOC can provide improved skill coverage and more shift coverage that the original roster. However, it is currently unable to produce satisfactory solutions due to a poor method of shift restriction to equalize shift allocations. Investigations into an appropriate algorithm for determine the restriction limits needs to be done, but it is expected a good equalizing algorithm will still produce improved solutions.

In addition to the issue of equalization, the model needs to allow for employees who wish to go on holiday. There is no current method in the model to take in such information. A method must be found that can respond to employees going on vacation before this can be replace the current method that is used by MOC.

The model of the problem has a lot of potential to provide an improved roster to MOC. If sufficient data is given, the model can be modified to take this information and use it to generate a solution that allows for the new data set. This could be in the form of known arrival/departure times, the aircraft that are used for the arrivals and flights, or otherwise.

Acknowledgements

I would like to thank my supervisors Professor David Ryan and Doctor Andrew Mason for the guidance they have provided me for this project. David was a huge help and inspiration, his knowledge and experience of the Rostering Problem was invaluable. Andrew also provided a lot of guidance during the length of the project and his own knowledge of rosters was also a great help. I would also like to thank Paul Keating, the

Air New Zealand liaison for helping me in this project, his enthusiasm for Operations Research was inspiring and was very helpful in obtaining useful data. I also thank Doctor Michael O'Sullivan Jr for the advice he gave me on this project with regard to CPLEX memory issues. I would also thank Carl Ho, whose previous project on rosters gave me some starting steps.

References

Ryan, D. 2010. 762 Set Partitioning Problem Notes. University of Auckland.

A Generic Nurse Rostering Algorithm for the INRC2010 Instances

E. Bulog
Department of Engineering Science
University of Auckland
New Zealand
ebul014@aucklanduni.ac.nz

Abstract

The issue of generating automated staff rosters has received a great deal of academic interest of late, and has also led to several timetabling competitions, most notably the First International Nurse Rostering Competition (INRC2010) which was held earlier this year (Haspeslagh et al. 2010).

Building on the work of several staff members and students in the Engineering Science Department (Dohn, Mason, and Ryan 2010), (Engineer 2003), (Mason and Smith 1998), this paper will present a column-generation approach to solving nurse rostering problems. The capabilities of this rostering engine have been expanded to allow it to be tested and benchmarked using the instances made available in the INRC2010.

One of the main issues which has traditionally plagued rostering problems is the lack of generality. Consequentially, customising a rostering engine to suit a specific instance can take a significant amount of time and effort. In our column generation approach, however, the use of pre-processor macro expansions allows us to easily customise the framework to suit a particular problem instance. Significant effort has been made to further automate this process for problem instances of the format given by the INRC2010.

Due to the inherently complicated nature of these problems, large instances can take a very long time to solve. In column-generation approaches, the bulk of this time is spent in thesub-problem, while the master-problem is comparatively quick. Consequentially, it will be the focus of future work to embed heuristic methods into the column-generation sub-problem in order to reduce the overall runtimes.

Key words: Rostering, column generation, pre-processing.

1 Introduction

1.1 Staff Rostering

In many large organisations, staff costs account for a very significant portion of all expenses, and so generating efficient staff rosters is of extreme importance from a financial perspective. However, rosters cannot be generated purely based on a financial basis. Staff preference is also a very important factor in need of consideration, as a roster which is financially efficient may lead to poor staff moral if, for example, it contains a succession of shifts which may be undesirable from a staff perspective.

This results in two conflicting objectives; minimising staff costs and minimising staff unhappiness. Traditionally, this has been approached by either representing staff preferences as hard-constraints, referred to as *rules*, or as soft-constraints which can have varying weights or *costs* assigned to them which reflect their relative importance (Burke et al. 2005), (Burke et al. 2001). If a roster violates such a soft-constraint, the total cost of the roster will be increased to reflect the violation.

Instances of such rostering problems can differ vastly between any two organisations due to specific regulations and other requirements. Traditionally, this has meant that a significant amount of time and effort is required to tailor the solution approach to each instance.

1.2 Column Generation

The Column Generation approach (Barnhart et al. 1994) provides a useful tool for solving Integer Problems with a large number of columns. The problem is broken down into a sub-problem and a master-problem. The sub-problem, or *pricing problem* generates a limited number of columns, with attractive reduced costs, and adds them to a *column pool*, containing a relatively small subset of all possible columns. The restricted master-problem then performs the actual optimisation using only columns from this column pool, saving us the time and computation effort involved with generating all remaining columns.

In the context of the nurse rostering framework, a column of the constraint matrix corresponds to a potential *roster-line* which a nurse could work. A roster-line consists of a sequence of shifts occurring at intervals over the rostering horizon which a particular staff member must work. If complete enumeration is used to produce these roster-lines, the number of columns will increase dramatically with the size of the problem, making the nurse rostering problem a perfect candidate for the column generation framework.

Because we elect to construct columns through an iterative process, as detailed in section 3, we can check for dominance at all stages, meaning we can eliminate some columns at early stages during the generation process, further increasing the efficiency of this algorithm.

In order to implement this column generation, code has been built around the Coin-OR Branch-Cut-Price, or *BCP* framework, forming what will henceforth be referred to as *BCP-nurse*. BCP is an open source framework for solving mixed integer programs using the branch, cut and price algorithms. The bulk of the additional code in BCP-nurse has been added to the pricing problem, or column generation phase.

1.3 Boost C++ Pre-Processor Library

The Boost C++ Libraries are a set of free libraries which contain many useful extensions to the existing C++ functionality. The existing modelling framework makes use of the Boost pre-processor library to express entities and their attributes using macro, or `#define`, expansions provided in the *user.hpp* C++ header file.

This means that the problem specific customisation can be performed as the code is compiled, rather than incurring overheads during runtime. For more detail, please refer to Mason and Ryan (2009).

2 Model Formulation

Indices

i = employee: $\mathbb{S} = \{0, 1, 2, ..., s\}$; j = demand: $\mathbb{D} = \{0, 1, 2, ..., d\}$;
k = roster-line: $\mathbb{R}'_i = \{0, 1, 2, ..., r\}$;

Parameters

c_{ik} = cost associated with assigning employee i to roster-line k;
b_j = staffing level required by demand j

$$a_{ijk} = \begin{cases} 1, & \text{if employee } i\text{'s roster-line } k \text{ contributes to demand } j \\ 0, & \text{otherwise} \end{cases}$$

Decision variables

$$x_{ik} = \begin{cases} 1, & \text{if employee } i \text{ works roster-line } k \\ 0, & \text{otherwise} \end{cases}$$

Model

$$\text{Minimize} \quad \sum_{i \in \mathbb{S}} \sum_{k \in \mathbb{R}'_i} c_{ik} x_{ik}$$

$$\sum_{k \in \mathbb{R}'_i} x_{ik} = 1 \qquad \text{for } i \in \mathbb{S} \tag{1}$$

$$\sum_{i \in \mathbb{S}} \sum_{k \in \mathbb{R}'_i} a_{ijk} x_{ik} \geq b_j \qquad \text{for } j \in \mathbb{D} \tag{2}$$

$$x_{ik} \in \{0, 1\} \qquad \text{for } i \in \mathbb{S},\ k \in \mathbb{R}'_i \tag{3}$$

Explanation

We are provided with a set $\mathbb{S}$ of s employees each with different skills, and a set $\mathbb{D}$ of d demands to be met. The objective is to minimise total cost of all roster-line assignments. Constraint (1) requires each staff member to be assigned to exactly one roster-line. Constraint (2) ensures minimum staffing level is satisfied for each demand specified. Note that $\mathbb{R}'_i$ is the set of roster-lines returned by the column generator for employee i, such that $\mathbb{R}'_i \subset \mathbb{R}_i$, where $\mathbb{R}_i$ is the set of all roster-lines for employee i.

3 The Column Generation Framework

In our formulation, a column of the constraint matrix corresponds to a potential roster-line which an employee could work. The aim of the column generation phase is to produce a small subset of all possible columns which may to be useful to the master problem. In order to achieve this, we construct a number of roster-lines for each staff member and then return a few of these with the most negative reduced costs, if any exist.

We do not simply enumerate all roster-lines during this stage. Instead, our column generation approach is based around the concepts of *entities* and *attributes*. Entities can be thought of as the building blocks for the roster-lines. The most fundamental entity is called a *shift*, which is simply a period of time in the rostering period during which a nurse may work in order to satisfy some demand. There are four other entity types; an *off-stretch* is a rest period between shifts, an *on-stretch* is a sequence of shifts falling on consecutive days, a *work-stretch* is a pairing of an on-stretch and an off-stretch, and the fifth entity is the roster-line itself. The process of constructing a roster-line using the other entities is outlined in section 3.1.

Attributes summarise the key characteristics of each entity. An attribute may be anything from the number of hours worked during a shift entity, to the number of consecutive weekends worked during a roster-line entity. Attributes may be used to ensure the feasibility of certain entities during the construction phases outlined in the rest of this section. For example, if we have a problem instance where the maximum number of consecutive working weekends is one, a roster-line generated with three consecutive weekends worked will be discarded as it is infeasible. Attributes can also be used to apply costs to the generated roster-lines, which is particularly useful when dealing with instances from the INRC2010, where everything must be modelled as a soft constraint. In the example given, our roster-line with three consecutive working weekends would incur a cost of two, one for each unit of violation.

3.1 Entity Construction

All shift and off-stretch entities are enumerated in the input files, as outlined in section 4.2, so are treated as inputs to this phase. The remaining entities are constructed using the process of either *initialisation* or *accumulation*. Initialisation is when a particular entity is first created, whereas accumulation creates a new entity by chronologically building off an existing entity of the same type.

For example, an on-stretch entity is initialised from a single shift entity. The resulting on-stretch, containing only a single shift, can then be combined with other shift entities using accumulation to produce a number of new on-stretch entities containing two shifts. In this example of accumulation, the first on-stretch containing only one shift is termed the *parent* entity, while the new on-stretches containing two shifts are referred to as *child* entities. The accumulation process can then continue by treating the on-stretches containing two shifts as parent entities, thus producing a number of child on-stretch entities which each contain three shifts.

Table 1 outlines the rules behind these processes; note that work-stretches cannot be accumulated, as they can only ever contain exactly one on-stretch and exactly one off-stretch. The combination of initialisation and accumulation allows for efficient generation of a large number of potentially useful roster-lines, which are the end product of the column generation process.

Table 1: Entity Initialisation and Accumulation

Entity	Initialisation	Accumulation
on-stretch	shift	on-stretch + shift
work-stretch	on-stretch + off-stretch	N/A
roster-line	work-stretch	roster-line + work-stretch

Table 1 also illustrates the three main phases of column generation; *on-stretch generation*, *work-stretch generation* and finally *roster-line generation*, in which on-stretches are generated from shifts, work-stretches are generated from on-stretches and off-stretches, and roster-lines are generated from work-stretches respectively. This relationships between entities are illustrated in figure 1.

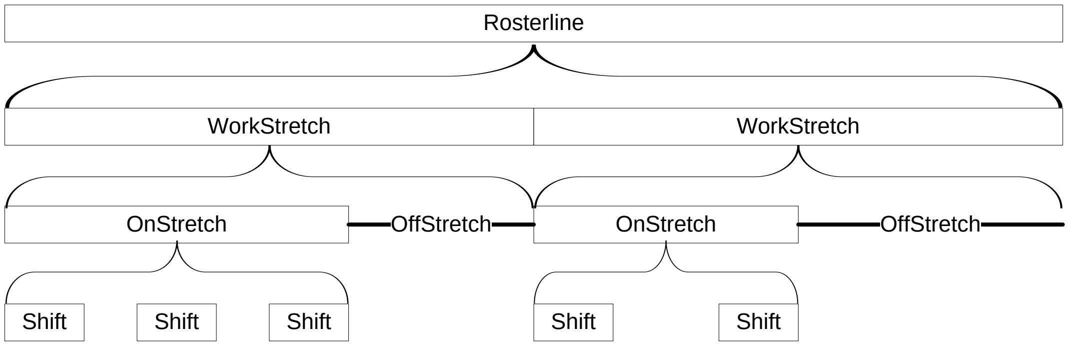


Figure 1: The Relationships between the Five Rostering Entities

Attributes for each entity are also calculated during all three phases following specific rules which are defined for the initialisation and accumulation phases. Some attributes follow simple additive rules, whereas others are more complicated. Examples of such calculation rules are provided in section 4.3. These attributes are also used to check for entity dominance as outlined in section 3.1.4.

3.1.1 On-Stretch Generation

An on-stretch can start with any shift and end with any later shift which satisfies some feasibility criteria; we may, for example, choose to limit our generation to on-stretches spanning a maximum of eight days. As such, we can construct a network whereby shifts represent nodes, and arcs represent feasible transitions between shifts. The nodes can be ordered chronologically, and there will be no arcs between shifts occurring on the same day, as we limit nurses to working one shift per day. Also, an arc will not exist between shifts occurring with a gap of a day or more between them, as an on-stretch, by definition, must consist of a sequence of shifts occurring on consecutive days. A small example of such a network is provided in figure 2.

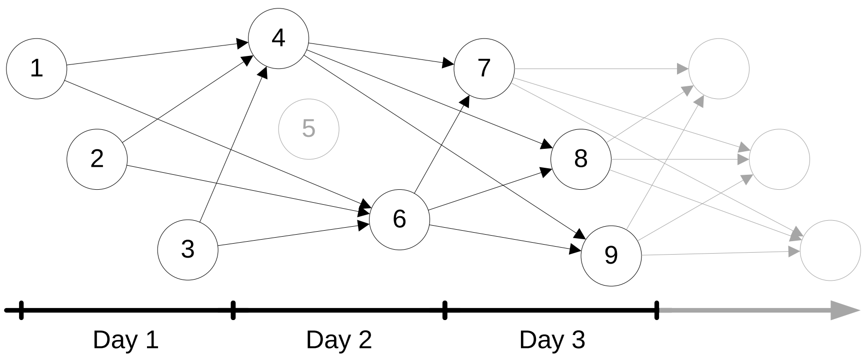


Figure 2: A Simplified Example of an On-Stretch Graph

This network also has resources which represent the attribute values. Generating an on-stretch beginning with shift i and ending with shift j corresponds to a path from node i to node j through the network. Such an on-stretch will only be feasible if the associated attributes are within the feasible range, so we can generate all feasible on-stretches by solving a one-to-all shortest path problem starting from each node in our network.

We can also use this network to generate only on-stretches which do not include a certain shift simply excluding the corresponding node. For example, node five has been excluded from the example in figure 2 meaning that we will only generate on-stretches (and therefore roster-lines) which do not include shift five. We may do this if an employee cannot work the given shift, or in order to enforce a branching decision made by the master problem, as the master problem branches on employee-shift assignments.

3.1.2 Work-Stretch Generation

As illustrated in table 1, work-stretches are generated directly from the pairing an on-stretch with an off-stretch, with no accumulation mechanism. Consequentially, the work-stretch generation phase is relatively simple. We examine each pair and construct the corresponding work-stretch if the time which elapses between the end of the on-stretch and the start of the off-stretch is within certain bounds.

3.1.3 Roster-Line Generation

In order to generate roster-lines, we need to combine a number of work-stretches together to span the entire rostering horizon. To this end, we can create another network where the work-stretches correspond to arcs forming transitions between days, or nodes. The first and last days of the rostering horizon correspond respectively to source and sink nodes in the network. Again, we must consider attributes as resources in the network, and so we can solve a shortest path problem with resource constraints. A feasible path through the network corresponds to a feasible roster-line.

3.1.4 Entity Dominance

The concept of entity dominance is used in all three stages of our column generation algorithm. At each stage, we compare the entities to see if any are dominated, in which case they are discarded, preventing further computational effort in generating extensions from them. An entity dominates another entity of the same type if it's cost is less than or equal to that of the dominated entity, and the cost of any future extensions will remain no more than the cost of the extensions of the dominated entity. All extensions of the dominated entity must remain feasible for the dominating entity.

This can be quite a complicated process, as each entity may have a number of attributes which behave in differing ways. Attributes of the entities are compared and dominance is established based on the methods specified by the instance-specific user.hpp C++ header file, which is generated by the Input File Generator outlined in section 4.2. There are several methods available, the most simple of which is only allowing dominance if all attribute values are equal, and the dominating entity has a cost which is less than or equal to the cost of the dominated entity. This particular case will always hold, but there are some situations where we can exploit the structure of the attribute to relax the equality requirement, resulting in improved efficiency.

4 Adapting the Framework to the INRC 2010

4.1 Code Synthesis XSD

The instances for the INRC2010 were provided in xml format, along with an xsd schema file which outlines the data format. The Code Synthesis XSD libraries were employed to read in the data. These libraries firstly create header files which implement the elements given in the schema file as C++ classes which have all appropriate members and functions to retrieve data. Once this framework has been set up based on the schema file, any xml instance file from the competition can be read in and the appropriate data stored in these classes. This approach allows for all instance data to be read in efficiently and easily accessed through code, since the format provided by the schema file is used by all instance files provided by the INRC2010.

4.2 The Input File Generator

C++ code was written around the XSD libraries to create a project referred to as the *Input File Generator*. This code is passed an xml instance file as an input and generates all required files for the nurse rostering software itself, or *BCP Nurse*. These include four text files which provide information about the available staff, all possible shifts, demands to be met and all possible off-stretches. A parameter file is also generated which is customised by the Input File Generator. The most complicated file generated here is the user.hpp C++ header file, which includes the attributes, rules and costs (among other things) which are specific to each problem instance, as mentioned in section 1.3. This file, containing Boost macro expansions, is required to be pre-processed and compiled into the BCP Nurse framework before the other five files are read in and the optimisation is solved. Figure 3 illustrates this process.

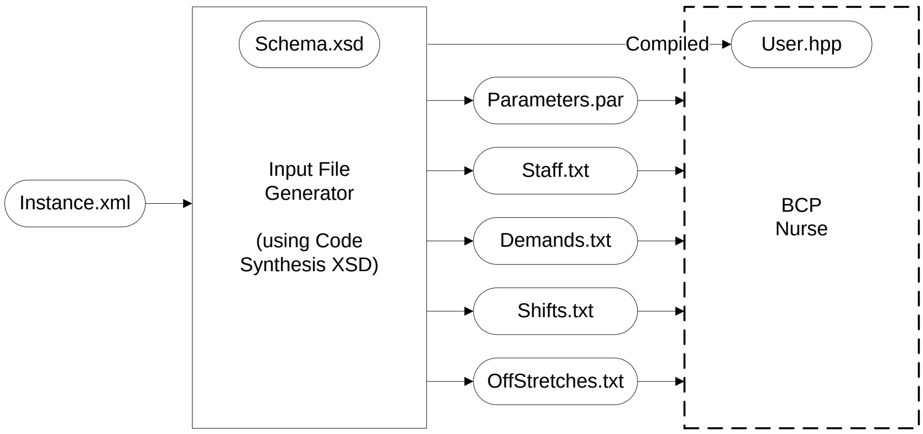


Figure 3: The Modified Setup Process

4.3 Additional Attributes

In order to improve the flexibility of the nurse rostering engine, several attributes were added to the various entities. These attributes allowed the model to penalise a number of additional unwanted features, such as the number of consecutive working weekends or patterns of shifts which staff may not want to work, as detailed in section 5. The attributes for all entities were all implemented in the user.hpp C++ header file, which is customised for the particular instance by the Input File Generator described in section 4.2. This customisation includes specifying the pricing scheme for each attribute, and any bounds on feasibility.

5 Applying Costs to Unwanted Shift Patterns

As mentioned, staffing preferences are a very significant aspect of nurse rostering. For a roster to be effective, it must not only be efficient in terms of cost and other concerns of the hospital, but it should also cater to the needs of the staff. This is a much harder aspect to quantify than monetary costs. As such, a key change which has been made to the rostering engine allows the model to handle sequences or patterns of shifts which nurses may not want to work.

These unwanted shift patterns may take several forms, but are characterised by a sequence of on-stretches and/or off-stretches over a number of consecutive days. Most of these patterns can be described by a single on-stretch; a common example of a two-day unwanted pattern of shifts might be a night shift on one day followed by an early shift on the second day, as illustrated in Figure 4. Nurses would not want to work a roster-line containing one (or many) of these patterns because this sequence of shifts does not allow for much rest in between. This example also illustrates why it may also be in the best interests of the hospital to avoid such patterns occurring, as patterns such as this can cause fatigue and/or demoralise staff.

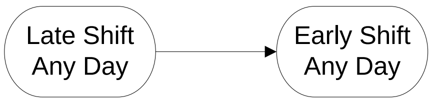


Figure 4: A Two-Entry Shift-Based Pattern

Unwanted shift patterns might also be defined not in terms of specific shifts worked, but simply by a sequence of days which are said to either be *on* or *off*. A day is termed an on-day if the employee of interest works a shift during that day, otherwise it is termed an off-day. An example of this type of unwanted pattern could be an off-day on a Friday followed by two on-days (Saturday and Sunday) as illustrated in Figure 5. This pattern essentially reflects that nurses do not want to work a full weekend if they are not working on the preceding Friday. Unwanted shift patterns such as this require both an on-stretch and an off-stretch in order to be fully described.

We require a general definition of an unwanted shift pattern which can include both on-stretches and off-stretches. To this end, each pattern is broken up into a number of *parts*. Each part contains a sequence of one or more consecutive shifts or days which can be expressed by either a single on-stretch or a single off-stretch. As such, a part can be referred to as either an on-stretch part or an off-stretch part.

Figure 5: A Three-Entry Day-Based Pattern

The individual pattern entries (shifts or days on/off) which make up each part are referred to as the *entries* of that part.

For example, if a particular rostering scenario included the two unwanted patters provided in Figures 4 and 5, then these two patterns would be composed of the parts and entities as outlined in Figure 6. Pattern 0 is consists of a single on-stretch part containing two entries. Pattern 1 consists of an off-stretch part comprised of a single entry and an on-stretch part containing two entries. For convenience, all numbering schemes start from zero.

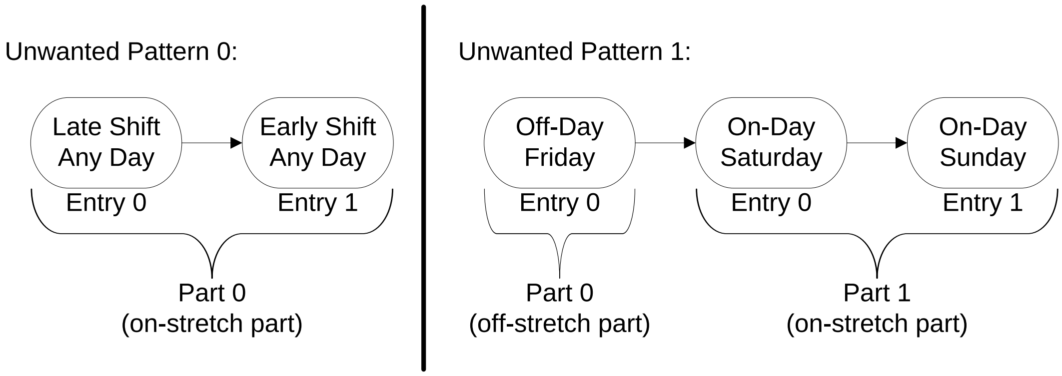


Figure 6: The Parts and Entries of Two Patterns

Attributes can be used to track the appearance of these unwanted shift patterns within each roster-line. Each part must be monitored separately within its respective entity, be it an off-stretch or on-stretch. This is further complicated by the fact that off-stretch attributes cannot be calculated (they can only be read as input), so the calculations for all attributes monitoring an off-stretch part must actually be performed in the work-stretch containing the off-stretch.

Each part of a pattern requires two attributes; a *tracker* and a *counter*. For convenience, each tracker and counter attribute are labelled with two numbers. The first of these numbers references the pattern which they correspond to, and the second number refers to the part within that pattern. For example, `pTracker1_2` denotes the pattern tracker attribute for part 2 of pattern 1 and `pCounter0_1` is used to denote the pattern counter attribute for part 1 of pattern 0.

The tracker attribute is used to monitor the construction of each part within the pattern. The value of this attribute is calculated based on how many consecutive entries have been added to the on-stretch or off-stretch which potentially belong to the part. Once the value of a pattern counter attribute has reached a maximum value (determined by the total number of entries in the part) the corresponding pattern tracker attribute is incremented. The pattern counter attribute simply counts how many complete parts are present in the given on-stretch or off-stretch.

Unwanted patterns can have different weights or costs assigned to them. These costs can be applied directly to the corresponding counter attributes, meaning that each occurrence of a pattern within a potential roster-line is penalised accordingly, usually using a simple linear relationship.

6 Future Work

At the time of submitting this conference paper, I remain unable to generate meaningful results for most of the INRC2010 instances due to issues with my code. I am hoping to have results to present at the conference, which can be compared with the results generated by the winners of the INRC2010.

The primary aim of my work remains to embed heuristic methods into the column generation sub-problem in order to reduce the time spent in this phase. I am hoping to begin trialling a number of approaches and comparing their effects on both the runtime and solution quality. I will be able to use both the numerous instances provided by the INRC2010, as well as other data sets available to me in order to test these methods on a wide range of problems.

Acknowledgments

Although this is still very much a work in progress, I would like to acknowledge my supervisor, Dr. Andrew Mason, for his untiring patience and support. I would also like to extend my thanks to the others who have provided the body of work on which the BCP-nurse framework is based, most notably Faram Engineer and Anders Dohn, the latter of whom has helped me on a number of occasions.

References

Barnhart, C., E.L. Johnson, G.L. Post, M.W.P. Savelsbergh, and P.H. Vance. 1994. “Branch-and-Price: Column Generation for Solving Huge Integer Programs.”

Burke, E.K., P. De Causmaecker, S. Petrovic, and G. Vanden Berghe. 2001. “Fitness Evaluation for Nurse Scheduling Problems.”

Burke, E.K., T. Curtois, G. Post, R. Qu, and B. Veltman. 2005. “A Hybrid Heuristic Ordering and Variable Neighbourhood Search for the Nurse Rostering Problem.”

Dohn, A., A.J. Mason, and D. Ryan. 2010. “A Generic Solution Approach to Nurse Rostering.”

Engineer, F.G. 2003. “A Solution Approach to Optimally Solve the Generalised Rostering Problem.”

Haspeslagh, S., P. De Causmaecker, M. Stølevik, and A. Schaerf. 2010. “First International Nurse Rostering Competition 2010.”

Mason, A.J., and D. Ryan. 2009. “Customised Column Generation for Rostering Problems: Using Complie-time Customisation to create a Flexible C++ Engine for Staff Rostering.”

Mason, A.J., and M.C. Smith. 1998. “A Nested Column Generator for Solving Rostering Problems with Integer Programming.”

Comparing the Efficiency of Stores at New Zealand Post

Harriet Priddey and Kane Harton
Department of Engineering Science
University of Auckland
New Zealand
hpriddey@gmail.com, kharton@gmail.com

Abstract

New Zealand Post operates more than one hundred Post Stores around the country. They wished to find a way of comparing these stores impartially. The method used to do so in this work is Data Envelopment Analysis (DEA), a performance measurement technique utilising linear programming. Each store uses inputs to produce outputs, and an efficiency score is calculated based on how well it does so.

A significant part of the work involved selecting the inputs and outputs for the model, done using discussion, sensitivity analysis and statistical tests. To solve the DEA model, software was developed using the Python language and PuLP module. Extensions to the basic model included non-discretionary variables, weight restrictions, virtual weights and a second phase LP.

Next, the results of the DEA model were analysed. This involved identifying efficient stores and areas where they provide an example of good practice. Also, inefficient stores were assigned efficient peers to emulate, and efficiency targets. Some innovative ways of investigating the data included graphs, virtual weights and totalising monetary factors.

Key Words: Data Envelopment Analysis, Efficiency, Post Office

1 Introduction

New Zealand Post is one of the largest and most visible businesses in the country. They own and operate many retail stores around the country. They wished to find a way to compare these stores fairly and identify good and poor performers.

The approach used to do so in this work is Data Envelopment Analysis. This technique involves formulating and solving linear programs. Each store receives an efficiency rating based on how well it converts inputs to outputs, compared to other stores. This rating allows each store to choose its input and output weights so that the weighted input/output ratio is as good as possible. If, even with these weights, there is another store with a better ratio then the first store is inefficient. The results of the analysis also provide efficient peers for inefficient stores, and allow us to identify areas where stores perform particularly well.

The most appropriate inputs and outputs for the DEA model were identified using discussion, sensitivity analysis and statistical tests. The results of the model were reported to New Zealand Post in tabular and graphical form. They were analysed further by looking at input and output weights, and carrying out additional analyses using totalised values. To do this, the solver DEA.py was developed using Python and PuLP.

2 DEA Theory

Data Envelopment Analysis, commonly abbreviated to DEA, is a non-parametric performance measurement technique. It compares Decision Making Units (DMUs) in a multiple input, multiple output situation. DMUs are the organisation entities concerned with making decisions regarding inputs and outputs. A DMU is considered fully efficient "if and only if the performances of other DMUs does not show that some of its inputs or outputs can be improved without worsening some of its other inputs or outputs" (Cooper, Seiford, & Zhu, 2004).

There are many different DEA models. The one used in this work was proposed by Banker, Charnes and Cooper (1984) and is commonly known as the BCC model. A mathematical program is formulated and solved for each DMU. This program aims to maximise the ratio of weighted outputs to weighted inputs, known as the efficiency, subject to the efficiency of all DMUs being less than or equal to one. The model shown here is a linear form that eliminates the fractional ratio. It takes into account variable returns to scale. This means that differently sized DMUs may have different proportional changes in outputs when inputs change.

There are two forms of the BCC model, the envelopment form and the multiplier form. These two forms satisfy the primal-dual relationship of linear programming, thus the same optimal objective (the efficiency score) will be found by solving either form. The equations are detailed below, using notation from Cooper, Seiford and Zhu (2004).

Indices

i = input: $1, \ldots, m$.
r = output: $1, \ldots, s$.
o = DMU that is being solved for.
j = DMU: $1, \ldots, n$.

Parameters

x_{ij} = amount of input consumed by DMU j.
y_{rj} = amount of output produced by DMU j.

Decision variables

$\theta = z$ = efficiency score
λ_j = weighting of DMU j for current DMU.
s_i^-, s_r^+ = slack for input i or output r.
u_r = weighting of output r for current DMU.
v_i = weighting of input i for current DMU.
μ_o = dual variable, relates to returns-to-scale.

Model E: BCC Input-oriented Envelopment Form

Minimise $\theta - \varepsilon(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+)$
(E1) $\sum_{j=1}^{n} x_{ij}\lambda_j + s_i^- = \theta x_{io}$ $\quad for\ i = 1, 2, \ldots, m.$
(E2) $\sum_{j=1}^{n} y_{rj}\lambda_j - s_r^+ = y_{ro}$ $\quad for\ r = 1, 2, \ldots, s.$
(E3) $\sum_{j=1}^{n} \lambda_j = 1.$
$\lambda_j \geq 0.$

Model M: BCC Input-oriented Multiplier Form

Maximise $z = \sum_{r=1}^{s} \mu_r y_{ro} - \mu_o$

(M1) $\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} - \mu_o \leq 0$ $\quad\quad for\, j = 1, 2, \ldots, n.$

(M2) $\sum_{j=1}^{n} v_i x_{io} = 1.$

$u_j, v_i \geq 0, \mu_o$ free in sign.

3 The New Zealand Post Problem

The New Zealand Post group is the second largest retailer in New Zealand, with over 17,000 employees. There are over 300 Post Shops and 650 Post Centres around the country (New Zealand Post, 2010). Services available include postal solutions, bill payments, Kiwibank services, car registration, Bonus Bonds, travel bookings and insurance. These retail stores operate in varying conditions around the country, from major urban cities to rural areas. Stores have differing levels of competition within the area, serve populations of different socio-economic backgrounds, and have different costs, profits and levels of customer satisfaction. The stores are split into three categories (T1, T2, and T3) depending on the banking functions they provide.

With all these factors to consider, New Zealand Post wished to find a way of comparing stores fairly, and identifying the best and worst performers. In particular, they wished to look at Post Shops, which provide more services than Post Centres. So they approached the Department of Engineering Science. The solution proposed was to use DEA, which would allow stores to be compared based on a variety of factors. This would also allow each store to choose which factors to be assessed on, rather than using pre-determined weightings.

4 Model Development

4.1 Process for Choosing Inputs and Outputs

In this section we discuss the process of deriving inputs and outputs to measure efficiency using DEA. We highlight different types of inputs and outputs that needed special treatment within the DEA analysis and also mention difficulties encountered with some of them.

Identifying the inputs and outputs to use in a DEA assessment is crucial, and arguably the most important part of the whole process (Thanassoulis, 2001). Choosing different inputs and outputs can significantly affect the results, and make different DMUs appear better or worse. It is important to use as few factors as possible, because too many factors means that many DMUs can appear efficient by choosing their own particular input and output weightings. This reduces the discriminating power of the DEA analysis.

It is also very important to get input from the decision-makers when choosing inputs and outputs. They know the practical realities, and may have crucial insight that is unavailable to the analyst. If decision makers are not consulted then they may later contest the results, reducing the value of the DEA assessment. In this work, New Zealand Post analysts and store managers were involved through the process.

There were several meetings with New Zealand Post for this work. A list of initial inputs and outputs was compiled, then narrowed down based on relevance and which data was actually available. This provided the basis for the initial DEA model. Later there was another meeting with New Zealand Post, including several store managers. A large number of inputs and outputs were suggested. These were then refined based on what was available, further discussion, statistical tests and sensitivity analysis. Some specific issues with choosing inputs and outputs are detailed in the following sections.

4.2 Types of Post Stores

New Zealand Post divides its stores into three types, depending on the services they offer. These types relate to whether a store has the ability to grant home loans and perform some other banking functions. This increases a store's ability to generate revenue. Some stores have full banking functions (T3), some have limited functions (T2), and some have none (T1).

The initial model carried out separate DEA analyses for T1 stores and T3 stores. We raised the possibility of including both types of stores in a single model and using categorical variables. However New Zealand Post management believes the stores are far too different to include together. In DEA, if the categories are not comparable then a separate analysis should be performed for each category (Cooper, et al., 2004).

4.3 Non-discretionary Inputs

Non-discretionary inputs are those which influence performance, but which the given DMU has little or no influence over. There exist factors which post store management cannot control, but which may influence performance, for example store floor area or median income in an area. Hence, the model should take this into account.

This was done using the formulation proposed by Banker and Morey (1986). This model splits the inputs into two subsets, discretionary (I_D) and non-discretionary (I_N). The non-discretionary input constraint does not include θ on its right-hand side, meaning these inputs cannot be reduced. This model was coded into the DEA Python program.

4.4 Competition Input

Competition was considered to be an important input as it affects how well a store can perform. Data was collected in order to get a competition 'score' for each store. The physical address was found, and then Google Maps was used to identify the latitude and longitude (Harton, 2010a). The number of other post stores within 5km was found using an Excel spreadsheet and a macro. In DEA, a store is more efficient if it uses fewer resources. So, the number of post stores within 5km was treated as an undesirable input (Harton, 2010b). A number was applied to each store, lower if there were more post stores within the area, higher if there were fewer. This way a post store in a competitive environment will appear more efficient if it performs the same as another store in an 'easier' environment.

4.5 Staff Engagement

One of these issues related to a variable in the initial DEA model called engagement, which represented staff satisfaction with their working conditions. There were a number of difficulties with including this variable. For one, its measurement could be unreliable and dependent on a staff member's mood on one particular day. Also, it was not clear whether engagement should be included as an input or an output. It could be considered as an input because staff members who are more engaged contribute to earning more revenue and producing better outputs. However by doing this, the DEA model rewarded stores who minimised the input while still achieving outputs, i.e. had very unsatisfied staff. Engagement could also be included as an output because it could be considered an objective to have contented staff.

Engagement was trialled in the model, but because of these problems it was removed. It had some effect on results, as fewer stores were efficient when it was not included. However because of the interpretation problems, it did not make sense to include it.

4.6 FTE and Staff Costs

In the original data we had two related inputs, FTE (Full-time equivalent, the number of staff working at the store) and Staff Costs. It was expected that these inputs would be correlated, as employing more staff (higher FTE) will generally lead to higher staff costs. So a regression analysis was done which found that the R^2 values for correlation between Staff Costs and FTE were 65.55% for T1 and 51.79% for T3 stores.The T1 and T3 categorisations relate to the banking functions of a Post Store. The closer an R^2 value is to 1, the stronger the correlation, so this suggests that the variables are correlated, though not excessively.

Another way to investigate the correlation was to try a sensitivity analysis, by running the DEA twice more, with only one of Staff Costs and FTE included each time. This showed that fewer stores were considered efficient when only one of the inputs is used, as opposed to both. This is as would be expected.

Overall, both inputs could be left in the model to allow for different scenarios. An example of this is many long term staff members who have resulting higher pay, versus part time or temporary staff. Including both inputs does not result in many more stores becoming efficient. Usually in DEA it is not a bad thing to include correlated factors, as it allows for slightly different situations to be accommodated. If the factors are fully correlated then the same stores would be efficient whichever factor was included. However, this must be balanced against the possibility of including too many inputs and outputs so that results become meaningless.

4.7 Mystery Shopper Results Output

Mystery shopper results were a potential output for the DEA model. Mystery shoppers are customers sent into stores on certain days to measure the level of service. They report back using a questionnaire, and the scores are averaged in a year-to-date (YTD) value. There was debate over whether to include this. It is more variable than the expenditure-based measures, as it depends on which day it is measured. New Zealand Post did not want stores appearing efficient only through their mystery shopper rating.

To deal with this we used two approaches. The first was to simply exclude mystery shopper results, and then do a sensitivity analysis to examine the effects. This resulted in 44 efficient stores without including mystery shopper results, compared to 57 with it included, for the T1 data. This is desirable, as 57 efficient stores out of 91 in total are too many to provide valuable information. However, mystery shopper results represent customer satisfaction, and as such it would be good to include it in the model in some form.

The second approach was to use a weight restriction. There are several different ways to restrict weights, including absolute, relative and input-output weight restrictions (Cooper, et al., 2004). The most appropriate type in this situation was virtual weight restrictions. Virtual weights are the input/output weight multiplied by the actual value of the input/output, and they loosely represent the proportion of the efficiency score that is determined by the particular input or output. These were used rather than actual weights because of the large difference in scale between mystery shopper results and the other variables, meaning mystery shopper weights were disproportionally large. It was decided that mystery shopper should be restricted to a virtual weight of below 0.3. This loosely corresponds to contributing no more than 30% towards the efficiency score.

When comparing the results it was found that adding a weight restriction reduced the number of efficient stores. For the T1/T2 data, 50 stores were considered efficient with the weight restriction compared to 56 without. There were also several inefficient stores which had a lower efficiency score after the weight restriction. This supports the conclusion that weight restrictions allow more discrimination within the model.

4.8 Location

Location was provided as a categorical variable. Each store was in one of three categories, based on Statistics New Zealand definitions – main urban area, satellite urban area or independent urban area (Statistics New Zealand, 2010). Main urban areas are situated in major cities and towns. Satellite urban areas are settlements with strong links to a main urban area, while independent urban areas do not have such links; they are in more rural areas. There are a number of ways to include categorical variables in a DEA model, the most common of which was outlined by Banker and Morey (1986). It involves ranking the categories in order of most to least difficult to operate in. Then each DMU is compared to other DMUs who operate in the same or more difficult conditions. This means DMUs operating in difficult conditions are not unfairly penalised by being compared to DMUs who have it easier.

However, in the New Zealand Post situation it was not clear how the categories should be ordered. There are arguments for each way. Main urban area stores have a larger population to draw customers from, however independent urban area stores have less competition, and tend to have a steadier workforce and customer base.

So an initial DEA analysis was carried out without location included. The results from this were grouped according to which location category they came from. It was found for T1 data that 84% of independent urban area stores were efficient compared to 58% of main urban area stores. For T3 data, 100% of independent urban area stores were efficient compared to 51% of main urban area stores. This suggests that rural conditions can be considered easier to operate in. There were not enough satellite urban areas stores to gain useful results, however we can surmise that their difficulty would be between that of independent urban area and main urban area.

5 Final Model

The inputs and outputs of the final DEA model are shown in figure 1.

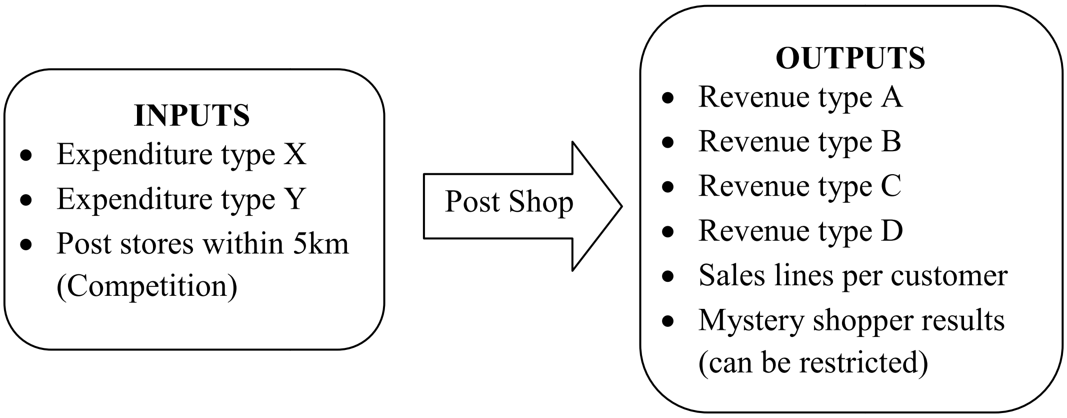


Figure 1: The Final New Zealand Post DEA Model

6 Open Source Python DEA Software

6.1 DEA Software

A program to solve DEA problems was written and developed. This program is called DEA.py, and is written in the Python language (Python Software Foundation, 2010). It uses the python library PuLP (PuLP Documentation Team, 2010) to write a series of linear programs which are then solved using the free solver coinOR (COIN-OR, 2010). It also uses the Python packages xlrd and xlwt for reading and writing to Microsoft Excel and yapgvb for producing a graph. All the components are free and open source. Some particular features of DEA.py are detailed in the following sections.

The input to this program is an Excel file, which needs to have specific formatting. The first column is a list of the DMUs, starting in the second row. There is a gap of one column, then a block of the inputs. The name of the input is in the top row, then below that the input values for each DMU. Next is another column gap then the non-discretionary inputs, if required. Finally comes a block of the outputs.

DEA.py can be run by opening it in Notepad or a similar text editor and change the required variables, then save. Then open a command window and enter the line *[location of python]\python [location of DEA.py]\DEA.py.* DEA.py generally solves the New Zealand Post problem in well under a minute. The more DMUs in the problem, the longer it will take, because an LP is solved for each DMU.

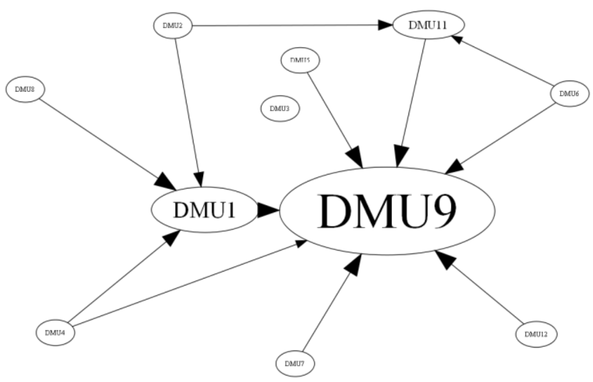


Figure 2: Graph produced by DEA.py for a small example problem

The Excel file output contains the solution to the DEA model. It gives the efficiency scores for each DMU, efficient peers, benchmarked DMUs, weights, targets and slacks.

An example of the output graph produced is shown in figure 2. The output graph represents each DMU as an oval. Arrows point from an inefficient DMU to its efficient peers, with larger arrow heads showing a larger weighting. The oval for a DMU is larger the more times it is referenced. Some DMUs are not linked to any others, meaning they are efficient but do not act as a peer to any others. This indicates that they may have somewhat different circumstances to any other DMUs. This graph gives an impression of which DMUs are most useful as examples of good practice.

6.2 Virtual Weights

Several extensions were added to the basic DEA software. One was adding a sheet that output the weighted data, also known as virtual weights. This is the input/output weight multiplied by the actual value of the input/output. Virtual weights do not depend on units of measurement (Cooper, et al., 2004), which is important in this case because inputs and outputs have very different scales. This allows comparison of the importance of inputs and outputs, and in addition can be easily graphed for visual comparison.

6.3 Envelopment and Multiplier Form

There are two versions of DEA.py, one solves the envelopment form of the model, the other solves the multiplier form. This allows results to be checked, and may provide alternative set of input/output weights. Solving DEA problems with the envelopment form is generally be faster to solve as there are fewer constraints. Using the multiplier model makes it easier to implement weight restrictions.

6.4 Weight Restrictions

Generally in DEA, all DMUs are free to choose their own input and output weights, in order to maximise efficiency, while maintaining feasibility for all other DMUs (Cooper, et al., 2004). An extension of the model was to add in input/output weight restrictions. This is desirable in cases where we know that some factors are not as important as others, and thus efficiency scores should not be overly influenced by these factors.

Absolute weight restrictions simply limit weights to within a specific range (Cooper, et al., 2004). These can be coded into DEA.py by following the instructions given in the code. A future development would be to allow weights restrictions to be included in the input sheet instead of hard coding.

6.5 Second Phase LP

Another addition was to implement a second phase LP. A DMU should not be considered efficient if it ignores an input or output efficiently. However this means that the non-Archimedean ε is included in the objective of the full DEA formulation. This is not possible to code. So the second stage LP attempts to maximise the sum of slacks. A DMU with an efficiency score of 100% is only truly efficient if its maximum sum of slacks is zero, meaning it is properly on the efficient frontier.

It was found that weakly efficient DMUs were not an issue for the New Zealand Post situation, as all stores with efficiency of 100% had their maximum sum of slacks equal to zero. This means that the efficiency score can safely be considered by itself. This was somewhat expected, as for the second phase to be relevant there would need to be DMUs that have the same values for a number of inputs and outputs. However if this software were to be used on a different data set, it may be very useful to have the second phase problem implemented.

7 Analysis

The results for this model, obtained from DEA.py, have been handed to New Zealand Post; however they cannot be given in full in this report because of privacy issues. Instead some examples of how the data was analysed are given.

The efficiency scores are the first useful information. Stores with a score of 100% are fully efficient. Others are inefficient, and should theoretically be able to improve their performance to reach the efficient frontier. It is also interesting to look at the efficient peers for each inefficient store. They provide a group of stores with similar strengths who can be used as an example of good practice. Reversing this, we can also look at the number of times a store acts as an efficient peer, or is 'benchmarked'.

The virtual weights are of interest, particularly for efficient stores. They indicate which areas a store performs particularly well in. It may be useful to investigate why this is. It should however be remembered that there can be multiple optimal solutions for the weights. For inefficient stores it is useful to look at the targets. These show how much a store should increase outputs or decrease inputs to become efficient.

Some graphs were also created to present the results to New Zealand Post, such as a bar graph of the percentage targets (figure 3) and a pie graph of the weights for efficient peers. It was also interesting to total up revenue and expenditure values so that a two-dimensional DEA analysis could be carried out, for easier graphing.

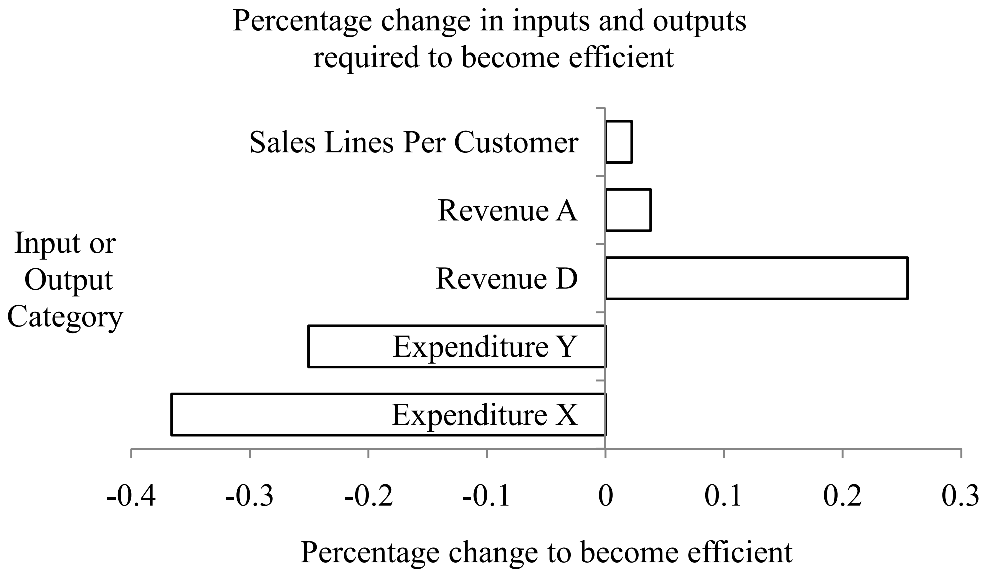


Figure 3: Percentage targets for a particular inefficient store

8 Conclusions

This project has applied Data Envelopment Analysis to New Zealand Post's retail stores. DEA is not a one-size-fits-all process; instead it must be applied differently depending on the situation, available data, and needs of the decision-makers. Choosing inputs and outputs is a vital part of the DEA model, and judgement is required. In this work statistical analysis, sensitivity analysis and discussion were used to narrow down the potential inputs and outputs.

This work involved the writing and development of the Python program DEA.py. It is now a useful, user-friendly piece of software that incorporates several DEA models and techniques. The results were analysed and examined in several ways.

The DEA process will have a number of benefits to New Zealand Post. They will be able to compare their retail stores, identify the best performers and areas where they do well. They will also identify inefficient stores, efficient peers for them to emulate and targets that they should be able to achieve.

Future work will involve making the solver DEA.py available open source. This means others will be able to develop it further and use it for their own DEA analyses.

Acknowledgments

Thank you to Dr Andrea Raith and A/Prof Matthias Ehrgott for their suggestions and guidance while supervising this work. Thank you also to Jody Snowdon and Raj Appana from New Zealand Post for their advice and assistance.

References

Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some Models for Estimating Technical and Scale Inefficiencies in Data Envelopment Analysis. *Management Science, 30*(9), 1078-1092.

Banker, R. D., & Morey, R. C. (1986). The Use of Categorical Variables in Data Envelopment Analysis. *Management Science, 32*(12), 1613-1627.

COIN-OR. (2010). COmputational INfrastructure for Operations Research. Retrieved November 8, 2010, from http://www.coin-or.org/

Cooper, W. W., Seiford, L. M., & Zhu, J. (2004). *Handbook on data envelopment analysis*. Boston: Kluwer Academic.

Harton, K. (2010a). *Data Envelopment Analysis with NZ Post*: Department of Engineering Science.

Harton, K. (2010b). *Using DEA.py*: Department of Engineering Science.

New Zealand Post. (2010). Our Major Business Streams. Retrieved August 18, 2010, from http://www.nzpost.co.nz/Cultures/en-NZ/AboutUs/OrganisationalInformation/WhatWeDo/OurMajorBusinessStreams.htm

PuLP Documentation Team. (2010). Optimization with PuLP. Retrieved September 20, 2010, from https://www.coin-or.org/PuLP/

Python Software Foundation. (2010). Python Programming Language – Official Website. Retrieved September 1, 2010, from http://www.python.org/

Statistics New Zealand. (2010). Urban/Rural Profile (experimental) Classification Categories. Retrieved September 11, 2010, from http://www.stats.govt.nz/browse_for_stats/people_and_communities/geographic_regions/urban-rural-profile-experimental-class-categories.aspx

Thanassoulis, E. (2001). *Introduction to the theory and application of data envelopment analysis : a foundation text with integrated software*. Norwell, Mass.: Kluwer Academic Publishers.

The Bi-objective Multi-Commodity Minimum Cost Flow Problem

Siamak Moradi
Department of Engineering Science
The University of Auckland
New Zealand
S.Moradi@auckland.ac.nz

Abstract

While there are several algorithms for bi-objective single commodity flow problems there is only a single paper on the problem with two commodities. In this paper, by extending the idea of the mentioned paper, we present an approach which allows us to split the bi-objective three commodity minimum cost flow problem into four standard bi-objective minimum cost flow problems with a single commodity and solve them with a parametric network simplex algorithm. Based on results obtained for the two and three commodity problems we extend the algorithm to address the general multi-commodity case.

Key words: Multi-commodity flow, bi-objective minimum cost flow problem

1 Introduction

In most real-world optimisation problems, there is usually more than one objective as well as several different commodities that have to be taken into account. Thus, it would seem that multi-objective multi-commodity flow models would be more appropriate for modelling real-world decision making situations in the field of network optimization. While there are several algorithms for multi-objective single commodity network flow problems e.g. Raith and Ehrgott (2009) and Sedeño-Noda (2003), there is only a single paper on the problem with two commodities by Sedeño-Noda, Gonzalez-Martin, and Gutierrez (2005). Their main idea is introducing a change of variables in the formulation of the bi-objective two commodity minimum cost flow (B2CMCF) problem and to split the problem into two bi-objective minimum cost flow (BMCF) problems with a single commodity. We extend the idea and propose a change of variables in the bi-objective three commodity minimum cost flow (B3CMCF) problem which allows us to split the original problem into four standard BMCF problems. Based on results obtained for the two and three commodity problems, we extend the method to address the general multi-commodity case.

The paper is organised as follows: In Section 2, the bi-objective multi-commodity minimum cost flow (BMCMCF) problem is introduced. We present the change of variables approach for B3CMCF in Section 3. In Section 4, a method to solve BMCMCF is explained. Finally, in Section 5, we conclude the paper.

2 Bi-objective multi-commodity minimum cost flow problem

In this section, terminology and basic theory of BMCMCF problems are introduced. We will follow the same notation, definition and formulation as in Sedeño-Noda, Gonzalez-Martin, and Gutierrez (2005) and Hu (1963). Let $G = (V, A)$ be an antisymmetric directed network with a set of nodes or vertices $V = \{1, 2, \ldots, n\}$ and a set of arcs $A \subseteq V \times V$ with $|A| = m$. A directed network is antisymmetric if $(i, j) \in A \Rightarrow (j, i) \notin A$. There are q commodities sharing the capacity $u_{ij} \geqslant 0$ for each arc $(i, j) \in A$. We distinguish two special groups of nodes in G; the source nodes $S = \{s^1, s^2, \ldots, s^q\}$, and the sink nodes $T = \{t^1, t^2, \ldots, t^q\}$. For each commodity $k = 1, 2, \ldots, q$, b^k units of flow should be shipped from its source node s^k to its sink node t^k. Let $\left(c^k_{ij}, d^k_{ij}\right)$ be the pair of unit flow costs on arc (i, j) for commodity k and x^k_{ij} presents the amount of flow going through arc (i, j) for commodity k.

The BMCMCF problem is defined by the following mathematical program:

$$\min \qquad f(x) = \begin{cases} f^1(x) = \displaystyle\sum_{k=1,\ldots,q} \sum_{(i,j)\in A} c^k_{ij} x^k_{ij} \\ f^2(x) = \displaystyle\sum_{k=1,\ldots,q} \sum_{(i,j)\in A} d^k_{ij} x^k_{ij} \end{cases}$$

$$\text{s.t} \qquad \sum_{\{j|(i,j)\in A\}} x^k_{ij} - \sum_{\{j|(j,i)\in A\}} x^k_{ij} = \begin{cases} b^k & \text{if } i \in S \\ 0 & \text{if } i \in V - \{S \cup T\}, \; k = 1, 2, \ldots, q \\ -b^k & \text{if } i \in T \end{cases}$$

$$\sum_{k=1,2,\ldots q} \mid x^k_{ij} \mid \leqslant u_{ij}, (i, j) \in A. \tag{2.1}$$

For notational convenience, we will denote symmetric directed network $G' = (V', A')$ where $V' = V$ and A' contains the arcs (i, j) and (j, i), with capacities $u'_{ij} = u'_{ji} = u_{ij}$, for each arc $(i, j) \in A$. Figure 1 illustrates the two networks.

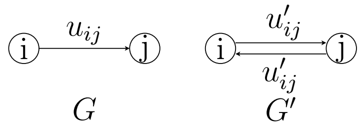


Figure 1: Network G' defined from the original network G

Definition 1. *Let X denote the feasible space for BMCMCF problem and $f(X) = \{(f^1(x), f^2(x)) \mid x \in X\}$ the objective space. A feasible solution $\hat{x} \in X$ of the BMCMCF problem is efficient if and only if there does not exist any $x' \in X$ with $f^i(x') \leqslant f^i(\hat{x})$ for both objectives and with $f^i(x') \neq f^i(\hat{x})$ for at least one i, and $i = 1, 2$.*

We will denote by $E[X]$ the set of efficient solutions of X and by $E[f(X)] = \{f(x) \mid x \in E[X]\}$ the set of non dominated points of $f(X)$. In this case, X is a compact polyhedron, and therefore $f(X)$ is also a compact polyhedron. Thus, $E[X]$ and $E[f(X)]$ through the set of efficient extreme points in the decision space $E_{ex}[X]$ and the objective space $E_{ex}[f(X)]$, respectively. Consequently, we are interested in obtaining $E_{ex}[f(X)]$ and $E_{ex}[X]$.

3 Change of variables method

Sedeño-Noda, Gonzalez-Martin, and Gutierrez (2005) and Sedeño-Noda, Gonzalez-Martin, and Alonso-Rodriguez (2008) presented a change of variables method in the formulation of the bi-objective two commodity minimum cost flow problem (B2CMCF), that splits the problem into two BMCF problems with a single commodity. We extend the idea and propose a change of variables in the bi-objective three commodity minimum cost flow (B3CMCF) problem which allows us to split the original problem into four standard BMCF problems.

3.1 Change of variables method for B2CMCF

In this subsection we briefly explain the method of Sedeño-Noda, Gonzalez-Martin, and Gutierrez (2005) for B2CMCF.
Let

$$x_{ij}^1 + x_{ij}^2 = z_{ij}^1 - z_{ji}^1,\ x_{ij}^1 - x_{ij}^2 = z_{ij}^2 - z_{ji}^2 \tag{3.1}$$

and hence,

$$x_{ij}^1 = \frac{z_{ij}^1 - z_{ji}^1 + z_{ij}^2 - z_{ji}^2}{2},\ x_{ij}^2 = \frac{z_{ij}^1 - z_{ji}^1 - z_{ij}^2 + z_{ji}^2}{2}. \tag{3.2}$$

Considering the above change of variables, the objective functions of the B2CMCF problem (2.1) for $q = 2$, can be formulated as follows:

$$f^1(x) = \sum_{(i,j)\in A} c_{ij}^1 x_{ij}^1 + c_{ij}^2 x_{ij}^2 = \frac{1}{2}\sum_{(i,j)\in A}\left(c_{ij}^1 + c_{ij}^2\right)\left(x_{ij}^1 + x_{ij}^2\right) + \frac{1}{2}\sum_{(i,j)\in A}\left(c_{ij}^1 - c_{ij}^2\right)\left(x_{ij}^1 - x_{ij}^2\right),$$

$$f^2(x) = \sum_{(i,j)\in A} d_{ij}^1 x_{ij}^1 + d_{ij}^2 x_{ij}^2 = \frac{1}{2}\sum_{(i,j)\in A}\left(d_{ij}^1 + d_{ij}^2\right)\left(x_{ij}^1 + x_{ij}^2\right) + \frac{1}{2}\sum_{(i,j)\in A}\left(d_{ij}^1 - d_{ij}^2\right)\left(x_{ij}^1 - x_{ij}^2\right).$$

To simplify the formulation, α_{ij}^p and β_{ij}^p are introduced as costs associated with the arc $(i,j) \in A'$ and the pth objective $(p = 1, 2)$. They are given by

$$\alpha_{ij}^1 = \begin{cases} \frac{1}{2}\left(c_{ij}^1 + c_{ij}^2\right) & \text{if } (i,j) \in \text{A}, \\ -\frac{1}{2}\left(c_{ji}^1 + c_{ji}^2\right) & \text{if } (j,i) \in \text{A}, \end{cases} \quad \text{and} \quad \alpha_{ij}^2 = \begin{cases} \frac{1}{2}\left(d_{ij}^1 + d_{ij}^2\right) & \text{if } (i,j) \in \text{A}, \\ -\frac{1}{2}\left(d_{ji}^1 + d_{ji}^2\right) & \text{if } (j,i) \in \text{A}, \end{cases} \quad \text{and}$$

$$\beta_{ij}^1 = \begin{cases} \frac{1}{2}\left(c_{ij}^1 - c_{ij}^2\right) & \text{if } (i,j) \in \text{A}, \\ -\frac{1}{2}\left(c_{ji}^1 - c_{ji}^2\right) & \text{if } (j,i) \in \text{A}, \end{cases} \quad \text{and} \quad \beta_{ij}^2 = \begin{cases} \frac{1}{2}\left(d_{ij}^1 - d_{ij}^2\right) & \text{if } (i,j) \in \text{A}, \\ -\frac{1}{2}\left(d_{ji}^1 - d_{ji}^2\right) & \text{if } (j,i) \in \text{A}. \end{cases}$$

Thus the objective functions become as follows:

$$f^1(x) = \sum_{(i,j)\in A'} \alpha_{ij}^1 z_{ij}^1 + \sum_{(i,j)\in A'} \beta_{ij}^1 z_{ij}^2 = f^1\left(z^1\right) + f^1\left(z^2\right),$$

$$f^2(x) = \sum_{(i,j)\in A'} \alpha_{ij}^2 z_{ij}^1 + \sum_{(i,j)\in A'} \beta_{ij}^2 z_{ij}^2 = f^2\left(z^1\right) + f^2\left(z^2\right).$$

For each fixed $i \in V$, the flow conservation constraints (2.1) are added and subtracted for $k = 1, 2$. Then making the change of variables, and considering that in the antisymmetric network G we have $\{j \mid (i,j) \in A\} \cap \{j \mid (j,i) \in A\} = \emptyset$ we obtain:

$$\sum_{\{j|(i,j)\in A'\}} z_{ij}^p - \sum_{\{j|(j,i)\in A'\}} z_{ji}^p = \begin{cases} b^1 & \text{if } i = s^1 \\ (-1)^{p-1} b^2 & \text{if } i = s^2 \\ 0 & \text{if } i \in V - \{s^1, s^2, t^1, t^2\} \\ -b^1 & \text{if } i = t^1 \\ (-1)^{p-1} b^2 & \text{if } i = t^2. \end{cases}, \ p = 1, 2.$$

Also for constraints (2.1) we can write $0 \leqslant z_{ij}^p \leqslant u'_{ij}$ for all arcs $(i,j) \in A'$ and $p = 1, 2$.

Applying the change of variables approach, the B2CMCF problem can be reformulated to give

$$\begin{aligned} \min f(x) &= \left(\sum_{(i,j)\in A'} \alpha_{ij}^1 z_{ij}^1 + \sum_{(i,j)\in A'} \beta_{ij}^1 z_{ij}^2, \sum_{(i,j)\in A'} \alpha_{ij}^2 z_{ij}^1 + \sum_{(i,j)\in A'} \beta_{ij}^2 z_{ij}^2 \right) = \\ & \left(f^1\left(z^1\right) + f^1\left(z^2\right), f^2\left(z^1\right) + f^2\left(z^2\right) \right) \end{aligned} \tag{3.3}$$

$$\text{s.t:} \quad \sum_{\{j|(i,j)\in A'\}} z_{ij}^1 - \sum_{\{j|(j,i)\in A'\}} z_{ji}^1 = \begin{cases} b^1 & \text{if } i = s^1 \\ b^2 & \text{if } i = s^2 \\ 0 & \text{if } i \in V - \{s^1, s^2, t^1, t^2\}, \\ -b^1 & \text{if } i = t^1 \\ -b^2 & \text{if } i = t^2 \end{cases} \tag{3.4}$$

$$\sum_{\{j|(i,j)\in A'\}} z_{ij}^2 - \sum_{\{j|(j,i)\in A'\}} z_{ji}^2 = \begin{cases} b^1 & \text{if } i = s^1 \\ -b^2 & \text{if } i = s^2 \\ 0 & \text{if } i \in V - \{s^1, s^2, t^1, t^2\}, \\ -b^1 & \text{if } i = t^1 \\ b^2 & \text{if } i = t^2 \end{cases} \tag{3.5}$$

$$0 \leqslant z_{ij}^p \leqslant u'_{ij}, p = 1, 2 \forall (i,j) \in A'. \tag{3.6}$$

Now the objective functions and the constraints of the above formulation can be separated in accordance with the variables z^p. According to the Proposition 1 in Sedeño-Noda, Gonzalez-Martin, and Gutierrez (2005), a solution of the above formulation is derived from the Cartesian product of the efficient sets of two classical BMCF problems, the first one with objective functions $\min\left(f^1\left(z^1\right), f^2\left(z^1\right)\right)$ with constraints (3.4) and (3.6) and the second one with objective functions $\min\left(f^1\left(z^2\right), f^2\left(z^2\right)\right)$ and constraints (3.5) and (3.6).

3.2 Change of variables method for B3MCF

Using the BMCMCF formulation (2.1) with $q = 3$, the B3CMCF problem can be stated as follows

$$\min z \quad = \quad \begin{cases} f^1(x) = & \sum_{(i,j)\in A} c^1_{ij}x^1_{ij} + c^2_{ij}x^2_{ij} + c^3_{ij}x^3_{ij} \\ f^2(x) = & \sum_{(i,j)\in A} d^1_{ij}x^1_{ij} + d^2_{ij}x^2_{ij} + d^3_{ij}x^3_{ij} \end{cases} \tag{3.7}$$

$$\text{s.t.} \qquad \sum_{\{j|(i,j)\in A\}} x^k_{ij} - \sum_{\{j|(j,i)\in A\}} x^k_{ij} = \begin{cases} b^k & \text{if } i = s^k \\ 0 & \text{if } i \in V - \{s^k, t^k\}, k = 1,2,3 \\ -b^k & \text{if } i = t^k \end{cases} \tag{3.8}$$

$$\sum_{k=1,2,3} | x^k_{ij} | \leqslant u_{ij}, (i,j) \in A. \tag{3.9}$$

Now we propose the following change of variables which allows us to split the above problem to four standard BMCF problems with a single commodity. Let

$$\begin{aligned} x^1_{ij} + x^2_{ij} + x^3_{ij} &= z^1_{ij} - z^1_{ji} \\ x^1_{ij} + x^2_{ij} - x^3_{ij} &= z^2_{ij} - z^2_{ji} \\ x^1_{ij} - x^2_{ij} + x^3_{ij} &= z^3_{ij} - z^3_{ji} \\ x^1_{ij} - x^2_{ij} - x^3_{ij} &= z^4_{ij} - z^4_{ji} \end{aligned} \tag{3.10}$$

and hence:

$$\begin{aligned} x^1_{ij} &= \frac{z^1_{ij} - z^1_{ji} + z^2_{ij} - z^2_{ji} + z^3_{ij} - z^3_{ji} + z^4_{ij} - z^4_{ji}}{4} \\ x^2_{ij} &= \frac{z^1_{ij} - z^1_{ji} + z^2_{ij} - z^2_{ji} - z^3_{ij} + z^3_{ji} - z^4_{ij} + z^4_{ji}}{4} \\ x^3_{ij} &= \frac{z^1_{ij} - z^1_{ji} - z^2_{ij} + z^2_{ji} + z^3_{ij} - z^3_{ji} - z^4_{ij} + z^4_{ji}}{4}. \end{aligned} \tag{3.11}$$

Considering the above change of variables, the objective function $f^1(x)$ of the B3CMCF problem in equation (3.7) can be formulated as follows:

$$\begin{aligned} & f^1(x) = \sum_{(i,j)\in A} c^1_{ij}x^1_{ij} + c^2_{ij}x^2_{ij} + c^3_{ij}x^3_{ij} = \\ & \frac{1}{4}\sum_{(i,j)\in A} \left(c^1_{ij} + c^2_{ij} + c^3_{ij}\right)\left(x^1_{ij} + x^2_{ij} + x^3_{ij}\right) + \\ & \frac{1}{4}\sum_{(i,j)\in A} \left(c^1_{ij} + c^2_{ij} - c^3_{ij}\right)\left(x^1_{ij} + x^2_{ij} - x^3_{ij}\right) + \\ & \frac{1}{4}\sum_{(i,j)\in A} \left(c^1_{ij} - c^2_{ij} + c^3_{ij}\right)\left(x^1_{ij} - x^2_{ij} + x^3_{ij}\right) + \\ & \frac{1}{4}\sum_{(i,j)\in A} \left(c^1_{ij} - c^2_{ij} - c^3_{ij}\right)\left(x^1_{ij} - x^2_{ij} - x^3_{ij}\right). \end{aligned} \tag{3.12}$$

To simplify the formulation (3.12) consider α^1_{ij}, β^2_{ij}, γ^1_{ij} and δ^1_{ij} as the costs associated with the arc $(i,j) \in A'$ and the objective $f^1(x)$, and given by :

$$\alpha^1_{ij} = \begin{cases} \frac{1}{4}\left(c^1_{ij} + c^2_{ij} + c^3_{ij}\right) & \text{if } (i,j) \in \text{A}, \\ -\frac{1}{4}\left(c^1_{ji} + c^2_{ji} + c^3_{ji}\right) & \text{if } (j,i) \in \text{A}, \end{cases} \quad \text{and} \quad \beta^1_{ij} = \begin{cases} \frac{1}{4}\left(c^1_{ij} + c^2_{ij} + -c^3_{ij}\right) & \text{if } (i,j) \in \text{A}, \\ -\frac{1}{4}\left(c^1_{ji} + c^2_{ji} - c^3_{ji}\right) & \text{if } (j,i) \in \text{A}, \end{cases}$$

$$\gamma^1_{ij} = \begin{cases} \frac{1}{4}\left(c^1_{ij} - c^2_{ij} + c^3_{ij}\right) & \text{if } (i,j) \in \text{A}, \\ -\frac{1}{4}\left(c^1_{ji} - c^2_{ji} + c^3_{ji}\right) & \text{if } (j,i) \in \text{A}, \end{cases} \quad \text{and} \quad \delta^1_{ij} = \begin{cases} \frac{1}{4}\left(c^1_{ij} - c^2_{ij} - c^3_{ij}\right) & \text{if } (i,j) \in \text{A}, \\ -\frac{1}{4}\left(c^1_{ji} - c^2_{ji} - c^3_{ji}\right) & \text{if } (j,i) \in \text{A}. \end{cases}$$

Thus the objective function $f^1(x)$ becomes

$$f^1(x) = \sum_{(i,j)\in A'} \alpha^1_{ij} z^1_{ij} + \sum_{(i,j)\in A'} \beta^1_{ij} z^2_{ij} + \sum_{(i,j)\in A'} \gamma^1_{ij} z^3_{ij} + \sum_{(i,j)\in A'} \delta^1_{ij} z^4_{ij} = f^1\left(z^1\right) + f^1\left(z^2\right) + f^1\left(z^3\right) + f^1\left(z^4\right). \quad (3.13)$$

Doing the same things for the objective function $f^2(x)$ we can rewrite the function as follows:

$$f^2(x) = \sum_{(i,j)\in A'} \alpha^2_{ij} z^1_{ij} + \sum_{(i,j)\in A'} \beta^2_{ij} z^2_{ij} + \sum_{(i,j)\in A'} \gamma^2_{ij} z^3_{ij} + \sum_{(i,j)\in A'} \delta^2_{ij} z^4_{ij} = f^2\left(z^1\right) + f^2\left(z^2\right) + f^2\left(z^3\right) + f^2\left(z^4\right). \quad (3.14)$$

For each fixed $i \in V$, the flow conservation constraints (3.8) are combined by using different combinations of addition and subtraction, similar to the multipliers of x^k_{ij} from (3.11), for $k = 1, 2, 3$. Then making the change of variables, we obtain the following constraints:

$$\sum_{\{j|(i,j)\in A'\}} z^1_{ij} - \sum_{\{j|(j,i)\in A'\}} z^1_{ji} = \begin{cases} b^1 & \text{if } i = s^1 \\ b^2 & \text{if } i = s^2 \\ b^3 & \text{if } i = s^3 \\ 0 & \text{if } i \in V - \{s^1, s^2, s^3, t^1, t^2, t^3\} \\ -b^1 & \text{if } i = t^1 \\ -b^2 & \text{if } i = t^2 \\ -b^3 & \text{if } i = t^3 \end{cases} \quad (3.15)$$

$$\sum_{\{j|(i,j)\in A'\}} z^2_{ij} - \sum_{\{j|(j,i)\in A'\}} z^2_{ji} = \begin{cases} b^1 & \text{if } i = s^1 \\ b^2 & \text{if } i = s^2 \\ -b^3 & \text{if } i = s^3 \\ 0 & \text{if } i \in V - \{s^1, s^2, s^3, t^1, t^2, t^3\} \\ -b^1 & \text{if } i = t^1 \\ -b^2 & \text{if } i = t^2 \\ b^3 & \text{if } i = t^3 \end{cases} \quad (3.16)$$

$$\sum_{\{j|(i,j)\in A'\}} z_{ij}^3 - \sum_{\{j|(j,i)\in A'\}} z_{ji}^3 = \begin{cases} b^1 & \text{if } i = s^1 \\ -b^2 & \text{if } i = s^2 \\ b^3 & \text{if } i = s^3 \\ 0 & \text{if } i \in V - \{s^1, s^2, s^3, t^1, t^2, t^3\} \\ -b^1 & \text{if } i = t^1 \\ b^2 & \text{if } i = t^2 \\ -b^3 & \text{if } i = t^3 \end{cases} \tag{3.17}$$

$$\sum_{\{j|(i,j)\in A'\}} z_{ij}^4 - \sum_{\{j|(j,i)\in A'\}} z_{ji}^4 = \begin{cases} b^1 & \text{if } i = s^1 \\ -b^2 & \text{if } i = s^2 \\ -b^3 & \text{if } i = s^3 \\ 0 & \text{if } i \in V - \{s^1, s^2, s^3, t^1, t^2, t^3\} \\ -b^1 & \text{if } i = t^1 \\ b^2 & \text{if } i = t^2 \\ b^3 & \text{if } i = t^3. \end{cases} \tag{3.18}$$

For constraint (3.9) we have:

$$\sum_{k=1,2,3} | x_{ij}^k | \leqslant u_{ij}, \iff \begin{cases} -u_{ij} \leqslant x_{ij}^1 + x_{ij}^2 + x_{ij}^3 \leqslant u_{ij} \\ -u_{ij} \leqslant x_{ij}^1 + x_{ij}^2 - x_{ij}^3 \leqslant u_{ij} \\ -u_{ij} \leqslant x_{ij}^1 - x_{ij}^2 + x_{ij}^3 \leqslant u_{ij} \\ -u_{ij} \leqslant x_{ij}^1 - x_{ij}^2 - x_{ij}^3 \leqslant u_{ij}. \end{cases}$$

After changing the variables the new set of constraints follows $-u_{ij} \leqslant z_{ij}^p - z_{ji}^p \leqslant u_{ij}$ for $p = 1, 2, 3, 4$. And for symmetric network G', the bounds are stated as $0 \leqslant z_{ij}^p \leqslant u'_{ij}$ for all arcs $(i, j) \in A'$ and $p = 1, 2, 3, 4$.

Applying the above change of variables approach the B3CMCF problem can be split into the four following problems:

$$\begin{aligned} \min \quad & \left(f^1\left(z^1\right), f^2\left(z^1\right)\right) = \left(\sum_{(i,j)\in A'} \alpha_{ij}^1 z_{ij}^1, \sum_{(i,j)\in A'} \alpha_{ij}^2 z_{ij}^1 \right) \\ \text{s.t:} \quad & \text{constraints (3.15)} \\ & 0 \leqslant z_{ij}^1 \leqslant u'_{ij}, \forall (i,j) \in A' \end{aligned} \tag{3.19}$$

$$\begin{aligned} \min \quad & \left(f^1\left(z^2\right), f^2\left(z^2\right)\right) = \left(\sum_{(i,j)\in A'} \beta_{ij}^1 z_{ij}^2, \sum_{(i,j)\in A'} \beta_{ij}^2 z_{ij}^2 \right) \\ \text{s.t:} \quad & \text{constraints (3.16)} \\ & 0 \leqslant z_{ij}^2 \leqslant u'_{ij}, \forall (i,j) \in A' \end{aligned} \tag{3.20}$$

$$\begin{aligned}
\min \quad & \left(f^1\left(z^3\right), f^2\left(z^3\right)\right) = \left(\sum_{(i,j)\in A'} \gamma^1_{ij} z^3_{ij}, \sum_{(i,j)\in A'} \gamma^2_{ij} z^3_{ij}\right) \\
\text{s.t:} \quad & \text{constraints (3.17)} \\
& 0 \leqslant z^3_{ij} \leqslant u'_{ij}, \forall (i,j) \in A'
\end{aligned} \tag{3.21}$$

$$\begin{aligned}
\min \quad & \left(f^1\left(z^4\right), f^2\left(z^4\right)\right) = \left(\sum_{(i,j)\in A'} \delta^1_{ij} z^4_{ij}, \sum_{(i,j)\in A'} \delta^2_{ij} z^4_{ij}\right) \\
\text{s.t:} \quad & \text{constraints (3.18)} \\
& 0 \leqslant z^4_{ij} \leqslant u'_{ij}, \forall (i,j) \in A'
\end{aligned} \tag{3.22}$$

The four above problems are classical BMCF problems with one commodity which can be solved by several existing algorithms. Then undoing the change of variables the efficient set of the B3CMCF problem can be derived.

3.3 Parametric simplex algorithm

As mentioned above, the change of variables method splits the B3CMCF problem into four BMCF problems with one single commodity which can be solved by any bi-objective one commodity minimum cost flow solution method. We used the parametric simplex method, as in the phase 1 algorithm used by Raith and Ehrgott (2009). In this method, one of the two lexicographically optimal solutions, e.g., the $lex\,(1,2)$-best solution, is obtained with a single-objective network simplex algorithm. As there are two cost components associated with each arc in the network simplex algorithm, the reduced cost of each arc also consists of two components. In each iteration of the network simplex algorithm, candidate entering arcs are selected with minimal ratio of their reduced cost derived from the current supported efficient solution. By doing so, the procedure generates a complete set of extreme efficient solutions moving in a left-to-right fashion. The parametric simplex algorithm finishes when no candidate arcs to enter the basis can be found which indicates that the $lex\,(2,1)$ best solution is obtained.

3.4 An example

Let us consider the example depicted in Figure 2. By applying the change of variables method for this B3CMCF problem, four standard BMCF problems with one commodity are obtained as shown in Figure 3, then we solved these four problems with the network simplex algorithm.

In Table 1, the efficient extreme points in the objective space for these four problems with their corresponding values $\left(f^1\left(z^p\right), f^2\left(z^p\right)\right)$, $p = 1, 2, 3, 4$ are shown.

In Table 2, the three commodity efficient extreme points in the objective spaces with their corresponding values $\left(f^1\left(x\right), f^2\left(x\right)\right)$ are shown.

Table 1: Efficient extreme points for the one commodity problems

	$(s^1,1)$	$(1,s^1)$	$(s^2,1)$	$(1,s^2)$	$(s^3,1)$	$(1,s^3)$	$(1,2)$	$(2,1)$	$(2,t^1)$	$(t^1,2)$	$(2,t^2)$	$(t^2,2)$	$(2,t^3)$	$(t^3,2)$	(s^1,t^1)	(t^1,s^1)	(s^3,t^3)	(t^3,s^3)	$f^1(z)$	$f^2(z)$	$\lambda^{t-1} \leqslant \lambda \leqslant \lambda^t$
Efficient extreme points for the (BP1) problem																					
$(z^1)^1$	0	12	20	0	0	5	10	7	0	5	20	10	0	12	30	0	30	0	159.25	305.5	$0 \leqslant \lambda \leqslant 1$
Efficient extreme points for the (BP2) problem																					
$(z^2)^1$	0	12	20	0	2	20	0	10	0	18	20	0	8	20	30	0	0	7	96.5	-0.5	$0 \leqslant \lambda \leqslant 1$
Efficient extreme points for the (BP3) problem																					
$(z^3)^1$	0	10	0	20	20	0	0	10	20	0	0	20	10	20	28	0	30	25	25.5	29.5	$0.56 \leqslant \lambda \leqslant 1$
$(z^3)^2$	15	0	0	20	0	5	0	10	0	5	0	20	20	5	3	0	30	0	50.5	-1.75	$0 \leqslant \lambda \leqslant 0.56$
Efficient extreme points for the (BP4) problem																					
$(z^4)^1$	20	0	0	20	10	20	0	10	0	10	0	20	20	0	0	2	0	15	-4	34.5	$0.67 \leqslant \lambda \leqslant 1$
$(z^4)^2$	20	0	0	20	20	15	10	5	5	0	0	20	20	0	0	2	0	30	0.25	27	$0 \leqslant \lambda \leqslant 0.67$

Table 2: Efficient extreme points for the main three commodity problem

	$(s^1,1)$	$(s^2,1)$	$(s^3,1)$	$(1,2)$	$(2,t^1)$	$(2,t^2)$	$(2,t^3)$	(s^1,t^1)	(s^3,t^3)	$f^1(x)$	$f^2(x)$	$\lambda^{t-1} \leqslant \lambda \leqslant \lambda^t$
x_1	(-3.5,-8.5,-7.5)	(0,20,0)	(-3.25,-8.25,7)	(-6.75,3.25,-0.5)	(-3.25,-8.25,7)	(0,20,0)	(-3.5,-8.5,-7.5)	(21.5,8.5,7.5)	(3.25,8.25,18)	113	330.25	$0.67 \leqslant \lambda \leqslant 1$
x_2	(-3.5,-8.5,-7.5)	(0,20,0)	(0.5,-12,7)	(-3,-0.5,-0.5)	(0.5,-12,7)	(0,20,0)	(-3.5,-8.5,-7.5)	(21.5,8.5,7.5)	(-0.5,12,18)	88	361.5	$0.56 \leqslant \lambda \leqslant 0.67$
x_3	(2.75,-14.75,-1.25)	(0,20,0)	(-5.75,-5.75,0.25)	(-3,-0.5,-0.5)	(-5.75,-5.75,0.75)	(0,20,0)	(2.75,-14.75,-1.25)	(15.25,14.75,1.25)	(5.75,5.75,2425)	113	330.25	$0 \leqslant \lambda \leqslant 0.56$

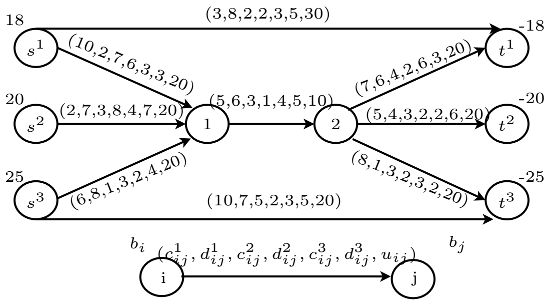


Figure 2: B3CMCF example

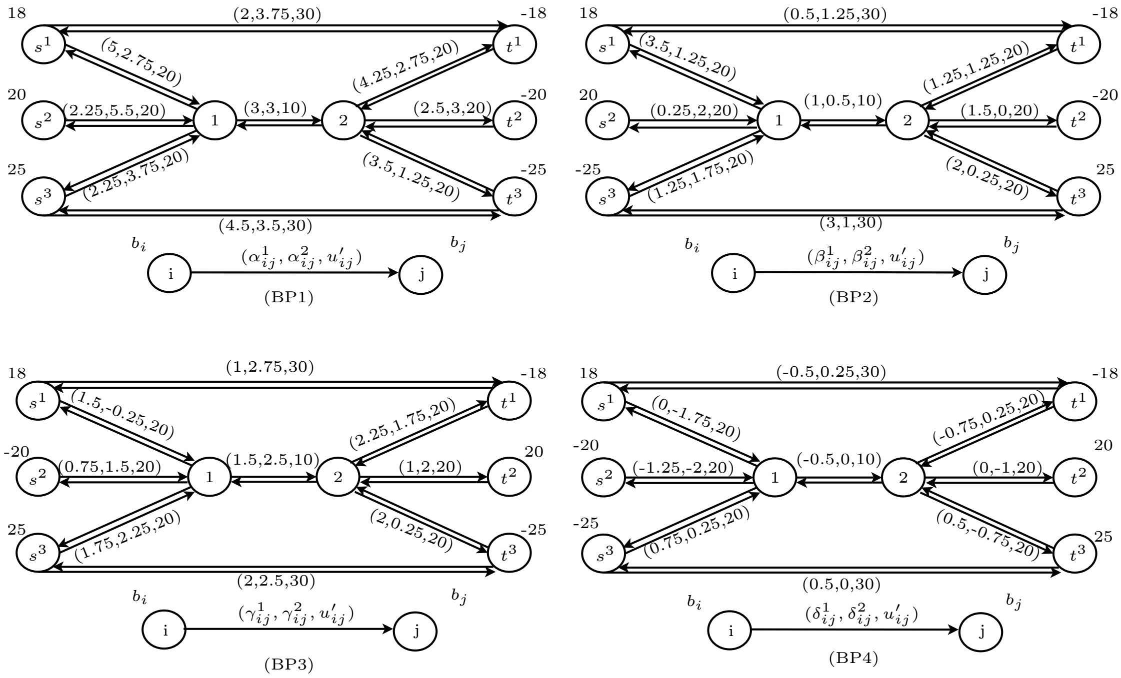


Figure 3: Networks with one commodity obtained from the B3CMCF problem

4 Change of variables method for BMCMCF

Based on results obtained for two and three commodity problems we can extend the change of variables method to address the general multi-commodity case. In this extension to solve the BMCMCF problem with q commodities ($q \geqslant 3$), by combining $x^k_{ij}, k = 1, 2, ..., q$ using different combinations of addition and subtraction in formulation (2.1) and introducing 2^{q-1} new variables $z^p_{ij}, p = 1, 2, \ldots, 2^{q-1}$, similar to (3.10), the BMCMCF problem can be split into 2^{q-1} classical BMCF problems with single commodity. These 2^{q-1} problems can be solved by different standard algorithms and then by undoing the change of variables the efficient set of the BMCMCF problem can be derived.

5 Conclusion

In this paper we have presented a method to solve the bi-objective multi-commodity minimum cost flow problem. For that we introduced a change of variables method which splits the BMCMCF problem with q commodities into 2^{q-1} classical BMCF problems with a single commodity, that can be solved by different standard algorithms.

Acknowledgements I am heartily thankful to my supervisor, Associate Professor Matthias Ehrgott, whose encouragement, supervision and support enabled me to develop an understanding of the subject. I would also like to sincerely thank Dr Andrea Raith for her assistance and guidance throughout this research.

References

Hu, T. C. 1963. "Multi-Commodity Network Flows." *Operations Research* 11 (3): 344–360.

Raith, A., and M. Ehrgott. 2009. "A two-phase algorithm for the biobjective integer minimum cost flow problem." *Computers & Operations Research* 36 (6): 1945–1954.

Sedeño-Noda, A. 2003. "An alternative method to solve the biobjective minimum cost flow problem." *Asia-Pacific Journal of Operational Research* 20 (2): 241–260.

Sedeño-Noda, A., C. Gonzalez-Martin, and S. Alonso-Rodriguez. 2008. "A new strategy for the undirected two-commodity maximum flow problem." *Computational Optimization and Applications* 47 (2): 289–305.

Sedeño-Noda, A., C. Gonzalez-Martin, and J. Gutierrez. 2005. "The biobjective undirected two-commodity minimum cost flow problem." *European Journal of Operational Research* 164 (1): 89–103.

Risk Aversion and Retail Electricity Markets

A. Downward
Department of Engineering Science
University of Auckland, New Zealand
a.downward@auckland.ac.nz

Abstract

In 2009 Prof. Frank Wolak carried out a review of the New Zealand electricity market. In this report he noted that there is a lack of competition in the NZEM, particularly during dry years and suggested that competition could be improved by reallocating generation assets amongst the market participants. We have designed an equilibrium model that encompasses both the retail and wholesale electricity sectors. In the future, this model will enable us to gain insights into the effects of asset reallocation.

Key words: electricity markets, game theory, risk aversion, equilibrium.

1 Introduction

Over the past two years, there has been much discussion regarding the competitiveness of the New Zealand electricity market (NZEM). In May 2009, the Commerce Commission released a report by Stanford University's Professor Frank Wolak [9]. This report highlighted that firms in the wholesale market possess *transient* market power. That is, firms occasionally have the ability (and the incentive) to increase spot prices significantly above the underlying long-run marginal costs of their power plants. This generally occurs during times when hydro reservoir levels are low (meaning that the opportunity cost of water is high). The Wolak report also raised a concern that only one firm in the NZEM owns significant generation in both islands, which can lead to limited competition at times when the HVDC link is constrained. Wolak suggested that reallocating generation assets amongst the state-owned firms could reduce the ability of firms to exercise this market power, in the wholesale market.

Later in 2009, a discussion document from the Electricity Technical Advisory Group [2], agreed with Wolak's recommendation of an asset reallocation and suggested three alternatives. This report stated that this reallocation should improve competition in the retail market, by incentivising *gentailers* (firms with both generation and retail arms) to enter into new markets; thereby driving prices down for consumers.

In this paper, we investigate retail competition over a *competitive* wholesale market, and attempt to understand how risk aversion on the part of gentailers may affect their retail pricing decisions and their incentives to enter new markets. This paper is laid out as follows: in the next section, we outline how we model and measure risk on behalf of retailers; we then present our model of retail markets; next, we discuss the clearing of the wholesale market. A simple example is constructed to illustrate the impact of risk-aversion on retail pricing and finally a general model formulation is presented that can, in the future, be used to examine the impact of asset reallocation.

1.1 Risk

When companies are making decisions regarding investment or entering contracts they must not solely consider the expected benefit from the decisions, they must also take into account the consequences if the return on the investment is lower than expected – this is known as risk-aversion. Companies are responsible to shareholders over both the short and long term, so they need to make decisions that seek to minimize their risk, while maximizing expected profit. Mean-risk optimization is commonly used in portfolio optimization; this approach involves solving a bi-objective optimization problem and typically results in finding a set of Pareto optimal solutions, known as the *efficient frontier*. In such models risk is typically measured by the variance of the return or by using the downside-risk, as introduced by Markowitz [7].

Another common way of incorporating a risk-attitude into optimization problems is to use a utility function. If a business is risk-averse, a concave, increasing utility function will reflect this. That business can then maximize its expected utility (as a function of profit) rather than its expected profit; the optimal solution to this will have a lower (or equal) expected return that if the firm had been profit maximizing, but the risk associated with this decision will be less. However, determining a precise utility function is difficult.

In order to formalize the concept of risk, *coherent* measures of risk were introduced by Artzner et al. in [1]. Any coherent risk measure must comply with the following four axioms: subadditivity; translation invariance; positive homogeneity; and monotonicity. Risk measures complying with these axioms exhibit key properties that are valuable for a risk-averse agent. For example, subadditivity ensures that there is a risk-pooling effect: the sum of risks is greater than or equal to the risk of the sum.

Conditional value at risk (CVaR)[1] is a coherent risk measure. The CVaR at level β of some random profit π is simply the expected loss if one's interest were restricted to the lowest $100\beta\%$ of returns. If profits π are continuously distributed with some distribution function $F(\pi)$ and associated probability density function $f(\pi)$ then $\mathrm{CVaR}_\beta(\pi)$ can be written as:

$$\mathrm{CVaR}_\beta\left(\rho\right) = -\frac{1}{\beta}\int_{-\infty}^{F^{-1}(\beta)} \pi f\left(\pi\right) d\pi.$$

Initially this measure of risk may seem impractical for optimization problems. The above description requires the order of the profits to be known a priori, when in fact, in many circumstances, the ordering of outcomes often will depend on the decision variables being optimized over. Fortunately, however, Shapiro et al. [8] present an alternate

[1]This is all known as average value at risk, or expected shortfall.

formulation in which the bottom $100\beta\%$ of outcomes can be computed through a linear program. Moreover, for profit functions that are concave in the decision variables, we can maximize a weighted combination of expected profit and risk. In the formulation below, α is a parameter between 0 and 1 and changes the weightings on risk and return[2]:

$$\max \qquad E_{\pi}[\pi] - \frac{\alpha}{\beta} E_{\pi}\left[\max\left\{(1-\beta)(\eta-\pi), \beta(\pi-\eta)\right\}\right]. \tag{1}$$

In this paper, we use the above mean-risk formulation to create a model of an electricity retail market with risk-averse firms. We will initially compare the behaviour of firms, competing in the retail power sector, who are risk-neutral with those that are risk-averse. A risk-neutral firm is not concerned about the need to hedge against possible negative outcomes; they are solely interested in maximizing their expected profit. Conversely, risk-averse firms are concerned with mitigating the adverse affects of negative outcomes (such as, in electricity markets, plant outages or high fuel costs) by changing their exposure to the spot market by altering their expected retail positions.

1.2 Retail Competition

Modelling of competition in the retail sector of electricity markets requires an understanding of consumer behaviour. In the literature, many models of consumer behaviour have been analysed (see, e.g., [6, 5]). These models are typically based on price competition, whereby firms submit their prices to the market and consumers choose a generator which offers the best deal. Under the assumption of Bertrand competition ([4]), this would mean that all consumers will go to the firm offering the lowest price, however, this is not empirically observed. Hotelling models ([5]) build upon this basic principle of price competition, but here the consumers may have an initial preference to a firm or there may be additional cost unique to each consumer for each firm.[3] This creates some elasticity in the market, allowing firms to set their price above that of competing firms, but still retain some customers.

Another branch of work concerned with consumer behaviour are consumer switching models. Here consumers have costs of switching or searching out firms. These models may allow incumbent firms to overcharge consumers, due to the consumers' reluctance to seek out better deals; see, e.g, [6].

2 Model

In this section we introduce the model that we will use. It consists of three stages, in the first stage firms decide whether or not they wish to participate in the retail market. In the following stage, firms participating in the retail market choose retail prices which they set for an extended period (for example, one year). In the third stage of the model, the uncertain wholesale market clears. We will discuss the final two stages in more detail below, starting with the wholesale market.

[2] See the Appendix, or [8] for a derivation of this expression.

[3] For example, the distance a consumer must travel may influence a consumer to purchase from the more expensive retailer.

2.1 Wholesale Market

In this work, for simplicity, we assume that the wholesale market is *perfectly competitive*. This is the simplest assumption about the behaviour of the firms, since the offers are set to the plants' marginal costs. These offers are received by the system operator and an optimization problem (the *economic dispatch problem*) is solved to minimize the cost of satisfying the demand, yielding the optimal generation, power flows and nodal prices. These nodal prices are defined to be the cost of an additional unit of power at a node.

Although, at the time that the system operator solves the dispatch problem there is no uncertainty, when firms are setting their retail prices there may be uncertainty surrounding the costs of the firms (water value or gas costs) or capacities of plants and lines (due to outages).

2.2 Retail Competition

We use a differentiated products formulation to model the demand each firm receives as a function of their price; here we assume that the products that the firms offer are partial strategic substitutes. Thus if each firm i chooses a price p_i, we can compute their demand from the following function:

$$x_i = X_i + \sum_{j \neq i} f_{ij}\left(p_i, p_j\right),$$

where X_i is the default load if all firms charge the same price and $f_{ij}(p_i, p_j)$ is any increasing function satisfying the following conditions[4]:

$$\begin{aligned} \frac{\partial f_{ij}}{\partial p_i} &\leq 0, \\ \frac{\partial f_{ij}}{\partial p_j} &\geq 0, \\ \left.\frac{\partial f_{ij}}{\partial p_i}\right|_{p_i = p_i^*} &\leq \left.\frac{\partial f_{ji}}{\partial p_i}\right|_{p_i = p_i^*}. \end{aligned}$$

If we define $f_{ij}\left(p_j - p_i\right) = b\left(p_j - p_i\right)$ and $X_i = \frac{X}{n}$ then we have

$$x_i = \frac{X}{n} - b\left(n-1\right)p_i + b\sum_{j \neq i} p_j.$$

Note that in the above model the overall demand is inelastic; however, for the individual firms, demand is lost or gained based on the cross-elasticity constraint b and the price differences between firms. For ease of computation, we will restrict our interest to the linear demand model, above, for the remainder of this paper.

[4]These conditions ensure that as the price of a firm increases the demand for that firm decreases and the demand of other firms increases. Moreover, the number of customers leaving one firm must be less than or equal to the number arriving at another.

2.3 Risk-neutral firms

If firms are risk neutral, this means they aim to maximize their expected profit. We can observe this in the single node case below. The (expected) profit function for firm i is given by

$$\begin{aligned} \pi_i &= E_\omega\left[(p_i - c)\left(\frac{X}{n} - b(n-1)p_i + b\sum_{j\neq i} p_j\right) + P_i\right] \\ &= E_\omega\left[(p_i - c)\left(\frac{X}{n} - b(n-1)p_i + b\sum_{j\neq i} p_j\right)\right] + E_\omega[P_i] \\ &= (p_i - E_\omega[c])\left(b\sum_{j\neq i} p_j - b(n-1)p_i\right) + p_i\frac{E_\omega[X]}{n} - E_\omega\left[\frac{cX}{n}\right] + E_\omega[P_i], \end{aligned}$$

where P_i is firm i's profit from the wholesale market.

Now we wish to compute the equilibrium to the retail game; this is a point where no firm has a strategy that can unilaterally improve its expected profit. Note that each player has a smooth, concave profit function, thus we can find the maximum profit from the first order condition; differentiating the expected profit function with respect to p_i gives

$$\frac{\partial \pi_i}{\partial p_i} = \frac{E_\omega[X]}{n} - 2(n-1)bp_i + b\sum_{j\neq i} p_j + (n-1)bE_\omega[c].$$

Now we solve for price p_i^* yielding maximum profit:

$$\left.\frac{\partial \pi_i}{\partial p_i}\right|_{p_i = p_i^*} = 0 \Rightarrow p_i^* = \frac{\frac{E_\omega[X]}{n} + b\sum_{j\neq i} p_j + (n-1)bE_\omega[c]}{2(n-1)b} \tag{2}$$

We sum the above equation for all i, yielding

$$\sum_i p_i^* = \frac{E_\omega[X] + \sum_i p_i + n(n-1)bE_\omega[c]}{2(n-1)b},$$

which can be rearranged to give

$$\sum_i p_i^* = \frac{E_\omega[X]}{(n-1)b} + nE_\omega[c].$$

Finally we substitute the above expression into equation (2) to find the equilibrium prices:

$$p_i^e = \frac{E_\omega[X]}{n(n-1)b} + E_\omega[c],$$

For scenario ω the retail demand that the firm gets is:

$$x_i^* = \frac{X_\omega}{n}.$$

In other words, we find that the firms all choose identical prices for their retail sales, and each firm gets an equal market share. This is because the profit from the wholesale market is not affected by the decisions made in the retail market, and is merely a (uncertain) constant term in the profit function. In a risk neutral setting, the expected value of this wholesale profit can be separated from the retail profit, so it plays no part in the optimization problem. On the surface this may seem peculiar, but optimizing expected profits cannot distinguish between opportunity costs and actual costs, so all firms have the same incentives. We illustrate this by way of an example in the next section.

2.3.1 Single-node Example

Consider a situation with one node having a totally inelastic demand of 150MW (all retail). In the market there are two firms, each with a retail base: firm A owns a hydro plant with capacity 100MW and cost $\$h$/MWh and firm B owns a thermal plant with capacity 100MW and cost \$50/MWh. The hydro costs are uncertain when the retail prices are chosen, however, it is common knowledge that the hydro costs, h are distributed uniformly from 0 to 100.

First, we solve for the optimal dispatch as a function of h. This gives the following dispatch quantities, q_A, q_B for each plant and clearing price, c:

$$\begin{aligned}
q_A &= \begin{cases} 100, & h \leq 50, \\ 50, & h > 50, \end{cases} \\
q_B &= \begin{cases} 50, & h \leq 50, \\ 100, & h > 50, \end{cases} \\
c &= \begin{cases} 50, & h \leq 50, \\ h, & h > 50. \end{cases}
\end{aligned}$$

Now we can compute the expected profits of the firms as below:

$$\begin{aligned}
E\left[\pi_A\right] &= E\left[q_A \times (c-h) + d_A\left(p_A - c\right)\right] \\
&= \tfrac{1}{100}\int_0^{50} 100\left(50-h\right)dh + d_A\int_0^{50}\left(p_A - 50\right)dh + d_A\int_{50}^{100}\left(p_A - h\right)dh \\
&= 1250 + d_A\left(p_A - 62.5\right).
\end{aligned}$$

Substituting in d_A (with $b = 1$) and differentiating with respect to p_A gives:

$$\frac{\partial E\left[\pi_A\right]}{\partial p_A} = 75 - 2p_A + p_B + 62.5.$$

Similarly for firm B we find

$$\frac{\partial E\left[\pi_B\right]}{\partial p_B} = 75 - 2p_B + p_A + 62.5.$$

Assuming an interior solution, these first order conditions yield the following equilibrium prices:

$$p_A^e = p_B^e = 137.5.$$

2.4 Risk-averse firms

In a risk-averse setting, however, this symmetry is lost. This is because the outcome of the wholesale market cannot be merely treated as a constant in the risk-adjusted profit maximization problem:

$$\rho_{\alpha,\beta}\left[\pi_i\right] = (1-\alpha)E\left[\pi_i\right] - \alpha\mathrm{CVaR}_\beta\left[\pi_i\right].$$

In the next example, we will demonstrate how risk aversion alters the optimal decisions of the firms.

2.4.1 Single-node Example

For this example, we will define the CVaR risk level, β=0.1, examine the affect of varying α on the behaviour of the firms. In order to compute the conditional value at risk, we need to rank the profit for each firm as a function of h for each retail price is may choose. The revenue for a given h and p_A for firm A is:

$$\pi_A = \begin{cases} \left(p_A - 50\right)\left(75 - p_A + p_B\right) + 100\left(50 - h\right), & h \leq 50, \\ \left(p_A - h\right)\left(75 - p_A + p_B\right), & h > 50. \end{cases}$$

Note that, from the above equation, we can see that so long as the the retail demand is positive the profit of firm A is decreasing in h. The revenue for a given h and p_B for firm B is:

$$\pi_B = \begin{cases} \left(p_B - 50\right)\left(75 - p_B + p_A\right), & h \leq 50, \\ \left(p_B - h\right)\left(75 - p_B + p_A\right) + 100\left(h - 50\right), & h > 50. \end{cases}$$

From above we can see that firm B's profit is constant over $h \in [0, 50]$, however for $h > 50$ we have:

$$\frac{\partial \pi_B}{\partial h} = 25 + p_B - p_A,$$

hence if $p_A - p_B \geq 25$ then the profit is decreasing in h, whereas if $p_A - p_B \leq 25$ the profit is increasing in h; it can be shown that the former case is not possible at equilibrium, hence we can write the risk-adjusted profit functions for both firms:

$$\begin{aligned} \rho_{\alpha_A,0.1}\left[\pi_A\right] &= \frac{1-\alpha_A}{100}\int_0^{100}\pi_A\left(h\right)dh + \frac{\alpha_A}{10}\int_{90}^{100}\pi_A\left(h\right)dh \\ &= \left(1-\alpha_A\right)\left[\left(p_A - 62.5\right)\left(75 + p_B - p_A\right) + 1250\right] + \alpha_A\left[\left(p_A - 95\right)\left(75 + p_B - p_A\right)\right], \end{aligned}$$

$$\begin{aligned} \rho_{\alpha_B,0.1}\left[\pi_B\right] &= \frac{1-\alpha_B}{100}\int_0^{100}\pi_B\left(h\right)dh + \frac{\alpha_B}{10}\int_0^{10}\pi_B\left(h\right)dh \\ &= \left(1-\alpha_B\right)\left[\left(p_B - 62.5\right)\left(75 + p_A - p_B\right) + 1250\right] + \alpha_B\left[\left(p_B - 5\right)\left(75 + p_A - p_B\right)\right]. \end{aligned}$$

We can then solve for the equilibrium, from first order conditions, giving:

$$p_A^e = \frac{5}{6}\left(165 + 26\alpha_A - 23\alpha_B\right), \quad p_B^e = \frac{5}{6}\left(165 + 13\alpha_A - 46\alpha_B\right).$$

Note that if we set $\alpha_A = \alpha_B = 0$ then we recover the risk-neutral equilibrium. Moreover, as firm A's risk-aversion weighting is increased the equilibrium prices of both players increase; whereas as firm B's risk-aversion weighting is increased the equilibrium prices of both players decrease. This result is due to the wholesale market positions of the firms. Firm A, who owns hydro generation receives the least profit when hydro costs are high, thus a risk-averse firm emphasizes these occurrences and when optimizing in the retail market sees a higher risk-adjusted 'average' spot price. Conversely, firm B, who owns thermal generation receives the least profit when hydro costs are low, thus when optimizing its retail price sees a lower risk-adjusted 'average' spot price. Furthermore, the effects of competition mean that the risk-aversion of one firm will affect the other's optimal pricing strategy.

This example has illustrated how risk-aversion can affect the retail equilibrium prices. In the next section we formulate the retail problem more generally and discuss the solution techniques.

2.4.2 General formulation

Now let us define the price-competition amongst firms $j \in \mathcal{F}$. The firms are able to compete for retail customers at any node $i \in \mathcal{N}$ in the network. However, there is uncertainty about future demand levels and nodal prices. The different scenarios are indexed by $s \in \mathcal{S}$.

Each firm j solves the following optimization problem to optimize its risk-adjusted profit function.

Parameters:

- c_{si} is the wholesale price in scenario s at node i;
- A_{ij} is a boolean matrix specifying whether a firm i competes in the retail market at node i;
- X_{sij} is the retail demand for firm j at node i in scenario s when all retailers offer the same price.
- p_{ik} is the retail price offered by firm $k(\neq j)$ at node i.
- pr is the probability of each scenario (assuming all scenarios are equally likely).

Variables:

- η_j is the β-quantile profit of firm j;
- v_{sj} is the positive deviation from the β-quantile profit of firm j;
- w_{sj} is the negative deviation from the β-quantile profit of firm j;
- d_{sij} is the realised retail demand for firm j at node i in scenario s;
- Z_{sj} is the profit for firm j in scenario s;
- p_{ij} is the retail price offered by firm j at node i.

Formulation:

$$\begin{array}{lll}
\max & \sum\limits_s pr \times Z_{sj} - \frac{1}{\beta}\sum\limits_s pr\left[(1-\beta)\, w_{sj} + \beta v_{sj}\right] & \\
\text{s.t.} & Z_{sj} = \eta_j + v_{sj} - w_{sj} & [\lambda_{sj}] \\
& Z_{sj} = \sum\limits_i A_{ij}\,(p_{ij} - c_{si})\, d_{sij} + R_{sj} & [\mu_{sj}] \\
& d_{sij} = X_{sij} + b\sum\limits_k A_{ik}\,(p_{ik} - p_{ij}) & [\nu_{sij}] \\
& w_{sj}, v_{sj} \geq 0. &
\end{array}$$

The above problem is convex in the decisions variables p_{ij}, $\forall i \in \mathcal{N}$, therefore KKT conditions are equivalent. Below we hold the KKT conditions of all firms $j \in \mathcal{F}$ simultaneously.

$$\begin{array}{ll}
Z_{sj} = \eta_j + v_{sj} - w_{sj} & \forall s \in \mathcal{S},\, \forall j \in \mathcal{F}, \\
Z_{sj} = \sum\limits_i A_{ij}\,(p_{ij} - c_{si})\, d_{sij} + R_j & \forall s \in \mathcal{S},\, \forall j \in \mathcal{F}, \\
d_{sij} = X_{sij} + b\sum\limits_k A_{ij}\,(p_{ik} - p_{ij}) & \forall s \in \mathcal{S},\, \forall i \in \mathcal{N},\, \forall j \in \mathcal{F}, \\
-pr - \lambda_{sj} - \mu_{sj} = 0 & \forall s \in \mathcal{S},\, \forall j \in \mathcal{F}, \\
A_{ij}\,(p_{ij} - c_{si})\,\mu_{sj} - \nu_{sij} = 0 & \forall s \in \mathcal{S},\, \forall i \in \mathcal{N},\, \forall j \in \mathcal{F}, \\
\sum\limits_s \left[A_{ij} d_{sij}\mu_{sj} + b\sum\limits_k A_{ik}\,(-\nu_{sij}) + b\nu_{sij}\right] & \forall i \in \mathcal{N},\, \forall j \in \mathcal{F}, \\
\sum\limits_s \lambda_{sj} = 0 & \forall j \in \mathcal{F}, \\
\lambda_{sj} + pr \geq 0 & \forall s \in \mathcal{S},\, \forall j \in \mathcal{F}, \\
-\lambda_{sj} + \frac{1-\beta}{\beta} pr \geq 0 & \forall s \in \mathcal{S},\, \forall j \in \mathcal{F}.
\end{array}$$

This system of equations is known as a mixed complementarity problem; this type of problem can be solved in `GAMS` using the `PATH` solver [3]. We will not present any models of the New Zealand system here; they will be available in a future paper.

3 Conclusions

Motivated by Prof. Frank Wolak's report on the NZEM and the Ministerial Review into the electricity industry, we have constructed a model which encompasses both the retail and wholesale markets. We first observe that with risk-neutral firms and a competitive wholesale market, retailers will reach a symmetric equilibrium. We then examine how their behaviour may change using a simple one-node example with two firms. We find that risk-aversion does not always mean that a premium is passed onto consumers, in fact, retail prices may drop as a firm becomes more risk-averse (this effect is dependent upon the particular circumstances of the retailers).

We finish this paper with a model formulation for computing an retail market equilibrium with arbitrary sets of firms, nodes and wholesale market scenarios.

Acknowledgments

I would like to thank Dr. Golbon Zakeri and Dr. David Young for their discussions and assistance with this work.

Appendix

Conditional value at risk as a mean–risk measure

The following function of the random variable Z gives the expected weighted deviation from any given quantile β

$$r_\beta[Z] = \min_\eta \{\varphi(\eta) := E[\max\{(1-\beta)(\eta - Z), \beta(Z-\eta)\}]\}.$$

To see that the optimal η does in fact correspond to the β-quantile, we first take left and right derivatives of the above function with respect to η:

$$\begin{aligned} \varphi'(\eta)_+ &= \Pr[Z \le \eta] \times (1-\beta) - \Pr[Z > \eta] \times \beta \ge 0, \\ \varphi'(\eta)_- &= \Pr[Z < \eta] \times (1-\beta) - \Pr[Z \ge \eta] \times \beta \le 0. \end{aligned}$$

Observe that at the optimal η the left derivative must be non-increasing and the right derivative must be non-decreasing.[5] We can then rearrange the above inequalities to find:

$$\Pr[Z < \eta] \le \beta \le \Pr[Z \le \eta],$$

which confirms that the optimal η is the β-quantile.

Hence if $F(\alpha)$ is the cumulative distribution function and $p(\alpha)$ is the probably distribution function corresponding to the random variable $Z(\alpha)$, then

$$\begin{aligned} r_\beta[Z] &= \int_0^\beta p(\alpha)(1-\beta)\left(F^{-1}(\beta) - Z(\alpha)\right)d\alpha + \int_\beta^1 p(\alpha)\beta\left(Z(\alpha) - F^{-1}(\beta)\right)d\alpha \\ &= \int_0^\beta p(\alpha)(1-\beta)F^{-1}(\beta)\,d\alpha - \int_0^\beta Z(\alpha)p(\alpha)(1-\beta)\,d\alpha + \int_\beta^1 Z(\alpha)p(\alpha)\beta d\alpha - \int_\beta^1 F^{-1}(\beta)p(\alpha)\beta d\alpha \\ &= \int_\beta^1 Z(\alpha)p(\alpha)\beta d\alpha - \int_0^\beta Z(\alpha)p(\alpha)(1-\beta)\,d\alpha \\ &= (1-\beta)\beta\left(E_{Z\ge Z(\beta)}[Z] - E_{Z\le Z(\beta)}[Z]\right). \end{aligned}$$

Moreover, note that the expectation of Z can be written as:

$$E[Z] = \beta E_{Z\le Z(\beta)}[Z] + (1-\beta)E_{Z\ge Z(\beta)}[Z].$$

Finally, we find that

$$\begin{aligned} E[Z] - \frac{1}{\beta}r_\beta[Z] &= \beta E_{Z\le Z(\beta)}[Z] + (1-\beta)E_{Z\ge Z(\beta)}[Z] - (1-\beta)\left(E_{Z\ge Z(\beta)}[Z] - E_{Z\le Z(\beta)}[Z]\right) \\ &= \beta E_{Z\le Z(\beta)}[Z] + (1-\beta)E_{Z\le Z(\beta)}[Z] \\ &= E_{Z\le Z(\beta)}[Z] = \mathrm{CVaR}_\beta(Z). \end{aligned}$$

This can be incorporated into a mean-risk optimization problem with a parameter $\alpha \in [0,1]$ controlling the weightings on risk versus mean return.

$$\begin{aligned} (1-\alpha)E[Z] - \alpha\mathrm{CVaR}_\beta(Z) &= (1-\alpha)E[Z] + \alpha\left(E[Z] - \frac{1}{\beta}r_\beta[Z]\right) \\ &= E[Z] - \frac{\alpha}{\beta}r_\beta[Z]. \end{aligned}$$

[5] Of course this assumes that φ is continuous in η, which can be easily verified.

References

[1] P. Artzner, F. Delbaen, J.-M. Eber and D. Heath. Coherent measures of risk. *Math. Finance* 9 (1999), pp. 203-228.

[2] Electricity Technical Advisory Group. Improving Electricity Market Performance. A preliminary report to the Ministerial Review of Electricity Market Performance (2009).

[3] M. C. Ferris and T. S. Munson. Complementarity problems in GAMS and the PATH solver. Journal of Economic Dynamics and Control, 24(2) (2000), pp. 165 - 188.

[4] D. Fudenberg, J. Tirole. *Game theory.* Cambridge: MIT Press (1991).

[5] H. Hotelling. Stability in competition. *Economic Journal*, 39(153) (1929), pp. 41-57.

[6] P. Klemperer. Markets with Consumer Switching Costs. *The Quarterly Journal of Economics*, 102, No. 2 (1987), pp. 375–394.

[7] H. Markowitz. *Portfolio Selection*, John Wiley & Sons: New York (1959).

[8] A. Shapiro, D. Dentcheva and A. Ruszczynski. *Lectures of Stochastic Programming: Modeling and Theory*, chapter 6. Society for Industrial and Applied Mathematics, Mathematical Programming Society, Philadelphia (2009).

[9] F. Wolak. An Assessment of the Performance of the New Zealand Wholesale Electricity Market. Report for the New Zealand Commerce Commission (2009).

Virtual Smelter Modelling for Metal Flow Management

Timothy W. Harton
Department of Engingeering Science
University of Auckland
New Zealand
tim@harton.co.nz

Abstract

The production of aluminium is an extremely expensive process because of the vast amounts of energy it uses. It is also a continuous, large scale process, thus it is very hard to experiment with. Metal flow management provides an opportunity to increase revenue without altering the base processes which produce the metal. Metal flow management means utilising the smelter's resources as efficiently as possible to produce the most profitable types of aluminium possible.

Metal flow through a smelter is extremely complex with many interactions which are difficult to model analytically. However this can be overcome by utilising simulation. This enables us to experiment with a smelter for very little cost. It also allows us to analyse the complex interactions of different parts of the metal flow process so that we understand the process better, thus we are be able to produce better optimisations for aluminium smelters. In this paper we describe how we developed what we believe is the first metal flow simulation for an aluminium smelter which utilises optimisation at several stages of the process and the results we obtained.

Key Words: Aluminium, alumina, metal, crucible, smelter, cell, furnace, cast house, batching, scheduling, VSmelter, LMRC (Light Metals Research Centre), metal flow, metal flow management.

1 Introduction

To model the flow of aluminium through an aluminium smelter it is important that we understand what aluminium is and how it is produced.

Aluminium is light metal produced in a smelter from aluminium oxide, commonly referred to as alumina. Bauxite is the primary ore for alumina; this means that alumina is obtained from the mining of bauxite ore. Bauxite comprises of many different minerals such as gibbsite ($Al(OF)_3$), along with some impurities. Bauxite is purified by the Bayer process and alumina is produced.

In an aluminium smelter the metal is produced using the Hall-Héroult process, this is a process of electrolytic reduction of alumina into aluminium. The alumina is dissolved into molten cryolite (a rare mineral) where it undergoes electrolytic reduction to obtain pure aluminium. This process takes place in reduction cells constructed out of steel and refractory bricks. The process is extremely energy intensive; a direct current of 150,000 to 250,000 Amps, depending on the type of cell is passed between an anode and a cathode within a bath of hot cryolite (around 960 ˚C).

Aluminium smelting is a major industrial process. The reduction part of the process consists of multiple pot rooms which in turn consist of multiple cell lines. Each cell line contains anywhere from 50 to 150 cells. The cells are tapped once a day to remove the produced aluminium.

The tapped metal then undergoes firing in the furnace where its chemical composition is altered to increase its value for sale. The aluminium will then undergo casting to create products for sale on the global market. The primary products are ingots and billets of different quality aluminium. These are then used as raw material in the production of goods which utilise aluminium, such as nails, bikes, cars and even airplanes.

The nature of the world aluminium market is such that as the purity increases, the price obtained for the metal increases exponentially (Piehl, 2000). Adding additional elements to form an alloy can also have a huge impact on the price of the metal. This means in certain situations it is better to produce small quantities of expensive metal than to produce large quantities of inexpensive metal. A decision must be made about how much of each type of aluminium should be produced in order to maximise profit.

Metal flow is the passage of metal through an aluminium smelter from its raw form as alumina casted products such as ingots. Managing metal flow within the smelter is critical because it will greatly affect its price in cast form. Optimal metal flow management is a necessity for an aluminium smelter to function at its peak efficiency.

Simulation is needed to model metal flow. This is because currently the only test to determine the effect of metal flow management decisions is to apply them to a smelter..

The different areas of metal flow management involve separate processes but the decisions that are made are intertwined and therefore have upstream and downstream effects on what and how the metal in the smelter is being produced.The simulation of a smelter will allow us to identify whether the optimisations are working in the real world and how they affect the smelter's overall efficiency.

There have been mathematical models developed of aluminium smelters previously (Duncan & Nicholls, 1993). These models have not been able to model the complex real world constraints. The only way to truly do this is through an aluminium smelter simulation.

Simulations and optimisations can also be used together for mutual improvement. As the simulation improves the feedback it allows the improvement of optimisations and vice versa.

2 Metal Flow

This section describes metal flow through an aluminium smelter. Before simulation or optimisation this flow has to be understood. This flow should be understood because if the flow isn't simulated improperly it will lead to invalid results. It is also necessary to understand metal flow so we know how to implement the optimisations within a smelter. To do this we have to understand how decisions are currently being made.

Metal flow though an aluminium smelter is a continuous process. Keeping the flow of metal through the smelter constant enables the production of high quality metal. This is because reduction within the cells is a carefully balanced process. Production can't be ramped up or slowed down without affecting the quality of the metal being produced. This is because it would affect the bath chemistry of the cell causing the process to become unbalanced. In addition if molten aluminium is left too long below its melting point it becomes solid thus ruining the vessel that contains it.

2.1 Feeding

This is the process of adding alumina into a cell at a rate which tries to maintain constant bath chemistry. The cells are being tapped at a rate of once per day. The quality of the alumina affects the purity of the aluminium being produced but other factors also affect aluminium quality; these include the current flowing into the cell (whether the current is constant) and the quality of the carbon in the anode. These factors are largely determined by the manner in which the cell has been operated during its life. The composition of the metal within a cell can also be affected by impurities such as iron, silicon and gallium. The same alumina is typically fed into every cell throughout a cell line so this will not influence the variation in purity between cells.

2.2 Cell Batching

The cells in a cell line are tapped once a day. Tapping is the process of extracting metal from a cell and loading it into a crucible . Smelter workers prefer to tap cells sequentially in batches (groups of 2 or 3 cells) to fill a crucible.

The choice of which cells are put into the same crucible affects the purity of the aluminium the crucible contains. If the cells are tapped sequentially the composition of the metal which fills the crucible has not been optimised and is effectively random. In this report the metal contained within the crucible is called a cell batch.

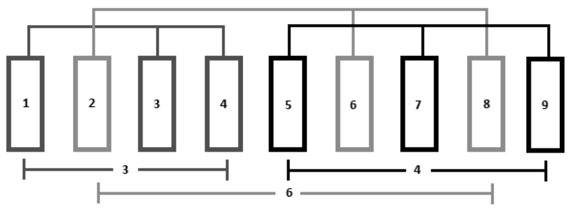


Figure 1, Cell Batch Spread

The spread of the cells in a batch is what defines the distance travelled when the workers are tapping.

2.3 Furnace Batching

Cell batches arrive at the furnace in crucibles, they are then completely emptied into the furnace. A furnace batch is the metal created when cell batches are mixed in the furnace. The process where the furnace is filled by these cell batches is called charging. The choice of cell batches within a furnace batch affects the purity and therefore the price which can be obtained for the metal produced by the furnace batch.

The furnace batch is then heated and alloying elements may be added to create the requisite metal type. The process of heating the metal within the furnace is called settling. The metal is then poured from the furnace into loads to be used in the cast house. Furnace batches are emptied from the furnace in lots called furnace pours. It is impossible to empty a furnace completely; a small amount of metal is left which will contaminate the next furnace batch.The small amount of metal left in a furnace after it is poured is called the heel.

2.4 Cast House

The cast house receives pours of metal from the furnace. From this the cast house produces products such as billets, ingots and cast rods. The products that a cast house can produce are determined by the machinery that it contains. The processes which take place within the cast house are casting, rolling, and extruding.

3 Literature Review

The first approach to the cell batching optimisation problem was described in Tuck (1997). The cell batching problem was applied to the New Zealand Aluminium Smelter (NZAS) at Tiwai point. Ryan (1998) states that the cell batching optimisation can be formulated naturally as a set partitioning problem (SPP). However Ryan (1998) does not take into account that this is part of a larger smelter scheduling problem which he states in his conclusion.

Ryan (1998) and Piehl (2000), who worked on a similar problem, both make the assumption that the tapping weight across a cell line is the same from every cell. In reality the planned tapping weight for every cell in a cell line is one of a set of maybe four or five choices depending on the cell’s condition. For example the planned tapping weights could either be 0 kg, 1500 kg, 1800 kg, 2000 kg or 2200 kg. The composition of the metal in a cell batch could vary significantly from predictions if the tapping weights of each cell were assumed to be the same. The actual weight of the metal tapped from the cell is never exactly that planned, rather it is normally distributed around the plan. This can only be shown using a simulation or a stochastic solution.

Ryan (1998) and Piehl (2000) both use column generation to limit the spread of the potential tapped batches in the cell batching optimisation problem. Ryan (1998) has a spread limit of 6 cells which can be violated a small number of times. When this happens the spread is limited to 10 cells. This reduces the problem size from 20825 variables to 3000 variables, Ryan (1998). The cell batching optimisation used in the Virtual Smelter does not use this spread limiting constraint instead only using a priori variable generation.

Tuck (1997), Ryan (1998) and Piehl (2000) all use the same alloy codes to obtain the premiums for the metal produced by each cell and each cell batch. Pascal Lavoie of the LMRC (Light Metals Research Centre) has advised that these codes are similar enough to the present situation and thus can be used for valuing the metal produced (Lavoie, 2010).

As mentioned earlier research in developing a mathematical model for an aluminium smelter has occurred (Duncan & Nicholls, 1993) (Nicholls, 1997). The models developed had different levels of aggregation as do the optimisation models put forward in this report (Nicholls, 1997). These papers mostly investigated the level of linking between sub models to develop one core model. Nicholls (1995) shows the core integrated model to be a nonlinear bi-level problem. The simulation utilised within a virtual smelter will allow for an investigation into the value of the improvement arising from this.

4 Simulation

The simulation, the optimisations and the supervisory framework are all part of VSmelter, a program created to simulate an aluminium smelter. Python was used as a base language because the simulation package SimPy and optimisation package PuLP were both available for python so could be easily linked. SimPy is a package available for python which easily allows creation of a discrete event simulation. The simulation’s purpose is to model metal flow through a smelter based on the metal composition at each point within the smelter.

4.1 Recovery Heuristics

Recovery heuristics are simple decisions initially put into the supervisory framework which enable the smelter to run without any optimisation. They will be employed if an optimisation fails within the simulation. This simulates reality where if an optimisation fails the smelter continues to function.

There are four areas of decision making within the simulation:

- Providing a list of cells to be tapped in their batches to the cell line
- Determining which furnace to send a crucible to
- Determining when to start the furnace settling and stop receiving crucible.
- Determining which cast house to provide with the pours from a furnace.

The recovery heuristics implemented initially to make sure the simulation ran:

- Tapping cell lines in sequential batches i.e. (1,2,3) , (4,5,6).
- The crucible will be sent to the first furnace found to be charging.
- Stop the furnace loading when it reaches a set mass.
- The furnace pour will be sent to the first cast house determined to be free.

5 Optimisation

It is possible to perform optimisations at key decision areas within the metal flow of a smelter. These key areas include cell batching, furnace batching, furnace scheduling, cast house scheduling, and the usage of scrap or recycled metal, metal storage and sales and marketing The metal flow model is extremely complex which means it is extremely hard to model as whole, but it is possible to model these decisions individually. The metal flow model as whole is nonlinear (Nicholls, 1995) and solving it would be far too time consuming. However aluminium smelters might not even be implementing local optimisations so these implementations should be modelled within VSmelter to prove their viability.

5.1 Cell Batching

The batching problem has been researched and applied to NZAS. For this reason it was the first optimisation applied to the VSmelter. Cell batching has been applied to a smelter and is still in use therefore the conclusion can be reached that it has been a profitable implementation. Thus we would expect that applying the cell batching SPP (Set Partitioning Problem) to VSmelter should increase the revenue generated.

Indices

i = Cells

j = Cell Batches

Parameters

a_{ij} = 1, if cell i is included in cell batch j
0, otherwise

r_j = Represents the revenue of cell batch j, which is the premium of the alloy in the possible batch multiplied by the mass

n = Total number of possible cell batches

c = The number of cells in the cell line

Decision variables

x_j = 1, if cell batch j is selected
0, otherwise

Model Cell Batching

$$\text{Maximise} \quad z = \sum_{j=1}^{n} r_j x_j$$

$$(1) \quad \sum_{j=1}^{c} a_{ij} x_j = 1 \quad for \;\; i = 1, 2, \dots, n$$

$$(2) \quad \sum_{j=1}^{n} x_j \le \lceil c/3 \rceil$$

$$x_j \in \{0,1\}$$

Explanation

The objective is to maximise the revenue. Constraint (1) limits one cell to one cell batch. Constraint (2) sets a maximum number of batches.

This is a pure set partitioning problem; however we limit the spread to a maximum number of cells within the batches by generating the variables a priori. The spread is limited to make tapping easier for the staff at a smelter. Because the spread is limited during a priori variable generation the spread is an implicit constraint. This means that a spread constraint is not added to the problems' formulation.

At this point the value of the simulation presents itself. The simulation can be used to analyse how changing the spread constraints affect the revenue of the smelter, the amount of distance travelled and the quantity of work done by the staff. Changing the maximum spread can be experimented with in the same way. The simulation could be used to determine whether the expense in time and staff of tapping extra batches is justified by the extra revenue produced.

5.2 Furnace Batching

This model defines a group of cell batches a priori then combines these into furnace batches to maximise revenue. The cell batches can be defined using any optimisation or even a simple heuristic. This will allow an analysis to be carried out to determine if cell batching before furnace batching provides additional revenue over and above using sequentially tapped groups. This optimisation will produce metal with a higher premium then creating furnace batches without any guidance.

The spread within a furnace batch is defined by the difference of the position of lowest and highest positioned batch, where the position of a batch is defined by the position of the lowest cell in that batch. This models a time constraint ensuring that furnace batches don't take too long to be collected, so that the furnaces run effectively with minimal wait times. The spread has a direct correlation to the work done when tapping the cell batches. If the spread is too large the workers on a line will have to travel the whole line to gather the taps, this takes extra time and is not efficient.

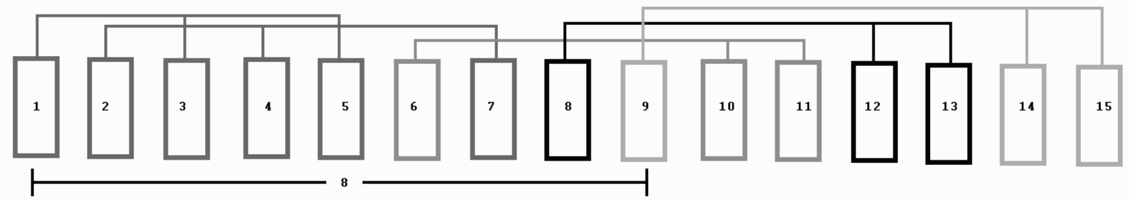


Figure 2, Furnace Batch Spread

As with cell batching the change in spread used in simulation can be used to analyse the extra time and work taken to collect the furnace batch. This allows experimentation, greater revenues may occur when a larger spread in furnace batching is used. As with in cell batching, variable generation a priori is used in this form of furnace batching to limit the spread of the furnace batches. This also has the effect of limiting the number of possible furnace batches and therefore significantly reduces the time required to generate these. This form of furnace batching was implemented.

Indices

$j =$ Cell Batches
$k =$ Furnace Batches

Parameters

$b_{jk} =$ 1, if cell batch j is included in furnace batch k
0, otherwise
$r_k =$ Represents the revenue of furnace batch k, which is the premium of the alloy in the possible batch multiplied by the mass
$m =$ Total number of possible furnace batches
$h =$ The number of cell batches
$f =$ Maximum number of furnace batches allowed per day

Decision variables

$y_k =$ 1, if furnace batch k is selected
0, otherwise

Model Furnace Batching

$$\text{Maximise } z = \sum_{k=1}^{m} r_k y_k$$

$$(3) \sum_{k=1}^{h} b_{jk} y_k = 1 \quad for\ j = 1,2,\ldots,m$$

$$(4) \sum_{k=1}^{m} y_k \leq f$$

$$y_k \in \{0,1\}$$

Explanation

The objective is to maximise the revenue. Constraint (1) limits one cell batch to one furnace batch. Constraint (2) sets a maximum number of furnace batches.

6 Results

This section will discuss the results of all the VSmelter simulation. The simulations were run under various conditions. There are three metal flow management decision areas currently able to be optimised within the supervisory framework under VSmelter:

6.1 VSmelter

A typical smelter was modelled within the VSmelter simulation framework. This smelter has two cell lines of 80 cells, 20 crucibles, 2 crucible trucks, 2 furnaces and 2 cast houses.

The cell batching heuristics including the optimisation limits the spread of the cell batches to a maximum of 9 cells. The furnace batching heuristics including the optimisation limits the spread of the furnace batches to a maximum of 25 cells.

Each situation was simulated 10 times for 20 days with the first 5 days of the simulation removed as data points, so as to remove the initial transient effects of the simulation. 150 data points were collected for each situation simulated.

6.2 Batching

Figure 3 below shows the effect of both furnace and cell batching on the average metal premium against a base of a smelter run on recovery heurisitcs.

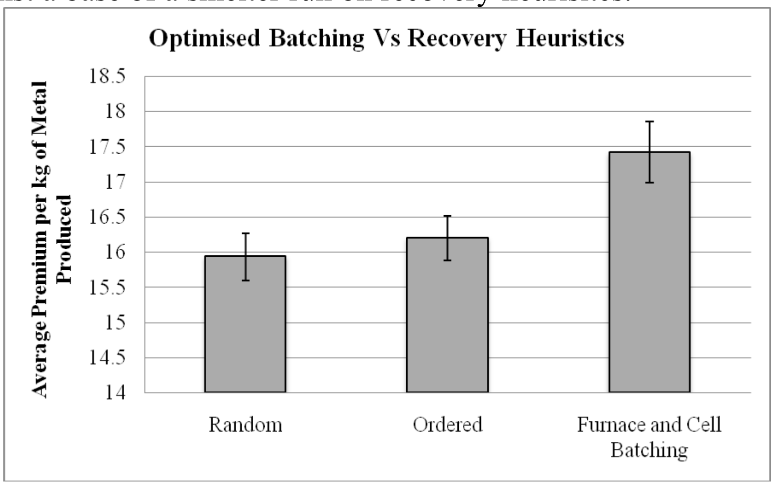


Figure 3, Effect of Furnace and Cell Batching

Utilising furnace and cell batching will yield a higher average premium than both the random and ordered situation The difference in the mean metal premium between the ordered and optimised situations is at least \$0.71 per kg and a maximum of \$1.71.

6.3 Furnace Scheduling.

Figure 4 below shows the effects of improved furnace scheduling heuristics on the average metal premium of the end product aluminium.

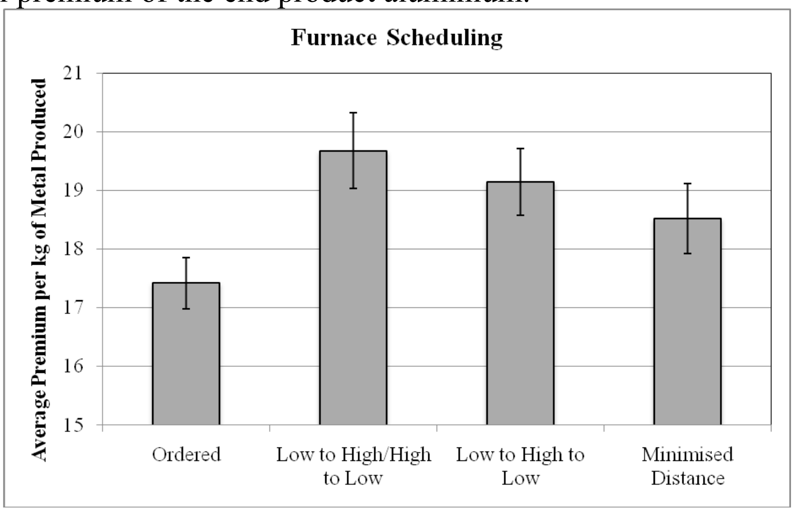


Figure 4, Effect of Furnace Scheduling

The graph and table above show that all the heuristics that aimed to improve the metal premium did. The best heuristic for ordering the furnace schedule appears to be Low to High/High to Low with an increase in the metal premium of at least $1.53 to a maximum of $2.99 per kg.

6.4 Value of the Model

Figure 5 below shows the effects of optimisation on the metal flow process and the difference in the optimisation models predictions versus what actually happened within the simulation.

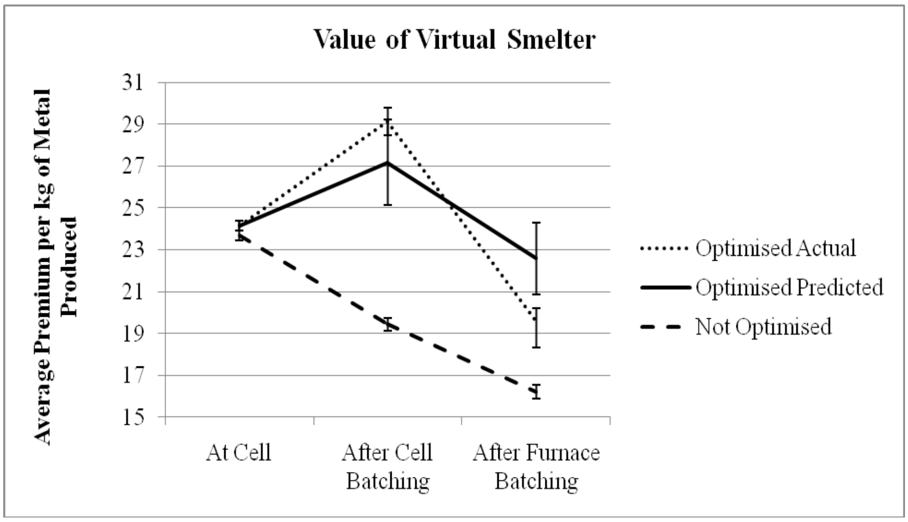


Figure 5, Metal Flow Value

The results show that a generic smelter should optimise its cell batching, furnace batching and schedule its furnaces so that they run from low to high purity and then high to low purity on alternate days. This changes the range for the mean daily production value from $3,094,650-$3,221,400 (under recovery heuristics) to $3,710,850-$3,962,400. This is a 20% increase in daily revenue. Over a year this could generate an additional $250 million of revenue.

The difference between the prodicted and actual lines show the value of the simulation as the optimisation models cannot be used to predict the value of the end product. This is because there is a significant difference the two values after the metal flow management decisions have been made.

7 Conclusions

The first simulation of metal flow through an entire aluminium smelter which incoroporated metal managmenent optimisations was created and run to analyse metal flow. Optimisations were then applied to this virtual smelter and analysed using an independent supervisory framework.

VSmelter has met its first goal. It is able to model a simple smelter and apply metal flow optimisation decisions as the simulation runs. The optimisations within VSmelter are simple but there is no reason that they can't become more complicated as research is continued into each area. VSmelter currently contains a simple simulation which is not totally realistic but is complicated enough for the analysis performed.

The simulation framework within VSmelter is able to simulate the main parts of aluminium smelter. It is also modular so the simulation's physical constraints can be changed. This is the first step to quickly being able develop simulation to simulate any smelter quickly.

Within the simple smelter modelled within VSmelter the most effective decisions heuristics were; optimised cell batching, optimised furnace batching, with a furnace schedule which ordered the batches from lowest purity to highest purity of aluminium, then from lowest to highest on alternate days.

The simple simulation within VSmelter has been used to evaluate the effect of metal flow decisions on the average metal premium, but there is no reason that the simulation could not be used to make a smelter in other areas such as energy expenditure from heat loss and input, or just utilising time and machinery more efficiently to increase production.

8 Future Work

VSmelter has immense potential to be extremely beneficial to the LMRC and the aluminium industry. As it becomes more complicated it also becomes far more useful allowing us to see in detail what effects metal flow management decisions have. The simulation could be expanded and applied to a real smelter to analyse the affects of metal flow management decisions and validate both the simulation and the optimisation models.

9 References

Duncan, H. J., & Nicholls, M. G. (1993). The Development of an Integrated Mathematical Model of an Aluminium Smelter. *The Journal of the Operationa Research Society* , 225-235.

Glenberg, A. M., & Andrzejewski, M. E. (2008). Learning from Data. Lawrence Erlbaum Associates.

Jensson, P., Kristinsdottir, B. P., & Gunnarsson, H. P. (2003). *Optimal Sequencing of tasks in an aluminium smelter casthouse.* Mechanical and Industrial Engineering Department, University of Iceland.

Lavoie, P. (2010, May). Interview for Virtual Smelter Specifications. (T. Harton, Interviewer)

Matloff, N. (2008). *Introduction to Discrete-Event Simulation and the SimPy Languages.*

Nicholls, M. G. (1995). Aluminium Production Modelling~ A Nonlineas Bilevel Programming Approach. *Operations Research, Vol 43* , 208 218.

Nicholls, M. G. (1997). Devoloping an integrated model of an aluminium smelter incorporating sub-models with different time bases and levels of aggregation. *European Journal of Operations Research 99* , 477-490.

Piehl, T. (2000). Cell Batching for the New Zealand Aluminium Smelter. *Annual Conference of ORSNZ.*

Ryan, D. (1998). Optimised Cell Batching for New Zealand Aluminium Smelters Ltd. *Annual Conference of ORSNZ.*

Tuck, S. (1997). *Optimal Cell Batching fo New Zeland Aluminium Smelters Tiwai Point Facility.* Year 4 Project, Dept of Engineering Science, University of Auckland.

Demand Management: Effective, Evil, or Just Everyday?

Tava Olsen
Department of Information Systems and Operations Management
The University of Auckland
New Zealand
t.olsen@auckland.ac.nz

Abstract

Revenue management is ubiquitous in the airline industry, but other forms of demand management (i.e., pricing mechanisms to smooth demand), are less common (and some have a downright unfriendly reputation). This presentation examines demand management in a variety of situations, some where it has been shown to be effective and some where it is not yet in use. It profiles some of Professor Tava Olsen's own research on lead time-based pricing, as well as examining the potential of peak-hour pricing as a way to manage road congestion. The presentation will also speculate on situations where Tava believes demand management may well become everyday and explains when it can be used effectively to create win-win situations. One of the biggest hurdle faced by demand management is consumer attitudes, and the presentation will suggest ways to mitigate its negative perception. Potential avenues for academic research on demand management will also be highlighted.

Achieve Smarter Decision Outcomes with Optimisation

Chun Zi Wu
ILOG Optimization (AP)
IBM Software Group
wucz@sg.ibm.com

Abstract

Discover how optimisation can help you achieve smarter decision outcomes for your organisations complex business constraints. In addition, you will hear about the latest industry applications, case studies, updates on IBM ILOG Optimization solutions and the recent IBM Academic Initiative Program for ILOG Optimization.
.

Collateralised Debt Obligation Portfolio Optimisation

Min Gih Choi
Department of Engineering Science
University of Auckland
New Zealand
mcho099@aucklanduni.ac.nz

Abstract

Billions of dollars worth of CDO portfolios are issued every year, with individual portfolios being worth hundreds of millions of dollars. Such large investments need to be managed, and conventionally this has been done manually under the discretion of a manager proposing potential trades. Managing these portfolios requires balancing various metrics designed to measure the performance of the portfolio. A complex calculation which would provide an indication of the value of the portfolio can be carried out, known as the Par Coverage value. Under association with leading asset manager, M&G Investments, such calculations were reformulated into a mixed integer linear programme (MILP), in order to identify a more efficient portfolio, through maximising the Par Coverage value of any given CDO portfolio. The implementation of the optimisation showed promising results and was able to identify an efficient frontier of potential portfolios.

Key words: Collateralised Debt Obligations, CDO, Portfolio Optimisation, Par Coverage value, Weighted Average Spread, Weighted Average Rating Factor, Efficient Frontier.

1 Introduction

Collateralised Debt Obligations (CDOs) are a type of investment that is common in the finance industry. A CDO portfolio can be constructed through combining multiple individual *obligations* (otherwise known as assets or investments). Most CDO portfolios are actively managed to ensure the overall health of the portfolio is maintained despite fluctuations in the quality of the individual obligations. The management of a CDO is done through buying and selling obligations. In order to prevent bad choices and to keep the quality of the CDO within the conditions offered to the investors, there exist certain criteria and measures of portfolio performance (Fabozzi 2002). Common criteria and measures include the *Par Coverage Value* test, the *Weighted Average Spread* (WAS) trigger level, and the *Weighted Average Rating Factor* (WARF) trigger level.

1.1 Aim

Given a list of obligations that could be added to a portfolio, and an initial starting portfolio, the aim was to identify more efficient portfolios by buying and/or selling individual obligations. The problem was converted into a mixed integer linear problem (MILP) and solved with a MILP solver. As the manner in which a CDO portfolio is operated and managed varies depending on the type of CDOs and style of the portfolio manager, this paper only deals with a particular type of CDO under advice from a sponsoring company, M&G Investments. In modelling this CDO portfolio problem, the following four calculations are taken into account. the Par Coverage value, Weighted Average Spread (WAS), Weighted Average Rating Factor (WARF), and the principal cash sum. The Par Coverage value was selected as the first indicator to optimise over due to its complexity. The formulation of the MILP can be summarised as:

Maximise:	Par Coverage Value		
Subject to:	WAS	$\geq$	*minimum WAS level*
	WARF	$\leq$	*maximum WARF level*
	Principal Cash	$\geq$	*minimum cash*

2 Modelling a CDO

2.1 Parameters

The input would be provided in the form of a list of obligations. This list includes all obligations that are currently in the portfolio, and further appended are obligations that are available to purchase into the portfolio. Each obligation in the list will have associated with it an array of attributes identifying things ranging from the value of the obligation, to whether or not the obligation has defaulted (see Table 1). These attributes are necessary for the calculation of the values in the MILP and are inputted as the following parameters:

Parameter	Definition	Type
isD	*Defaulted Obligation:* an obligation that has failed to pay back a loan.	Binary
isLD	*Long Dated Obligation:* the maturity date of the obligation is past a predefined date.	Binary
isZC	*Zero Coupon Securities:* obligations that do not make periodic interest payments, and instead pay all interest at the end of maturity.	Binary
isDisc	*Discount Obligation:* obligation was purchased at less than a predefined threshold of the principal amount.	Binary
isN	*Normal Obligation:* when an obligation is not a Defaulted, Long Dated, or Zero Coupon obligation.	Binary
isCCC	*CCC Obligation:* has a credit rating that is considered CCC or below. They are considered highly risky investments.	Binary
inPort	Obligation was in the portfolio before optimisation.	Binary
MR	*Moody's Recovery:* the recovery rate as advised by Moody's Investor Service. The value of Long Dated obligations.	Continuous
SPR	*S&P recovery:* the recovery rate as advised by Standard & Poor's.	Continuous
AV	*Accreted Value:* the value of Zero Coupon securities.	Continuous
PP	*Purchase Price:* the price of Discount Obligations.	Continuous

MV	*Market Value:* the determined bid price.	Continuous
WV	*Worst Value:* Minimum of the *MR*, *SPE*, and *MV*. The value of Defaulted Obligations	Continuous
CCCMV	*CCC market value:* re-calculated *MV* for CCC obligations.	Continuous
RF	*Moody's Rating Factor:* the credit rating factor as determined by Moody's Investor Service.	Integer
SPE	*S&P rating Factor Equivalent:* the credit rating factor as determined by Standard & Poor's.	Integer
Spread	*Spread:* The rate the loan is paying above the inter-bank lending rate (percentage).	Continuous
Buy	*Cost of Buying* the obligation.	Continuous
Sell	*Cost of Selling* the obligation.	Continuous
B	*Block Size:* obligations will usually trade in defined blocks, consisting of a certain monetary value. Purchases must be bought in blocks (e.g. purchasing six $10,000 blocks, equals $60,000).	Continuous
FV	*Total Nominal/Principal/Face Value* of the obligation.	Continuous
M	*Maximum blocks:* limited by the Total Face Value. $M = \lfloor FV/B \rfloor$.	Integer

Table 1. Parameters per obligation for the CDO portfolio problem.

Additionally, there are additional parameters pertaining to the entire portfolio as follows:

Parameter	Definition	Type
ipc	*Initial principal cash*	Continuous
minimum WAS	*WAS minimum,* trigger level for the portfolio	Continuous
maximum WARF	*WARF maximum,* trigger level for the WARF	Continuous
minimum cash	Minimum amount of principal cash the portfolio can have.	Continuous

Table 2. Additional parameters for the CDO portfolio problem.

2.2 Processes Before Solving

Before solving for the MILP, it is necessary to manipulate the input parameters in a form that is applicable for the linear formulation. All the binary parameters; *isD*, *isLD*, *isZC*, *isN*, *isDisc*, *isCCC*, and *inPort* indicate categories that the obligations can be in, and each category has associated with it a multiplier; *WV*, *MR*, *AV*, 1, *CCCMV*, and *PP* respectively, and no multiplier for the special category of *inPort*. It is necessary for the first 4 categories to be mutually exclusive. This is done by comparing the respective multipliers of the categories the obligation is in, and finding the lowest value. The obligation will be considered being in this category only, and no longer in any of the other three. All obligations must be in exactly one of these first 4 categories.

The categories *isDisc*, and *isCCC*, also need to be mutually exclusive and similarly, the respective multipliers are compared and the lowest selected. It is possible for an obligation to be neither of these 2 categories, or one of these categories, but not both.

The special category of *inPort* indicates which obligations were originally in the portfolio, and no changes are made to this category.

Finally all obligations in the entire list of obligations need to be ordered by *CCCMV*, such that the i^{th} obligation would always have a *CCCMV* equal or smaller to the j^{th} obligation for $j = i, i+1, ..., n$.

2.3 Problem Definition

The Par Coverage value, WAS, WARF, and Principal Cash sum calculations are as follows. Certain calculations can differ depending on the type of CDO portfolio, and the style of the management. Therefore the calculations utilised were as advised from M&G Investments for a specific Par Coverage value calculation.

For calculating the Par Coverage value of a given portfolio, first the value of all obligations in each of the categories of *isD*, *isLD*, *isZC*, and *isN* are summed up multiplied with the their respective multipliers to form a value known as the Aggregate Collateral Balance (ACB). The value of all obligations that are in the category of CCC obligations are also summed together to form a value known as the Aggregate Principal Balance of CCC obligations (APBCCC). If the APBCCC is larger than 7.5% × ACB, then the amount that is in excess is known as the CCC excess size. If it is less, then the following calculations for the CCC obligations are not performed.

A special subset of CCC obligations are selected (known as the CCC excess list) such that the aggregate value of this subset is equal to, this CCC excess size. The subset must be chosen in ascending order of *CCCMV* value. It is unlikely that the chosen subset will fit exactly within the CCC excess size, and so the last obligation in the subset will specifically be covering the threshold defined by the 7.5% × ACB. For this special obligation, only the fraction of that is included in contributing to the CCC excess size is considered (see Figure 1). If the list fits exactly, then the fractional proportion is 0.

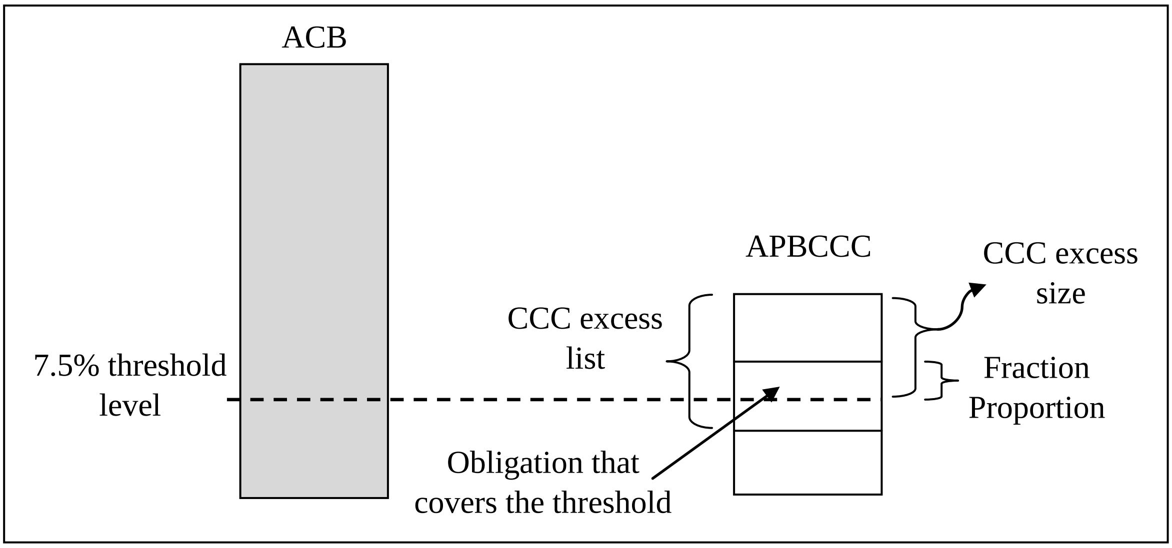


Figure 1. Visual representation of the CCC excess components.

Once this subset is obtained the value of the obligations in the list are multiplied by $1 - CCCMV$ and summed up. For the special obligation that may be covering the threshold, only the fractional proportion is multiplied and considered. The final entire summed value is known as the CCC adjustment value.

For obligations in the category of *isDisc*, the value is summed up multiplied by 1-*PP* and this sum is known as the Discount obligation adjustment.

The Par Coverage value is then the ACB subtracted by the CCC adjustment and Discount obligation adjustment values.

The Weighted Average Spread and Weighted Average Rating Factor are the average Spread, and Rating Factor respectively, weighted by the value of the obligations. Defaulted obligations are not considered when calculating the WAS and WARF.

There also exists an amount of *principal cash* associated with the portfolio for buying obligations. If obligations are sold, then the proceeds are added to the principal cash. It is necessary for there to be enough principal cash if an obligation is to be bought.

2.4 Model Formulation

Indices

i = obligation in list, 1, 2, ..., n.

j = obligation in list, i, i+1, ..., n.

Decision Variables

$$x_i = \begin{cases} 1 & \text{if obligation } i \text{ is in Portfolio} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i = 1, 2, \ldots, n$$

$$z_i = \begin{cases} 1 & \text{if obligation } i \text{ is in the CCC excess list, excluding covering obligation} \\ 0 & \text{otherwise} \end{cases}$$

for i = 1, 2, ..., n

$$w_i = \begin{cases} 1 & \text{if obligation } i \text{ is in the CCCexcess list, including covering obligation} \\ 0 & \text{otherwise} \end{cases}$$

for i = 1, 2, ..., n

$xn_i \in \{0, 1, 2, \ldots, M_i\}$, the Number of blocks of obligation i that is in the portfolio

for i = 1, 2, ..., n

$zn_i \in \{0, 1, 2, \ldots, M_i\}$, value of xn_i if obligations i is in z list, 0 otherwise

for i = 1, 2, ..., n

$wn_i \in \{0, 1, 2, \ldots, M_i\}$, value of xn_i if obligations i is in z list, 0 otherwise

for i = 1, 2, ..., n

$$Tsub_i = \begin{cases} \text{FP, if i the threshold obligatio}n \\ \quad 0 \text{ otherwise} \end{cases}$$

Fractional Proportion of Threshold Obligation Substitute

for i = 1, 2, ..., n

$extra$ = a slack variable preventing negative CCC excess list sizes.

Model

Maximise:

Par Coverage value:

$$
\begin{aligned}
&= \sum_{i=1}^{n} isN_i\, B_i\, xn_i \\
&+ \sum_{i=1}^{n} isD_i\, WV_i\, B_i xn_i \\
&+ \left(ipc \; - \sum_{i=1}^{n} (1 - InPort) Buy_i\, B_i xn_i \right. \\
&+ \left. \sum_{i=1}^{n} (InPort) Sell_i\, B_i (M_i - \boldsymbol{xn}_i) \right) + \sum_{i=1}^{n} isLD_i\, MR_i\, B_i xn_i \\
&+ \sum_{i=1}^{n} isZC_i\, AV_i\, B_i xn_i - \sum_{i=1}^{n} (1 - CCCMV_i)\, B_i zn_i \\
&- \sum_{i=1}^{n} (1 - CCCMV_i) Tsub_i - \sum_{i=1}^{n} isDisc_i\, (1 - PP_i)\, B_i xn_i
\end{aligned}
$$

Subject to:

(1)

$$
\begin{aligned}
&\left(\sum_{i=1}^{n} isCCC_i\, B_i\, xn_i \right) \\
&-0.075 \\
&\times \left(\sum_{i=1}^{n} isN_i\, B_i\, xn_i + \sum_{i=1}^{n} isD_i\, WV_i\, B_i xn_i \right. \\
&+ \left(ipc \; - \sum_{i=1}^{n} (1 - InPort) Buy_i\, B_i xn_i \; + \sum_{i=1}^{n} (InPort) Sell_i\, B_i (M_i - xn_i) \right) \\
&+ \left. \sum_{i=1}^{n} isLD_i\, MR_i\, B_i xn_i + \sum_{i=1}^{n} isZC_i\, AV_i\, B_i xn_i \right) \\
&+ extra \geq \; 0
\end{aligned}
$$

(2) (i)	$xn_i \;\geq\; x_i$	for i = 1, 2, ... n
(2) (ii)	$xn_i \;\leq\; M_i\, x_i$	for i = 1, 2, ... n
(3) (i)	$z_i \;\leq\; isCCC_i\, x_i$	for i = 1, 2, ... n
(3) (ii)	$w_i \;\leq\; isCCC_i\, x_i$	for i = 1, 2, ... n
(4) (i)	$zn_i \;\geq\; z_i$	for i = 1, 2, ..., n
(4) (ii)	$zn_i \;\leq\; M_i z_i$	for i = 1, 2, ..., n

(5) (i) $zn_i \le xn_i$ for i = 1, 2, ..., n

(5) (ii) $zn_i \ge xn_i - M_i(1 - z_i)$ for i = 1, 2, ..., n

(6) (i) $wn_i \ge w_i$ for i = 1, 2, ..., n

(6) (ii) $wn_i \le M_i w_i$ for i = 1, 2, ..., n

(7) (i) $wn_i \le xn_i$ for i = 1, 2, ... n

(7) (ii) $wn_i \ge xn_i - M(1 - w_i)$ for i = 1, 2, ..., n

(8) (i)

$$\begin{aligned}\sum_{i=1}^{n} B_i\ zn_i \le & \left(\sum_{i=1}^{n} isCCC_i\ B_i\ xn_i\right) - 0.075 \\ & \times \left(\sum_{i=1}^{n} isN_i\ B_i\ xn_i + \sum_{i=1}^{n} isD_i\ WV_i\ B_i xn_i\right. \\ & + \left(ipc - \sum_{i=1}^{n}(1 - InPort)Buy_i\ B_i xn_i\right. \\ & \left. + \sum_{i=1}^{n}(InPort)Sell_i\ B_i(M_i - xn_i)\right) + \sum_{i=1}^{n} isLD_i\ MR_i\ B_i xn_i \\ & \left. + \sum_{i=1}^{n} isZC_i\ AV_i\ B_i xn_i\right) + extra\end{aligned}$$

(8) (ii)

$$\begin{aligned}\sum_{i=1}^{n} B_i\ wn_i \ge & \left(\sum_{i=1}^{n} isCCC_i\ B_i\ xn_i\right) - 0.075 \\ & \times \left(\sum_{i=1}^{n} isN_i\ B_i\ xn_i + \sum_{i=1}^{n} isD_i\ WV_i\ B_i xn_i\right. \\ & + \left(ipc - \sum_{i=1}^{n}(1 - InPort)Buy_i\ B_i xn_i\right. \\ & \left. + \sum_{i=1}^{n}(InPort)Sell_i\ B_i(M_i - xn_i)\right) + \sum_{i=1}^{n} isLD_i\ MR_i\ B_i xn_i \\ & \left. + \sum_{i=1}^{n} isZC_i\ AV_i\ B_i xn_i\right) + extra\end{aligned}$$

(9) (i)

$$\sum_{i=1}^{n} z_i \le \sum_{i=1}^{n} w_i$$

(9) (ii)

$$\left(\sum_{i=1}^{n} z_i\right) + 1 \ge \sum_{i=1}^{n} w_i$$

(10) (i) $$z_j \le 1 - (x_i - z_i)$$
$i = 1, 2, ..., n, \;\; j = i, i+1, ..., n$
for i and j that are *isCCC*

(10) (ii) $$w_j \le 1 - (x_i - w_i)$$
$i = 1, 2, ..., n, \;\; j = i, i+1, ..., n$
for i and j that are *isCCC*

(11)
$$\begin{aligned}Tsub_i \ge {} & \left(\sum_{i=1}^{n} isCCC_i\, B_i\, xn_i\right) - 0.075 \\ & \times \left(\sum_{i=1}^{n} isN_i\, B_i\, xn_i + \sum_{i=1}^{n} isD_i\, WV_i\, B_i xn_i \right. \\ & + \left(ipc - \sum_{i=1}^{n} (1 - InPort) Buy_i\, B_i xn_i \right. \\ & \left. + \sum_{i=1}^{n} (InPort) Sell_i\, B_i (M_i - xn_i) \right) + \sum_{i=1}^{n} isLD_i\, MR_i\, B_i xn_i \\ & \left. + \sum_{i=1}^{n} isZC_i\, AV_i\, B_i xn_i \right) + extra - \left(\sum_{i=1}^{n} B_i\, zn_i\right) - B_i M_i \times (1 \\ & - (w_i - z_i))\end{aligned}$$
for $i = 1, 2, ..., n$

(12)
$$\sum_{i=1}^{n} (1 - isD_i)\; spread_i B_i xn_i \ge minimum\, WAS \times \sum_{i=1}^{n} (1 - isD_i)\; B_i xn_i$$

(13)
$$\sum_{i=1}^{n} (1 - isD_i)\; RF_i B_i xn_i \le maximum\, WARF \times \sum_{i=1}^{n} (1 - isD_i)\; B_i xn_i$$

(14)
$$\begin{aligned}ipc - \sum_{i=1}^{n} (1 - inPort_i)\, Buy_i\, B_i xn_i + \sum_{i=1}^{n} inPort_i Sell_i\, B_i (M_i - xn_i) \\ \ge minimum\; cash\end{aligned}$$

3 Results

The formulation was implemented in the Python programming language, through the PuLP optimisation interface. Various solvers were tested, however the following results were obtained from the CBC solver.

Considering the inverse correlation between the Par Coverage and the WAS value, and the fact that it would also be beneficial to consider portfolios with high WAS values, an efficient frontier could be created by maximising the Par Coverage value multiple times, incrementing the *minimum WAS* parameter each time (see Figure 2).

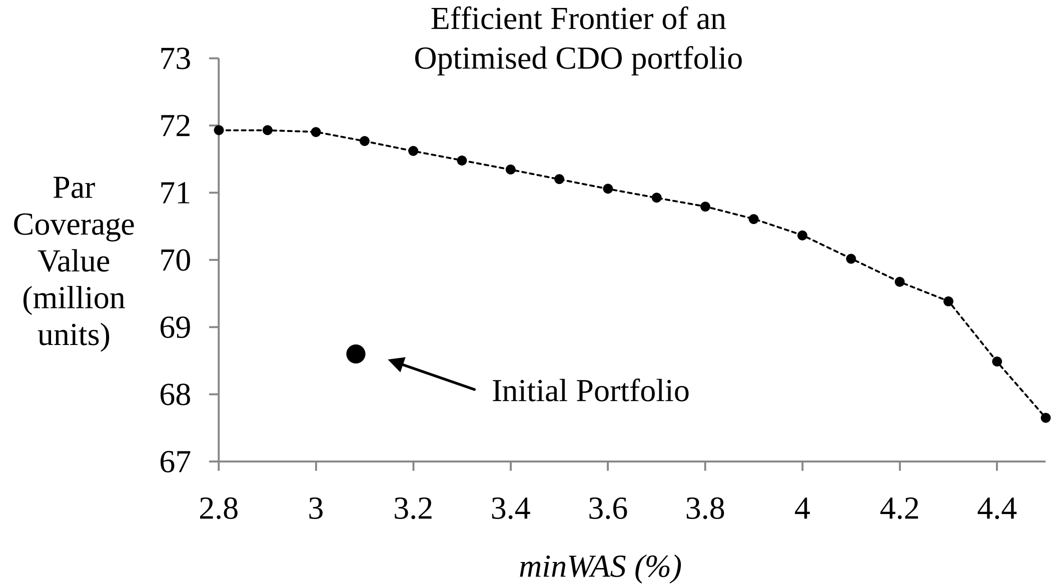


Figure 2. Results from a small test CDO portfolio constructed out of real data provided by M&G Investments. Values for real investment CDO portfolios were omitted as the data is confidential, however were tested and show similar results.

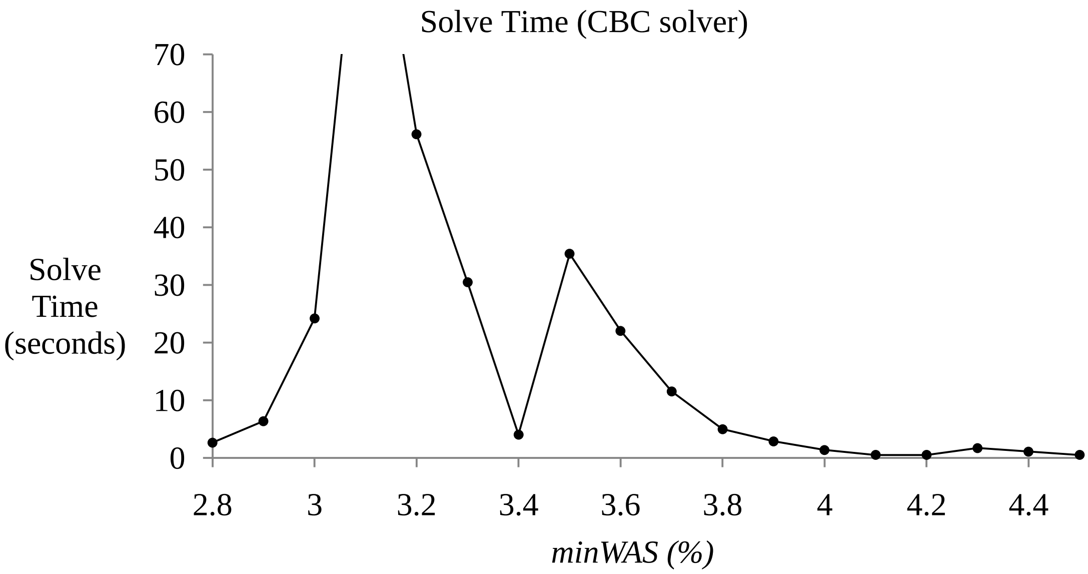


Figure 3. Solve Time resulted from obtaining the results in Figure 2.

The Solve speed was found to be highly variable and was a major issue with certain portfolios taking a considerable amount of time to solve, or unsolvable due to exceeding computer memory. This can be seen in for the small test portfolio with a value of around 69 million (currency) units solved in Figure 3, near a *minimum WAS* of 3.1. A typical CDO portfolios are much larger with values ranging in the hundreds of millions of units

The variability in solve time was found to be due to the consideration of CCC obligations. For low *minimum WAS* (i.e., in Figure 3, near 3% and below), the CCC obligations are removed in order to minimise the CCC excess list to 0. For high *minimum WAS* (i.e., near 3.6% and higher), CCC obligations were mostly included in the portfolio, also resulting in fast solve speeds. In between however, the solver must make a lot of considerations between all the CCC obligations against the rest of the portfolio, and this resulted in a very difficult problem to solve.

4 Conclusion

The combined calculations of the Par Coverage value, WAS value, and WARF value were successfully converted into a mixed integer linear programme(MILP) that could be solved in a MILP solver. From the results of the numerous test inputs, general observations could be concluded; with the Par Coverage value as the objective function, there was a tendency for the MILP to attempt to minimise excess CCC category obligations.

Due to the inverse relationship of the value of obligations and the spread of obligations, an efficient frontier could be obtained in the results. For all test problems that were solved, the MILP was able to identify an efficient frontier of portfolios with higher Par Coverage and WAS up to this frontier.

Acknowledgments

Acknowledgments go to Dr. Michael O' Sullivan and Dr. Cameron Walker who oversaw and supervised this paper. Furthermore, M&G Investments provided all the commercial data and documents that were essential for implementing the model. Special thanks to Matthew O'Sullivan from M&G Investments for all the help and advice on the necessary calculations and data.

5 References

Adelson, M. 2004, *CDOs in Plain English*, Nomura Fixed Income research, Tokyo

COIN-OR Branch and Cut (CBC) Solver [https://projects.coin-or.org/Cbc]

Dippy 0.8.7 MILP Solver [https://projects.coin-or.org/Dip]

Fabozzi, F, Goodman L. 2002, *Collateralized Debt Obligations: Structures and Analysis*, Chichester, UK.

Gurobi Optimizer 3.0 [http://www.gurobi.com/]

PuLP Python LP interface [http://www.coin-or.org/PuLP/]

Python Programming Language 2.6 [http://www.python.org/]

Note: Internal documents provided by M&G Investments were consulted and are confidential in nature and thus are not referenced.

Optimisation of Demand-Side Bidding

Jason Alfons Undan
Department of Engineering Science
University of Auckland
New Zealand
jund013@aucklanduni.ac.nz

Abstract

We consider the problem of formulating an optimal bid curve (demand function) for a large consumer in an electricity market aiming to maximise its profits. A dynamic programming approach was suggested in this paper to construct the optimal bid strategy.

The aim of this approach is to use the consumer's profit function to approximate the demand needed in order to maximise its returns. This method could potentially try to reduce consumption of electricity and increase the efficiency of the market by having more active participations from the demand-side. We present numerical results using artificial offer-curves and real-offer curves (obtained from New Zealand Electricity Market for a large consumer).

Key words: Dynamic programming, bidding strategy, revenue function, electricity market.

1 Introduction

Electricity is an important element for large consumers because it is an input for many production and manufacturing processes. As significant amount of electricity are required by these users, it normally becomes the major cost after raw materials. Let us consider an aluminium smelter example where q units of electricity are needed to produce 1 tonne of aluminium. The 1 tonne of aluminium is then sold at market price and the company collects \$ t per tonne of aluminium. Total gross profit gained by the company in producing and selling 1 tonne of aluminium is the revenue of selling 1 tonne of aluminium less the cost of purchasing q units of electricity to produce the 1 tonne of aluminium. This suggests that the revenue generated by the aluminium smelter depends on the amount of electricity units, q purchased to produce a certain amount of aluminium.

Thus, the cost of electricity has direct impact on the revenue of these electricity users. If the price of electricity at a particular trading period is too high that it becomes not profitable to operate, the large consumer has the option to shut down or continue with their production. In our problem, we consider large consumers to have the flexibility to reduce their consumption depending on the spot prices. These consumers can remain shut down, with a cost associated to that decision, if the price of electricity is high that to operate is not a feasible option.

The aim of this project is to develop a methodology for a purchaser to bid for electricity optimally (i.e. submit an optimal demand stack or bid curve to the centralised system) as to maximize the expected profit from purchasing and using certain amount of electricity units, q. We present a dynamic programming approach to solve the optimal bidding problem for the purchaser based on the revenue generated from the purchase of electricity units. In other words, the revenue of the purchaser applied into our method is a function of electricity units purchased. A previous work had been done by Pritchard (2007) to formulate an optimal offer curve for an electric power generator with market power. What this project is trying to do is to develop the opposite formulation for a large consumer to formulate their optimal bid curve.

We developed the method using simple, one-offer curve and three-offer curves examples which can easily be solved analytically. We then used the methodology to solve problems using realistic offer curves obtained from the New Zealand Electricity Market (NZEM) to evaluate its potential of solving real-world problems.

The NZEM has been chosen to explore our methodology in this project. However, other electricity markets may as well be an appropriate market for our methodology. In New Zealand, the term "electricity market" typically refers to the wholesale market where demand every half hour at every node in the network is satisfied at the lowest possible price based on the offers of the generators. In the wholesale market, electricity is purchased by retailers and large consumers.

We describe the dynamic programming formulation and algorithm in Chapter 2 and present the one-offer curve and three-offer curves examples and real-offer curves from NZEM for consideration. For the real-offer curves problem, we assume to be one of the large consumers in New Zealand, i.e. The New Zealand Aluminium Smelter. We analyse and discuss the results in Chapter 3 and finally in Chapter 4 we draw conclusions from this project.

To illustrate an option that a large consumer can consider rather than shutting down completely, supposed in a single node market, where all other demands have been satisfied. There is a residual generation curve and as a large consumer who has a demand that has not been dispatched, the question faced is how to bid optimally in order to maximise the purchaser's return.

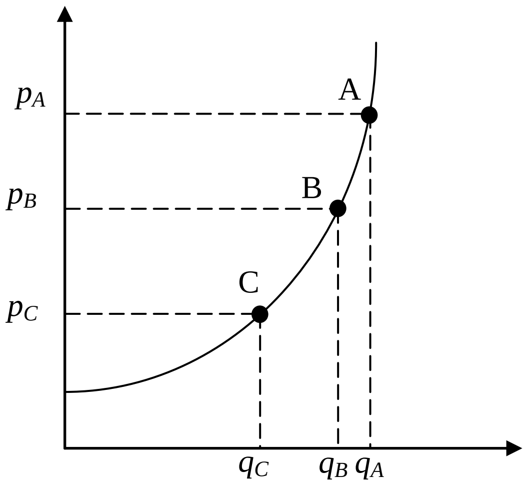


Figure 1.1: An offer stack that has three possible points of dispatch that can be considered by a purchaser

In the diagram above, there are three possible points of dispatch that the large consumer can consider. The initial demand is point A. A natural way of bidding is to bid at A as to get the desired demand of electricity. However, if the consumer can be flexible enough to reduce consumption, bidding at pB is better considering that the price at B is less than A for a relatively small reduction in consumption. Consequently, the cost of electricity can be reduced quite significantly with the decision to be dispatched at reduced consumption.

An optimal point of dispatch cannot be guaranteed unless the actual revenue function of the company is known. Perhaps being dispatched at C may be a good decision too, however it requires greater amounts of consumption to be reduced. The large consumer needs to consider the economic benefits of being dispatched at a reduced consumption, or not being dispatched at all.

Demand-side management proves to provide benefits in electricity markets especially in making the whole system more efficient and effective (Electricity Commission, 2009). However, different market structures and norms pose themselves as challenges for the demand-side management potentials. It will require many consultations with stakeholders, investment capitals and time before a decent demand-side policy will take off effectively in an electricity market. Nevertheless, it is hoped that our method will be able to contribute to more active demand-side participation into the electricity market taking into consideration its unique structures and constraints.

2 Methodology

2.1 Dynamic Programming

Dynamic Programming is an optimisation methodology developed in the 1940s by Richard Bellman to describe the process of solving problems where one needs to find the best decision one after another. The main concept of this technique lies in the principle of optimality which can be stated as follows:

An optimal policy has the property that whatever the initial state and the initial decision are; the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision (Luus, 2000).

Generally, today, it is known as an approach to solving complex problems by breaking them into simpler steps or structures. To put the method into context, we have a complex electricity bidding problem faced by a consumer. Given an offer curve consisting of few tranches, we are interested in knowing the best way to bid for electricity, possibly submitting a bid curve that will maximise our expected profits as consumers. This approach can be done by first subdividing the quantity-price (q, p) plane to form smaller structures or sub problems of the main problem.

2.1.1 Discretisation of (q, p) plane

We begin by subdividing the (q, p) plane with a finite grid of M by N rectangular cells. Any class of admissible bid curves are restricted to follow only the edges of this grid (i.e. a bid curve must intersect each cell of the grid only in its boundary), and are monotone decreasing. Consequently, there are only a finite amount of admissible bid curves, each consisting of a finite sequence of exactly M + N horizontal and vertical line segments (grid edges).

A key observation is that for a fixed grid edge e, the occurrence of a point of dispatch on e is independent of what other grid edges are included in the bid curve. Therefore, for each grid edge that intersects with the offer curves (point of dispatch occurring at e), there is an expected profit V (e) associated, which will be realised if and only if e is part of the bid curve. The expected profit of an admissible bid curve r is then

$$f(r) = \sum_{e \in r} V(e)$$

2.1.2 Edge Value and Revenue Function, $\boldsymbol{\rho}$

Each grid edge, e in the discretised (q, p) plane has an edge value. It is the expected profit that a consumer will receive by purchasing q units of electricity at the price of p, if and only if the grid edge, e is included in the bid curve and point of dispatch occurs at e. To calculate the expected profit for each grid edge in the plane, we need to have a revenue function, ρ for the problem. This can be done via an equation such as $f(e) = \rho(q) - C(q)$; expected profit at any given grid edge is the revenue from using q units of electricity less the payoff (cost of electricity) to the generators when purchasing q units of electricity. Our problem is to find the maximum of f over its (finite) domain.

2.1.3 Problem Formulation

Suppose our grid covers the region $q_{min} \leq q \leq q_{max}$, $p_{min} \leq p \leq p_{max}$, and that all possible points of dispatch lie within this region. For each vertex x of the grid, let W(x) denote the maximal expected profit, due to points of dispatch above x, of any bid curve which passes through x.

Then

$$W(x) = \max\left(W_l(x), W_u(x)\right),$$

where

$$W_l(x) = \begin{cases} -\infty, & \text{if } x \text{ is on } q = q_{min} \\ V\left(e_l(x)\right) + W\left(v_l(x)\right), & \text{otherwise,} \end{cases}$$

and $e_l(x)$, $v_l(x)$ are respectively, the vertex adjacent to x to the left, and the edge linking x to that neighbour.

Similarly,

$$W_u(x) = \begin{cases} -\infty, & \text{if } x \text{ is on } p = p_{max} \\ V\left(e_u(x)\right) + W\left(v_u(x)\right), & \text{otherwise,} \end{cases}$$

where $e_u(x)$, $v_u(x)$ are respectively, the vertex adjacent to x above, and the edge linking x to that neighbour.

We have $W(x) = \left((q_{min}, p_{max})\right) = 0$. It is thus straightforward to successfully evaluate W(x) for each vertex x of the grid, and hence determine the optimal admissible bid curve. This procedure clearly yields a global optimum for the discretised problem which can also be taken as an approximation of the optimal bid curve for the original problem.

2.1.4 Estimation of Optimal Expected Profit

There are two main factors that affect the optimal expected profit of an admissible bid curve for the discretised problem, i.e. size of the grid and the revenue function that is put into the problem.

The size of the grid refers to the vertical height and horizontal width of a particular rectangular cell in the (q, p) plane after being subdivided into M by N cells. In theory, we will expect finer grids to give better optimal expected profits. However as the plane can be divided into any arbitrary values, it can be expected that the optimal expected profit will have an upper bound that defines the maximal limit the profit can be, provided the revenue function is kept constant.

Revenue function is important in computing the expected profit in the discretised problem as the calculation is done at each intersection of the offer curves with the grid to compute the edge values. An accurate revenue function will likely to give an acceptable optimal bid curve for the consumer because our problem is to maximise the sum of edge values in the grid so as to form a monotone decreasing bid curve. A poor revenue function will most probably give an ineffective bid curve and will not likely make any meaningful decision for the consumer.

2.2 One-Offer and Three-Offer Curves Examples

To verify the effectiveness of the dynamic programming algorithm, we use it to solve simple one-offer and three-offer curves examples. Given a (q, p) plane with quantity and prices ranging from $0 \leq q \leq 40$ and $0.0 \leq p \leq 4.0$ respectively. We let the revenue function at any point on the grids be $\rho(q) = 5q$. We subdivide the plane into four by four rectangular cells. Table 3.1 and table 3.2 shows the offer stacks and aggregated offer stack submitted for the one-offer curve and three-offer curves examples respectively.

Offer Curve A		
Price ($)	MWh	Cumulative MWh
0.00	5	5
0.04	12	17
1.50	8	25
2.80	3	28

Table 2.1: One-offer curve example

Offer Curve A			Offer Curve B			Offer Curve C		
Price ($)	MWh	Cumulative MWh	Price ($)	MWh	Cumulative MWh	Price ($)	MWh	Cumulative MWh
0.00	5	5	0.00	5	5	0.00	2	2
0.04	12	17	0.05	18	23	0.06	11	13
1.50	8	25	0.07	3	26	1.70	11	24
2.80	3	28	3.20	3	29	2.40	5	29

Table 2.2: Three-offer curves example

We repeat the example using a smaller grid size plane of fifteen by fifteen rectangular grid cells and a different revenue function, $\rho(q) = 6\sqrt{9q}$ is also considered using both grid sizes to provide comparisons between different scenarios.

2.4 Real-Offer Curves from NZEM

After looking into the two examples above, we proceed to use our method to solve a realistic problem using real historic offer curves from the NZEM. As an example to illustrate the potential application of our method specifically for large consumers, we considered ourselves as being the New Zealand Aluminium Smelter at Tiwai Point with our grid exit point (GXP) as TWI2201. The data are obtained from M-Co Ltd website.

The period that was taken as samples for our method is a 5-day week both in summer and winter of 2008, i.e. 11-15 February 2008 and 7-11 July 2008 respectively. However, not all of the 48 periods in a day were considered. We focused on a part of the day (weekday daytime) from 6.30 a.m. to 5.00 p.m. (i.e. period 13 to 34) which are treated as “equivalent periods” which was similarly done by Pritchard and Zakeri (2003). The multiple offer stacks obtained from these periods provide us with a distribution of offer curves that we believe to be similar to each other. These offer stacks present a general pattern that suggests a distribution of an offer curve. For our project, the estimation of the offer stacks distribution is not explored. However with our data, we can strongly believe, it represents an accurate variation of the offer stacks investigated.

2.4.1 Aggregated Residual Generation Curves

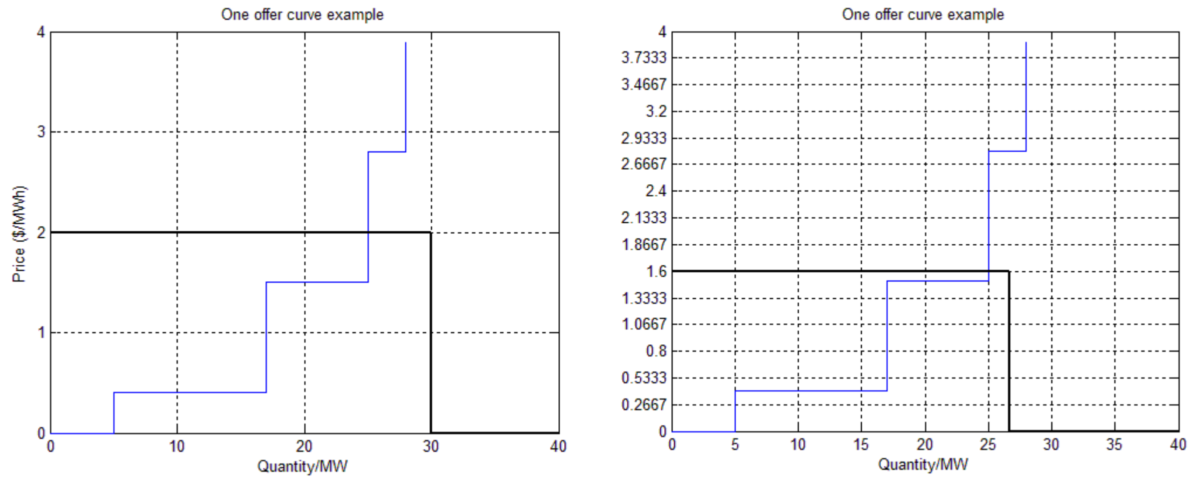
We have taken all the offer curves submitted by all generators in the NZEM and combined them to form aggregated generation offer. We also obtained all the demands except the demand at TWI2201 for the periods and aggregated them to form aggregate demand curves. In theory, the NZEM works in such a way that the intersection between the aggregate generation offer and aggregate demand curve is the point of dispatch at a specific period of the day. Having satisfied all the demands except the demand at TWI2201, we have a residual generation curve which is the quantity that the market supplies that is not consumed by other demanders at any given price.

With the residual generation curve, as a large consumer, our method aims to find the best way to bid in for electricity in a way to maximise our expected profit. We have used a linear revenue function to evaluate our optimal expected profit which is $\rho(q) = 250q$. Variations on the grid sizes were also done to analyse the effectiveness of the methodology on a larger scale.

3 Results and Discussion

3.1 One-offer and Three-offer Curves Examples

For the one curve example, we created a four by four rectangular grid on the (q, p) plane. The solution is shown in Figure 3.1. The optimal expected profit from submitting the bid curve is \$75.00 (based on a revenue function, $\rho(q) = 5q$). The same one curve example is solved using a finer set of grids of fifteen by fifteen rectangular cells. The optimal expected profit increases to \$85.00. In both solutions, the offer stack intersects with the bid stack at the same quantity but different prices. As the bid curves are only allowed to follow the grid edges, a finer grid size will result in more meaningful and useful insights.

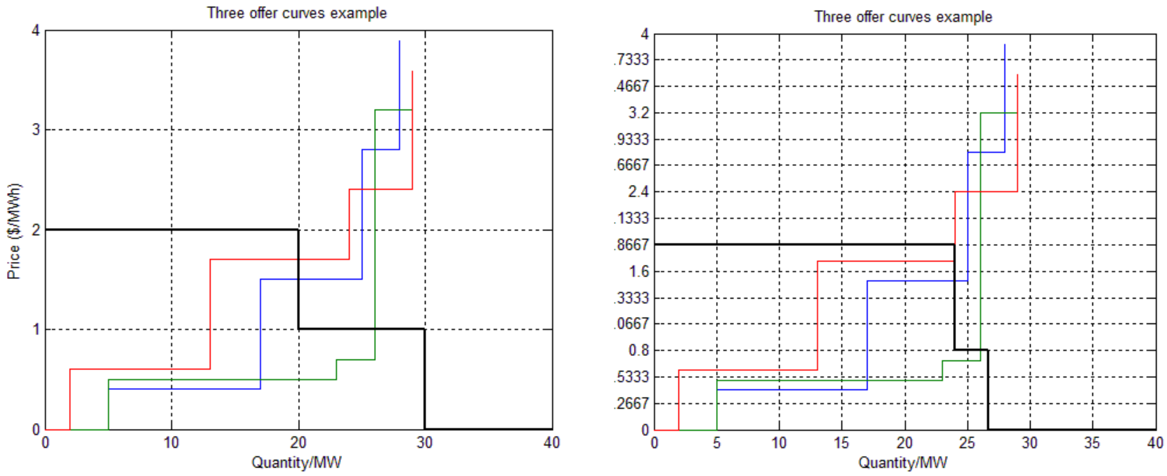


(a) (b)

Figure 3.1: Result for one offer stack example (a) 4 by 4 grid (b) 15 by 15 grid using a linear revenue function, $\rho(q) = 5q$

We also try solving the one curve example using a different revenue function, $\rho(q) = 6\sqrt{9q}$ on a four by four rectangular grid cells. The optimal expected profit from bidding the demand curve is $57.22. We repeat the example with a fifteen by fifteen rectangular grid and the optimal expected profit increases to $65.00.

For the three curves example, using a four by four rectangular grid yields an optimal expected profit of $184.00, while a fifteen by fifteen rectangular grid on the same example gives an optimal expected profit of $ 272.40. Notice the differences between both solutions in Figure 3.2 are in the setting of the bid price and quantity of demand.



(a) (b)

Figure 3.2: Result for three offer stacks example (a) 4 by 4 grid (b) 15 by 15 grid using a linear revenue function, $\rho(q) = 5q$

The three-offer curves example is solved using a different revenue function, $\rho(q) = 6\sqrt{9q}$ to yield optimal expected profits of $97.78 and $193.71 on a four by four and a fifteen by fifteen rectangular grid respectively.

It can be observed that every time we decrease the size of the grid cells, the optimal expected profit tends to increase and the bid curve shows a slight variation. It is either the consumption that is bid becoming less, or the price that we bid electricity is lesser than the price at using the initial four by four rectangular grid.

The maximum optimal expected profit gained by bidding a particular bid curve is the accumulated profit instead of actual profit. This is because in the methodology, every intersection between the offer curves and grids are considered, but in reality only one point of intersection will be realized. Nevertheless, the curve constructed using backward recursion based on maximum optimal expected profit gives a clear indication that any point of dispatch that occurs on the curve will give the best expected profit compared to others.

3.2 Real-Offer Curves from NZEM

The following table show the result of the methodology applied using real offer curves obtained from NZEM. For each season (summer and winter), two refinements of the grid sizes was done in order to evaluate the effects of grid sizes on the maximal optimal expected profits a large consumer can get from a particular bid strategy. The revenue function that is used is $\rho(q) = 250q$.

Season	Initial Result	1st Refinement	2nd Refinement
Summer	$ 2,469,100	$ 3,298,800	$ 3,617,300
Winter	$ 1,522,000	$ 1,537,800	$ 1,543,600

Table 3.1: Summary of result for real-offer curves from NZEM

The initial bid curve gives a maximum expected profit of \$ 2,469,100. As it can be observed the curve starts off at a high price of \$180/MWh taking into account any generation below that price. However, the price per unit of electricity that the consumer should pay starts to decrease if 200MW or more is required. A further refinement of the grid size is done and this refinement yielded a maximum expected profit of \$ 3,298,800; a 33.6% increase in expected profit.

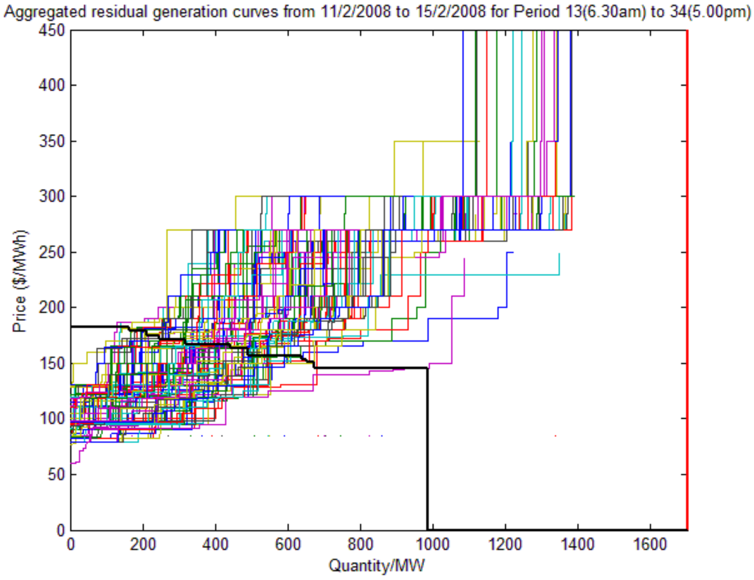


Figure 3.3: Result for real offer curves from NZEM using a linear revenue function

From the result, the difference that can be observed between the refinements is the starting price for the consumer in its bid curve is lower compared to the previous result. Instead of demanding up to approximately 1000MW, a considerable reduction in quantity of electricity demanded is noted (approximately 700MW). The price starts to decrease in a greater amount well before reaching 200MW. A further refinement of the size grids is done and the maximum expected profit found to be \$ 3,617,300, which is a 9.6% increase from the previous result. There are not many differences between these two results other than the curve following different edge lines between 150MW and 600MW.

For the periods in winter of 2008, a similar approach is done to the data. The following Figure 3.5 shows the bid curve that a large consumer can bid in to maximise its return. The bidding strategy yields a maximal expected profit of \$ 1,522,000. A refinement of the grid size gives a slight increase of 1% in the expected profit, i.e. \$ 1,537,800.

A further refinement improves the expected profit to \$ 1,543,600. It can be noted that very small improvement can be done onto the result due to the less variability in the winter curves compared to the summer curves; following particular edge lines are necessary in order to ensure maximum optimal expected profit is achieved.

The expected profits during the winter of 2008 are much less compared to the summer of the same year because the electricity generations become more expensive in winter with increasing demands, especially for heating purposes. Note that another reason for the increasing values of optimal expected profits is due to the dynamic programming methodology to bid much less consumption rather than expected actual demand. In particular, less demand means more profit can be made. Due to the finer grids in which the bid curves are confined to follow, more offer stacks are likely to intersect with these grids which lead to calculation of expected profit and accumulated values at each vertex. The dynamic programming method do not take demand into account but rather finds the best way (most optimal and profitable) way in which a purchase, in our case, the large consumer to bid in.

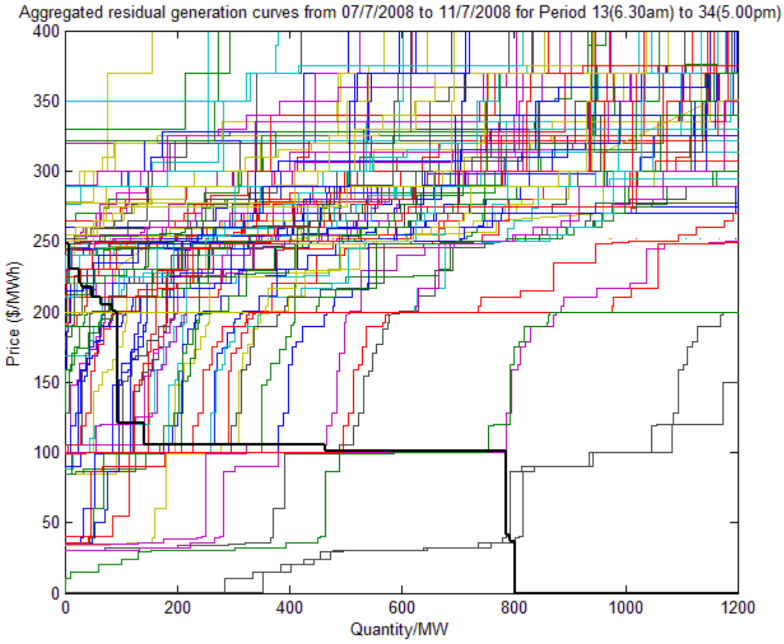


Figure 3.4: Result for real offer curves from NZEM using a linear revenue function with second improvement on grid sizes

This can become quite a significant limitation to the method as certain consumer, although we assume to be flexible, still requires some lower bound of electricity based on its demand to operate efficiently to generate profits. To operate below the identified lower bound, it may be comparable with shutting down for a particular trading period.

In reality, a large consumer may likely have a step-wise revenue function or even demand-related functions. However, as revenue functions differ significantly going from one consumer to another, the dynamic programming method is flexible in taking various classes of revenue functions to evaluate the expected profit at any point in the grid of the (q, p) plane. The consideration in submitting a revenue function into our method is that it should be scaled and forecasted accurately in order to obtain a meaningful and useful bidding strategy for the user.

4 Conclusions and Future Work

In this project, we have managed to develop a dynamic programming method for a large consumer to bid optimally into a given offer stack. This was done by discretising the (q, p) plane into M by N rectangular cells and only allow the bid curve to follow by the grid edges as to form a monotone decreasing bid stack. Each vertex in the grid is the maximal expected profit that can be obtained by a consumer if it was to bid a horizontal edge to the right or vertically downwards. The dynamic programming method then finds the best way of bidding into the offering problem by tracking back the maximal values of each vertex in the grid forming an optimal bid curve.

This method especially allows large consumers to access the potential or possibility to reduce consumption when the price of electricity is very high. A slight reduction in consumption can lead to significant cost savings for the consumer, thus impacting positively onto the grid system as a whole. It will also encourage an active participation on the demand-side of the market as consumers can manipulate the prices and bid quantity depending on the conditions of the market.

One limitation of the method is that it does not take the demand of the consumer into account. We assume consumers have the flexibility to reduce consumption, however in reality, the ability to do so is not easily achieved due to the regulations of the system and types of business a consumer is operating.

A business that has non-disruptable processes does not have as high a flexibility to shut down but it may consider reducing consumptions to avoid high spot prices. Our method gives the alternative to shutting down by allowing consumers to analyse what the total economic benefits are by reducing consumption by how much it can do so legitimately.

Another limitation of the method is that the grid is constructed over a huge (q, p) plane (i.e. maximum range of between 3000 - 6000MW) where in practice, the possibility of being dispatched for a large consumer lie at the maximum range between 450 – 700 MW. To construct a large grid where most of the spaces are not used presents itself as inefficiency in the computational technique. Multiple grid sizes in the (q, p) plane where focus is given in more likely dispatched zones is thus a possible area of future development.

More empirical studies can improve the implementation of the methodology. The data that was used to represent the offer curves distribution was quite a small sample. Perhaps using more data and applying the method using data from other electricity markets to see the variation in offer patterns is also a possible extension.

As presented in this project, the methodology was developed using single-node market examples. Another future work that can be considered is to extend the problem into a multiple-node electricity market setting with nodal pricing scheme implemented. By doing so, the effectiveness of the methodology to find an optimal bid curve, taking consideration of congestion constraints within a system network can be evaluated.

Acknowledgements

I would like to thank my project supervisor, Dr Golbon Zakeri from the Department of Engineering Science, University of Auckland for her guidance, patience and support in completing this project. Thanks to Tony Downward for his opinions and ideas that he had willingly contributed.

5 References

Electricity Commission. 2009. Dispatchable demand: options – Consultation Paper. September 2009.

Luus, Rein.2000. *Iterative Dynamic Programming*. Chapman and Hall/CRC.

M-Co Ltd. *Historic Orders*. [Online] Available: http://www.electricityinfo.co.nz/comitFta/Ongoing_bidoffer.ongoing [18 August 2010]

M-Co Ltd. *Demand*. [Online] Available: http://www.electricityinfo.co.nz/comitFta/ftaPage.demand [23 August 2010]

Pritchard G. 2007. Optimal offering in electric power networks. *Pacific Journal of Optimisation*, 3(3):425-438.

Pritchard G. and Zakeri G. 2003. Market offering strategies for hydroelectric generators. *Operations Research*, 51(4):601-612.

Aircraft Route Guidance through Convective Weather

Lawrence Li
Department of Engineering Science
The University of Auckland
New Zealand
lli143@aucklanduni.ac.nz

Abstract

When planning and choosing a flight route to take for aircrafts, an important problem is to determine whether or not a pilot should fly through or to fly around convective weather, so that the air travel delays and costs are minimised. One method of determining if a pilot should fly through or not is to find the shortest path which minimises the total distance and risk a pilot would take for a route.

The aim of the research was to determine a set of route choices for guiding pilots through convective weather based on available weather data. The problem is modelled as a bi-objective shortest path problem, which is a shortest path problem with two objectives. A flight network of the airspace is generated firstly as a 2D network, and then a 3D network.

Key words: aircraft, bi-objective shortest path, convective weather, weather avoidance

1 Background

1.1 Introduction

Convective weather is the process of rising air, which is generally warm moist air above cold air, to form clouds. This process is normally associated with heavy rainfall and hail and is known to produce strong, turbulent winds and thunderstorms. This severity is of concern, as rain and clouds can reduce visibility greatly for aircraft pilots while in flight, and the strong winds can reduce or even overwhelm the pilot's ability to fly safely. In addition, hail, icing, and thunderstorms can damage aircraft sensors and mechanical parts, which can endanger the safety of the passengers onboard the aircraft.

There are several methods that are used to find the best route to take based on the available weather data. One method is to minimise the distance an aircraft will travel and to minimise the risk the path imposes to the aircraft, for which a number of optimisation methods can be used. These methods are able to determine the best set of routes to take in terms of distance and safety, allowing the pilots to choose a route that matches their situation and needs. This research aimed to find a set of route choices for guiding pilots through convective weather by modelling the problem as a shortest path with two objectives.

1.2 Weather Risk

The precipitation intensity is one measurement used by flight dispatchers and meteorologists to help determine the severity of weather. In general, it is the amount of liquid within the atmosphere, which includes rain and hail. One method of measuring this is with vertically integrated liquid (VIL), which shows the amount of water contained in a vertical column. This is obtained from measurements of the amount of reflection from air via weather radars. VIL is a measured on a scale from the one to six, with one being the lowest level of intensity, and six representing the highest level of intensity.

Based on observations of true pilot behaviour over a period of 40 days, it was consistently found that aircrafts avoided weather that was at VIL levels from three and above (Kuhn, 2008). At each level below this however, more aircrafts were found to fly through the weather. In some cases, pilots would deviate from their current route even though it was passable, and in other cases, pilots would penetrate through the weather when it was deemed unsafe. This would indicate that there are other factors involved in the decision making process for pilots that has not been taken into account when the flight route was created. This shows that VIL alone makes for a poor predictor in determining a pilots' decision to deviate or penetrate, as there is some uncertainty in determining whether or not a pilot should fly through based solely on precipitation intensity.

The echo top is another measurement used by dispatchers, and currently is a major factor involved in a majority of weather-related deviations (Kuhn, 2008). The echo top shows how high the precipitation or storm reflects and extends upwards; in other words, it shows the height of the top area of the precipitation, relative to the ground. This should not be confused with the height of the clouds or storms, which is generally higher, as it generally does not contribute towards the storm intensity.

Echo top is of great value in the aviation field, as these heights provide an indication of pilot's intent to penetrate weather, relative to the altitude of the aircraft. This is because echo tops are useful in identifying areas of air currents, where high intensities of air currents would indicate the likelihood of severe turbulence if flown through.

Echo tops alone however cannot identify all severe weather, and are hence interpreted together with other measurements, more notably with VIL. Together with VIL, the intensity of precipitation and the movement of vertical air can be determined, allowing for a more comprehensive overview of the weather severity.

1.3 Problem

The size and quality of the search space presents a limitation on which an algorithm can perform, as larger search spaces in terms of the number of nodes and arcs would increase the memory usage and computational time to find efficient paths.

This project aimed to determine a set of route choices for guiding pilots through convective weather based on the real weather data, and utilised the bi-objective shortest path (BSP) problem to address to find these sets of efficient paths. The flight network in which the aircrafts flew was generated to try and best model the conditions the aircrafts could fly in, in order to achieve the most efficient paths possible. Routes obtained by the BSP program would be compared against different models of realism to determine the flight networks' capabilities in determining the route choices for guiding pilots through convective weather.

1.4 Mathematical Formulation

The problem can be formulated as a bi-objective shortest path problem. The problem can be described as follows:

Let the digraph, or directed network, be represented as G = (N,A), where the set of nodes is N = {1,...,n}, and the set of arcs, or directed edges, is A = $\{(i_1,j_1),...,(i_m,j_n)\}$, which joins the nodes in N. Also, we let P_{ot} represent the set of all possible paths, p, from the origin,o, to the target node, t.

For each arc (i,j) ∈ A, two costs (r_{ij}, d_{ij}) are associated with it, which in this case represents the risk involved and the distance travelled when an aircraft travels on that arc. The distance travelled along a path from the origin node to the destination node can be obtained by summing the length of the arcs, a, and the risk along a path by summing the risk scores.

We wish to minimise the sum of all distances for each arc on a path, and also to minimise the sum of all risk scores along this path. Each path belongs to a set of all possible paths connecting the origin to the destination.

Minimise $d(p) = \sum_{a \in p} d_a$
Minimise $r(p) = \sum_{a \in p} r_a$
subject to $p \in P$

With this, the goal is to obtain routes where it is not possible to obtain one with a better objective value in one aspect without compromising the other. This will allow paths with trade-offs between the most safest and the most direct routes to be found, and subsequently allow pilots to choose generated routes depending on their preference, circumstances, and experiences.

2 Methodology

2.1 Flight Network

The flight network represents the airspace in which the aircraft travels. To model the aircraft as it travels through the airspace, the area in which it travels can be split and discretised into grids of squares. These squares represent a node, or a point, at which the aircraft would be within the airspace at any given time or place. Each of these nodes are connected to other neighbouring nodes by arcs, which represent where the aircraft travels in the airspace from one point to another, like a series of waypoints.

The modelling approach proceeds in two states for the flight network, where each state progressively adds more realism to the problem. Within a 2D environment, we consider the altitude and weather as static; in other words, the plane is flying through static, non-moving weather at a fixed altitude. A model such as this is best used when the aircraft is moving very fast relative to the airspace area, as it is unrealistic otherwise to assume weather being static and not move at all. In a 3D environment, we consider only the weather being static, with the plane being allowed to change its altitude to avoid convective weather.

This setting can be seen as an improvement towards the 2D model, as the aircraft can now change its flight level to compensate for any severe weather it encounters. In this case, real weather data was available for use during the generation of the flight network, however in a real-time planning environment, it should be noted that weather forecasts are utilised.

2.2 Distance and Time measure, d_{ij}

Distance is an important measure in this model. For airlines, the distance their flights have to travel approximately equates to the time it takes to travel to a destination, and also the amount of fuel used. It is in the airlines' best interest to reduce the distance it takes to travel, as this will allow them to save on operating costs, and even on the hourly wages it pays to flight attendants.

2.3 Weather Risk Measure, r_{ij}

Weather, and the risk this imposes to the aircraft in the airspace, is another important measure in this model. The weather for this model is based on real weather data that had been obtained. The airspace in which the weather was obtained from is a set airspace size of 336 by 364 kilometres squared. Each kilometre square has weather data pertaining it with VIL and Echo Top data, measured with weather radar. The VIL and Echo top data represent the different levels of the precipitation and heights of the clouds.

To determine the risk involved for an aircraft travelling within the airspace, another set of weather data is used, which is the occupancy. The occupancy is the likelihood that any aircraft will travel in an area relative to VIL and echo top data in that area. When VIL and Echo Top was relatively high, the occupancy was relatively low, and vice versa. In between, and with an extreme of one particular weather data, the occupancy was more varied in nature, and was not as clear.

To compute the weather risk rij, the VIL, and echo Top values are determined based on the position in the Cartesian coordinate system. From this, the values are then looked up against the occupancy table to determine the likelihood of an aircraft travelling through the area. High occupancy values are given low penalties, and likewise lower occupancy values are given increasingly higher penalties. The penalties themselves are arbitrary values determined to best reflect what is believed the aircraft should be avoiding and not avoiding. With empirical calibration though, it can be expected that the paths obtained would give much better results that have a high correlation with actual flight paths that are tracked.

2.4 Flight Network Generation

The flight network is generated with the use of MATLAB to create and represent the airspace in which the aircraft travels within two dimensional, three dimensional, and four dimensional space. The flight network grid is a necessary component in this model, as this defines the airspace and available nodes in which the aircraft can travel to and from.

To recreate the airspace in which the aircraft travels in 2D, we first take the size of the weather air space into account, as this determines where the aircraft can travel. Afterwards, we use this to set the upper limits onto which the aircraft can travel up to within the airspace.

2.5 Two-Dimensional Space

To generate the flight network in two-dimensional airspace, the following pseudo-code given in algorithm 1 details the basic idea behind the generation of the network.

Algorithm 1 – 2D Flight Network Generation

1. Input – weather data, source and destination, occupancy data, altitude
2. Initialise – Distance values, search parameters, altitude height
3. *for* y = 1 to maximum longitude
4.   *for* x = 1 to maximum latitude
5.   *for* all possible headings (North, NE, East, SE, South, SW, West, NW)

```
            if heading is possible 1 step ahead
                Calculate echo top index
                Calculate weather penalty by looking up occupancy
                Output current (x,y), new (x,y), distance, and weather penalty
            end if
        end for
    end for
end for
```

2.6 Three-Dimensional Space

To generate the flight network in three-dimensional airspace, the inclusion of another nested loop before the longitude in algorithm 1 to take into account the altitude would be required. As a result, the possible headings in which a node can connect to increases as well, as it can go either above or below the current node. However, the speed at which the network was generated would slow down considerably.

The algorithm was improved by reducing the number of nested loops to improve the speed of the network generation. Before the improvement, the generation would have been estimated to take several days for 3D depending on the altitude, and could well take over a couple of weeks for 4D for at least 20 units of time. With the reduction of nested loops to just one loop, the generation of a 3D network dropped drastically to around two hours on a standard Intel Quad-Core computer.

The improvement made was to cycle through all the possible nodes in the airspace, given the size of the weather and altitude, and then re-converting it back to Cartesian co-ordinates, and can be demonstrated in algorithm 2.

Algorithm 2 – 3D Flight Network Generation

```
Input – weather data, source and destination, occupancy data, altitude
Initialise – Distance values, search parameters
for index = 1 to maximum nodes
    Convert to Cartesian co-ordinates
    for all possible altitude levels (current, above and below)
        for all possible headings (North, NE, East, SE, South, SW, West, NW)
            if heading is possible a pre-defined step ahead
                Calculate echo top index
                Calculate weather penalty by looking up occupancy
                Output current (x,y), new (x,y), distance, and weather penalty
            end if
        end for
    end for
end for
```

2.7 Output

The output of the files utilises the Forward and Reverse Star representation to store arcs that come from the nodes into a single array structure. The advantage of this method is that it saves space, is efficient for manipulation, and is suited for dense and sparse networks (Zhan, F., 1998). Each row represents an arc, and the four columns represent the starting node of the arc, the ending node of the arc, and the respective costs for distance travelled and the weather risk at the ending node of the arc.

The size of the problem is dependent on the size of weather data provided. For an airspace of a given size, by increasing the number of weather points available and making the “resolution” of the weather clearer, the number of points increase dramatically. The number of nodes and arcs will increase along with the number of weather data points available, which makes for a more refined path for the plane to follow. For a given problem of the a size of roughly 336x364 with a point of data in between, the total possible number of paths go up to 974236 for a 2D environment.

2.8 Spatial Size

The network size can be reduced, to speed up and simplify the time required to generate the flight network. The end result shows the grid being similar in looks but at a reduced quality.

The cell sizes are discretised such that for each defined area of space in the original airspace, say 3x3 for example, this now becomes 1 square in the smaller version. The underlying VIL and echo top data from the original airspace are also averaged for the new cell, which is averaging the value of the cells. This is one method that can be used to improve computational speed and efficiency.

This effectively reduces the network size, however of note is that the distances have to be compensated by a factor of three for accuracy.

2.9 Bi-objective Shortest Path Algorithm

The bi-objective shortest path (BSP) algorithm is a label setting algorithm, which is an extension of the single objective label setting; Dijkstra's Algorithm. An implementation of this algorithm, in the form of a program by Andrea Raith (Raith & Ehrgott, 2008), was used to help find efficient paths for this model. A number of different algorithms are available for the user to select from when running the BSP program.

The program was coded in the C programming language, and runs on Linux. The parameters the program takes to run is the input file, and a number corresponding to the algorithm to be used, along with any subsequent parameters that need to be specified with the algorithm, if any.

3 Result

The implementation of the weather avoidance system is compared as viable tools for the pilots and dispatchers. The models compared will be the 2D environment, which is with static weather and a fixed altitude, the 3D environment, which is with static weather.

The input weather given for this comparison of the program is of the same weather data as the 2D to provide a grasp of the differences the algorithm has to account for with the inclusion of altitude.

It can be noted that the algorithm produces paths that are in line with the 2D paths generated, however, as the emphasis on safety increases, the utilization of altitude becomes more apparent.

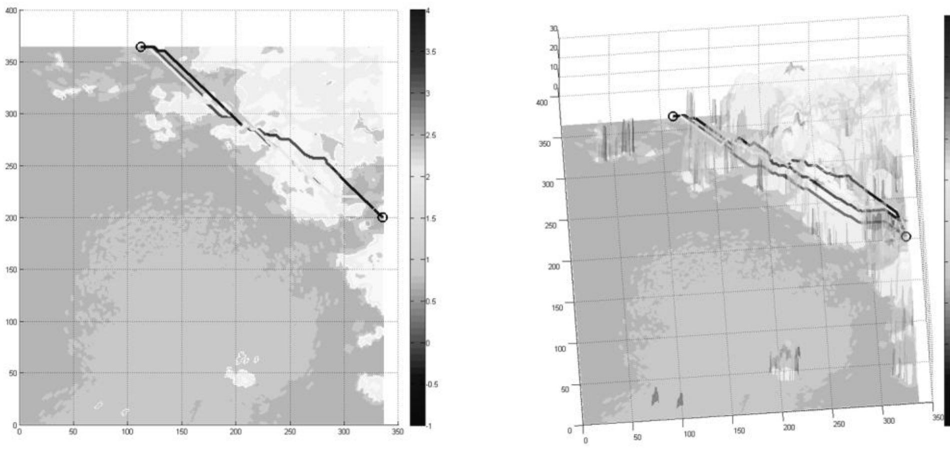


Figure 1. Representation of weather over an airspace, with VIL data over echo top data for cloud heights and severity, and with efficient flight paths overlaid on top.

It can be seen that from figure 1, the paths follow a similar path from the origin to its destination. When looked at from a slightly angled view of the paths, the routes show differences in altitude. Those closer to the ground show more emphasis towards the shortest distance, whereas those higher up, in an attempt to fly above the clouds, show more emphasis towards safety.

What is peculiar is that the paths do not ever leave the weather at all. It can be deduced that once a particular deviation is reached, anything above that will result in a compromise in the distance travelled that is so great that it outweighs the distance.

4 Conclusion

The project aimed to determine a set of route choices for guiding pilots through convective weather based on the available weather data given. This was modelled as a bi-objective problem in order to give pilots the freedom of choice of routes between minimizing risk and maximising efficiency. In particular, flight networks of a two-dimensional, and three-dimensional setting were explored to help achieve this.

5 Future Work

Future work expands to a wide variety of possibilities which have yet to be explored, such as inclusion of aircraft performance constraints like angles of attack and headings for realism, the implementation of 4D with respect to time for dynamically changing weather and comparing actual flight paths to efficient paths obtained with the program. Further work would also include improving the solution algorithm and allowing multiple aircrafts to be routed in the same airspace, instead of just one.

Acknowledgments

I would like to acknowledge Andrea Raith and Kenneth Kuhn for the tremendous support and help that they have given me throughout the year. Additional thanks to Christopher Vogel, David Ryan, Imran Ishrat, and Ming Cen for their help on aviation and simulation related problems.

6 References

Kuhn, K. (2008). "Analysis of Thunderstorm Effects on Aggregated Aircraft Trajectories." Journal of Aerospace Computing, Information, and Communication 5, 108-119 (2008).

Raith, A. and M. Ehrgott (2009)."A comparison of solution strategies for biobjective shortest path problems." Computers & Operations Research 36(4): 1299-1331.

Zhan, F., (1998). "Unit 064 - Representing Networks" Retrieved 8th August 2010, from http://www.ncgia.ucsb.edu/giscc/units/u064/

Design of Road Networks

Jason D. Drake
Department of Engineering Science
University of Auckland
New Zealand
jdra024@aucklanduni.ac.nz

Abstract

One of the major problems faced by residents of Auckland is traffic, especially traffic congestion. The process of designing road networks and improvements that help alleviate congestion and shorten travel times is a difficult task, but one that can be addressed through the use of traffic models and traffic assignment techniques.

This project focuses on the use of these techniques, and in particular applies them to the road network north of Auckland in order to analyse the performance of the network on public holidays. On public holidays a large increase in demand for travel north causes major congestion on the two primary routes leading north, being state highways 1 and 16, with the majority of travellers choosing the highly congested state highway 1 over the less congested state highway 16.

Through the application of traffic modelling techniques and by comparisons between model results and real world traffic flows, we make conclusions about the performance of the road network north of Auckland in regards to the goals of the user and of network planners, and evaluate potential solutions for reducing congestion on state highway 1.

Key words: Road networks, network design, traffic prediction, traffic assignment, user equilibrium system equilibrium.

1 Introduction

One of the major problems faced by residents of Auckland is traffic, especially traffic congestion. The process of designing road networks and improvements that help alleviate congestion and shorten travel times is a difficult task, but through the use of traffic models and traffic assignment techniques we can begin to address the issues surrounding the design of transportation networks.

Of particular importance are the times of peak demand where a large amount of traffic is placed on the road network, such as during special events or holidays. These spikes in demand cause heavy congestion on roads which do not have the capacity required to facilitate demand in peak periods.

An example of such a scenario is the situation of the state highways north of Auckland, where the large surge of demand on long weekends from people wishing to travel north to Warkworth, Wellsford and further. This peak demand causes major congestion on the two primary routes leading north, being state highways 1 and 16, henceforth referred to as SH1 and SH16 respectively. In particular congestion is highest on SH1, with the majority of travellers choosing it over the longer but less congested SH16 (Traffic count data, 2008).

It is the opinion of the New Zealand Transport Agency (NZTA) that the usage of SH16 during these times is not sufficiently high given the heavy congestion present on SH1, and as such NZTA often runs advertisements on radio and in print in the days prior to a long weekend encouraging motorists to travel on the less congested alternative of SH16 in order to reduce the congestion experienced on SH1.

The focus of this project was to investigate the performance of SH1 and SH16 in these heavy congestion scenarios, which was accomplished through the application of traffic forecasting models and techniques; these methods allow a transportation network to be evaluated under particular demand loads by predicting the relative usage of the elements of the network. For this project these elements are the roads and highways of the network, but models may be extended to include aspects such as rail and ferry routes.

By predicting road usage under different assumptions about user behaviour and comparing the resulting traffic flows against observed real world flows, we can make conclusions about the performance of the road network north of Auckland in regards to the goals of the user and of the network planners.

2 Traffic Forecasting

The goal of traffic forecasting is to predict the relative usage of elements of a transport network, typically roads, given a specified demand on the network. This is achieved through forming a computer model that represents the transport network, and using special algorithms to accurately map the demand on the system into the resulting flows that would be observed in a real-world situation.

The ability for transportation scenarios to be modelled and performance evaluated has many practical applications. Applications include: analyzing user behaviour; evaluating performance of current systems under current or future demand; or evaluating modifications to transport infrastructure or policy and quantifiably assessing the relative performance of the changes.

2.1 Aspects of the Computer Model

2.1.1 Network

The main component of the computer model is the transportation network itself. The transportation network is represented in the model by a directed graph, where the nodes of the graph represent locations such as intersections and suburbs. The arcs in the graph represent a mode of travel between two locations, and are commonly referred to as links in transport literature. For this project these links will represent roads, but transportation models are commonly extended to include other modes of transport such as rail or ferry when such modes are deemed important.

Formally, the network is represented by a graph $G = (V, A)$ where $V = \{1, \dots, n\}$ denotes the n nodes of the network and $A \subseteq V \times V$ is the set of directed arcs between nodes.

2.1.2 Link Cost

In addition to representing the physical layout of the transport network and the modes of transport between locations, the model must also consider the cost of travel on each of the arcs in the network. This cost of travel will form the basis for user decision making when solving.

To model this we associate a cost function c_a for each arc $a \in A$ in the network reflecting the cost of travel. For a typical network of roads the cost of travelling on a particular road is given by the travel time. This is generally given as a function of the traffic quantity on the road, denoted by v_a.

2.1.3 Demand

As stated at the beginning of section 2 the goal of traffic forecasting is to predict the usage of parts of a transport network under a particular loading; this loading is the demand of the road users and forms a large part of the modelling process.

Demand in this case is stated at a basic level, where for pairs of locations we state the quantity of people who wish to travel from the first location to the second. Generally a set of special nodes called centroids are selected from or added to the model, these centroids represent particular areas of the network, and are the primary endpoints for the set of demands in the model.

The pairs of locations used in the statement of demand are termed origin-destination (OD) pairs, the set of which we denote by W. For each $w \in W$ we have an associated demand between the locations of d_w. Of note is that this formulation is independent of the links in the network, in that it does not state or restrict the links that will be used to meet the demand. The decisions of what links will be used are instead carried out by the solver.

2.2 The Four Stage Modelling Process

Traditionally when attempting to model a transportation scenario a four-stage process is followed. Initially developed during the 1950s for use in the Chicago Area Transport Study (Black, 1990), the four-stage process is the basis for most modern techniques (Ortuzar & Willumsen, 2001).

The four stages are *trip generation*, where demand is estimated between regions of the model. Region-based demand is then broken down into specific origin-destination demand pairs in the *trip distribution* stage. The modes of transport to be used for each origin-destination pair is then modelled in the *mode choice* phase. In the final stage the demand for each origin-destination pair is assigned to routes in the network via a route selection algorithm, resulting in a set of traffic flows for every link in the network. These results may then be fed back into earlier stages to refine the input parameters used.

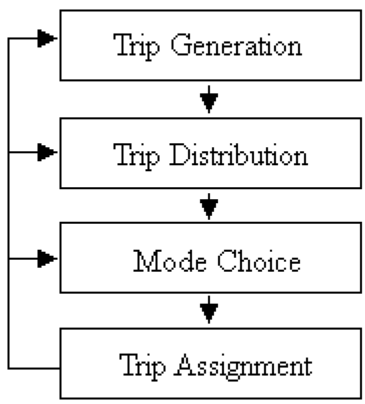


Figure 1. The four stage modelling process

2.2.1 Trip Generation

Trip generation is the initial stage, and involves examining the regions of interest and for each region determining the frequency of trips into and out of the region. This information is based on the aggregate data of the particular region, with characteristics such as population size, population demographics, and economic factors taken into consideration.

2.2.2 Trip Distribution

Trip distribution takes the inter-region demands produced by the trip generation stage and refines the demand locations to produce origin-destination pairs for use in solving. This is accomplished by considering the density of population and industry within the zone, and proportionally assigning demand endpoints in line with these densities.

2.2.3 Mode Choice

The mode choice stage takes the demand quantities for each origin-destination pair found in the trip distribution stage and decides what fractions will be satisfied by each mode of transportation, commonly automotive, bus, rail, or ferry.

2.2.4 Trip Assignment

Trip assignment is the fourth stage in the four stage transportation modelling process. In this stage the completed demand information from previous stages is mapped onto the transportation network to get the resultant traffic flows.

The main aspect of trip assignment is modelling what routes users take through the network to reach their destinations. This is a complex task, requiring careful formulation of user behaviour and goals such that the results produced accurately reflect the desired behaviour.

2.2.5 Flow back

After completing any of the four stages of the modelling process, results from that stage may be used to refine or recalibrate the input parameters or methods of previous stages. An example of such a flow back of information is from the trip assignment phase back to the mode choice phase. After the trip assignment phase has run, the travel times and other costs of the links in the network can be determined, these updated costs may result in people choosing alternative modes of transport.

3 Traffic Assignment

The focus of this project is on the traffic (or trip) assignment stage of the typical four-stage modelling process. By focusing on this stage of the modelling process we hope to be able to analyze the situation of SH1 and SH16 and the choices users make, including reasoning about the causes of such choices, in addition to drawing conclusions about the state of the northern highway network in regards to network design goals, which may differ from the goals of the individual user.

When attempting to predict the flows that would happen in the real world, we need to consider the behaviour of the users who are travelling through the network. In particular we need to consider what criteria the users use to evaluate different routes to achieve their goal of arriving at their destination.

3.1 Generalized Cost

The travel time on a particular link is one of many factors that influence the route choice of a user. While the travel time is the typically the dominating factor the combination of other factors can be significant enough to produce inaccuracies in results if not considered. Examples of such factors include distance, road type, safety, and road tolls.

To incorporate the factors present in users' route choice we introduce the concept of a generalized cost for each link. This generalized cost is formed by weighting and aggregating the various factors determined to be important in the users' perception of the cost of the link, and may be a function of other variables such as the traffic flow or traffic type.

3.2 User Behaviour

The next consideration is, given the perceived cost of each link, what decisions would a user make in regards to route choice. When attempting to model real life situations we generally model the users as selfish, taking the shortest route available to them, with the assumption that the user knows the travel time of each road at that instant, either through experience or knowledge of typical road conditions.

3.3 Equilibrium

Given that each user takes the shortest path available to them, we observe that there will always be a motivation for users to switch from the path they were assigned to as long as a lower cost path is available, and thus in a stable solution it must be the case that for all users the cost of alternative routes is greater or equal to their currently assigned route.

3.4 System Optimality

So far we have only considered 'user equilibrium', that is the equilibrium resulting from users choosing the shortest path available to them. However, there exists another equilibrium commonly used, termed the 'system equilibrium'. This equilibrium represents the state of minimum total travel time, where no collection of users can switch routes to reduce the total travel time.

This equilibrium state is of interest to network planners as they wish for the traffic flows to be close to this minimal travel time state. The differences between the system equilibrium flows and the real world or user equilibrium flows can help planners identify areas of the network which need improvement or modification to motivate users to switch to more system-optimal routes.

The way this system equilibrium is formulated is similar to that of the user equilibrium, with the primary change being that link costs are modified to reflect not only the cost incurred by the next user on the link, but to also include the extra cost incurred by the existing users on the link due to the increased congestion caused by the additional user.

$$c_a^{SE} = c_a(v_a) + v_a * \frac{\delta c_a}{\delta v}$$

The above equation shows how the cost c_a for an arc a is modified in a system equilibrium formulation when under a traffic volume of v_a, with the additional derivative term representing the increased congestion for exiting users.

4 Model Creation

This section details the steps that were required to build the model of Auckland's transport infrastructure, and to produce the demands on the network during the long weekend period. This process involved three main areas: the network, the demands, and the solver.

4.1 Network

The first step of the forming the model was to create or obtain the network representing Auckland's roads. The network we used in the project was based on the regional transport models developed and maintained by the Auckland regional council.

The current model in use by the Auckland regional council is the Auckland regional transport model 3 (ART3), shown in Figure 2. Consisting of 7,665 nodes and 14,752 links and including the Auckland region from Wellsford south to Papakura (ART3 Documentation, 2010). The ART3 model is the most detailed model available, in addition to containing the most accurate road cost functions, and as a result is the primary source for network data.

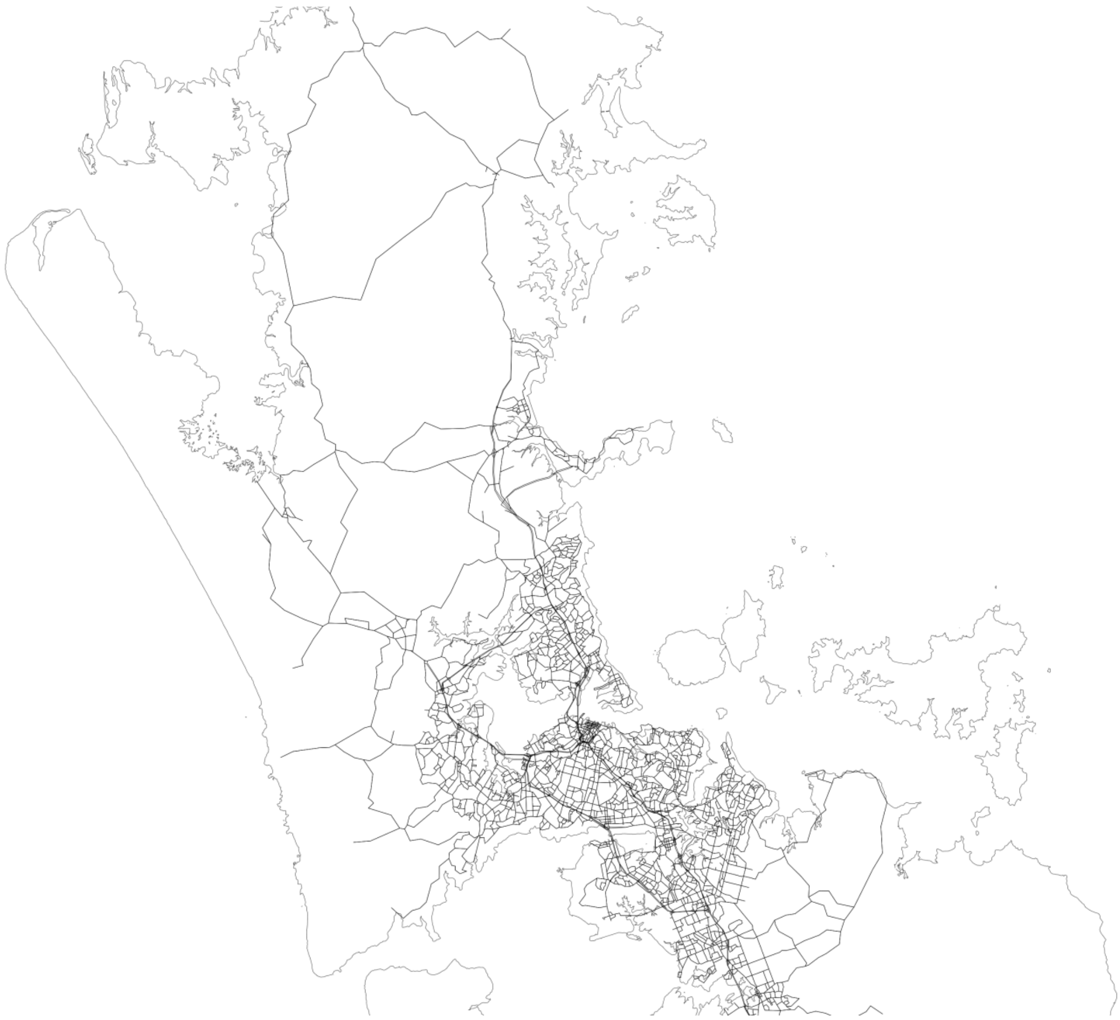


Figure 2. ART3 network

4.1.1 Computational Issues

When attempting to solve the full ART3 model with our solver we encountered numerous computational issues. The primary issue was that the large number of nodes and arcs present in the ART3 model were preventing the model being solved due to memory limitations.

Due to these computational issues, compromises had to be made with regards to the model size. To accomplish this we combined the ART3 with its predecessor, the ART2 model, shown in Figure 2. The ART2 model is much less detailed than the ART3 model, containing just 2353 nodes and 5753 links. This allowed us to considerably reduce the number of nodes in the network, while maintaining the accuracy of the network representation.

4.2 Demand

The next stage was to assess the demand on the model. The first consideration to be made was the period for which the demand is to be obtained. The focus of this project is how the network handles the demand spikes of events and public holidays, so the first stage involved analyzing hourly-count data provided by Sinclair Knight Mertz Ltd. This hourly-count data was collected from electric coils placed under the surface of the road which provides accurate data about the number of vehicles passing a particular point.

From the hourly count data for the year 2008, it was possible to identify the days of peak demand, and for each day the magnitude and pattern of that demand. From this the Auckland anniversary long weekend starting on the 26th of January was found to result in demand representative of the spike-demand produced at the beginning of most holiday periods.

For the project the inter-peak demand figures, representing the period from 9 am to 3 pm, were chosen to most accurately represent the state of Auckland’s central roads during a public holiday periods, given that the inter-peak demand is evenly spread across the network, without large demand into or out of the CBD which has the potential to skew results.

5 Solver

The next stage of the modelling process, after obtaining the base network and the demands required, is the solving process. This involves the selection and implementation of a route choice algorithm to determine the resulting flows of a scenario. The solver used in the project was one written by Dr Andrea Raith, of the University of Auckland, this solver is based on utilizing repeated iterations to converge the solution towards the equilibrium state. It does this by examining each origin-destination pair and the routes which the corresponding demand is assigned to.

By identifying the shortest and longest route that the demand uses, flow may be transferred from the longest route onto the shortest route until the costs for each route become equal. This approach makes use of the observation that at equilibrium the costs of the routes used are equal, thus by identifying routes whose times are not equal and modifying our solution such that the costs become the same we iteratively drive the solution towards the equilibrium state.

This approach was first proposed by Dafermos & Sparrow (1969), and its convergence is proved in the same article, based on the principle that each time we equilibrate two paths we decrease the overall objective function value.

6 Validation

With the network constructed, demand figures obtained, and a method with which to solve for resulting flows, the next step was to validate the entire model. This was done by solving for user equilibrium, the state that is most likely to represent real life user behaviour and flows. This data was then compared to observed flow count data.

Results from the model indicated a flow of 1280 cars/hour going north on SH1, with 370 cars/hour using SH16. This was found to be consistent with the observed flow counts at data sites on state highways 1 and 16, with counts of 1250 cars/hour and 400 cars/hour respectively. In addition to validating against the main routes of SH1 and SH16, count data was also matched to model figures at sites between Warkworth and Wellford, and on SH1 from Wellsford north to Whangarei, the model was found to closely match collected data at these sites as well.

As an additional validation step the travel times to Wellsford and Warkworth from Auckland as calculated by the model were compared against figures published by the New Zealand Transport Agency (NZTA Website, 2010). The times reported by the NZTA for users on public holidays using the state highways were found to be consistent with the times reported by the model.

7 Results

To investigate the claim that SH16 is underused the baseline counts were compared to a modelled situation in which users are modelled as cooperating to achieve the minimum overall travel time, with particular users taking routes slower than their personal optimal route in order to benefit others drivers greater than their individual loss. This minimization of overall travel time is the optimal goal of transportation planners, so by comparing this result to the baseline result and observing the differences we can make conclusions on the performance of the network from a planning point of view, and identify areas that warrant attention or improvement.

When solving with users seeking to minimize total travel time, it was found that the usage of SH16 increased by 235 cars/hour to 605 cars/hour, with the usage of SH1 decreasing by a similar amount to 1025 cars/hour. This result confirms the opinion of NZTA that SH16 is underused, and that users switching from the primary route of SH1 to the alternative of SH16 would result in a lower total travel time for users travelling north on a public holiday.

With the model having confirmed that there is a net benefit for drivers if more drivers were to switch to SH16, we began investigating the situation of SH16 attempting to determine the main factors that are causing its underuse.

The first factor considered was that SH16 was disadvantaged by being far west, requiring drivers originating from central to eastern Auckland to travel cross-city before travelling north via SH16. Given that the majority of the demand originates at non-west locations, it is possible that the additional travel time from travelling cross-city makes SH16 unfavourable relative to the centrally located SH1 alternative.

To investigate this theory we modified the model, doubling the lanes of SH16 in the inner region of Auckland, this doubling of lanes doubles the practical capacity of SH16 which reduces the level of congestion and results in faster travel times. These changes to SH16 match changes currently proposed by NZTA in a project aiming to upgrade SH16 between St Lukes and Westgate, where the proposed changes include a doubling of the number of lanes, so by solving for this situation we also gain insight into how the network will perform if the proposed changes are undertaken in the future.

After solving for the user equilibrium we found that the usage of SH16 as a route north remained largely unchanged, in addition the usage of SH1 was also unchanged. This result indicates that the cross-city travel time is not a primary factor in users' decisions between taking SH1 or SH16 north. This lead us into the next potential factor, that the reason people do not choose SH16 is due to the performance of SH16 itself. In particular the fact that portions of SH16 consist of a single lane, which causes issues with passing and hence congestion can occur when slow moving vehicles such as trucks are present.

To evaluate whether the performance of SH16 itself is the cause for its underuse we solved the model after doubling the number of lanes on the entirety of SH16, from central Auckland north to Wellsford. Solving for this scenario showed a small increase of 30 cars/hour on SH16 for the user equilibrium model, and a larger increase of 40 cars/hour for the system equilibrium. While these results show an increase in usage of SH16 when extra lanes are added, they magnitudes of the increases indicate that the performance and hence usage of SH16 is not tied to its performance under congestion, but rather the larger distance that SH16 covers over the much shorter SH1, 80 km and 60 km respectively, especially for demand destined for Warkworth.

To determine whether the bottleneck in SH16s performance is due to the distance alone, over any congestion factors, the model was adjusted to give the highway 50 lanes in each direction, completely eliminating congestion effects from the model. When the model was rerun with these parameters and compared to the original model representing the current state of SH16, an increase of only 50 cars/hour on SH16 was observed.

Figure 3. Results

	User Equilibrium		System Equilibrium	
Scenario	SH1	SH16	SH1	SH16
Base counts	1250.0	400.0	-	-
Normal model	1225.3	369.3	994.7	601.3
2-lane SH16 inner-city only	1208.7	352.7	981.3	614.7
2-lane SH16	1192.0	403.3	956.7	637.3
50-lane SH16	1177.3	417.3	950.7	643.3

8 Conclusions

The results gathered from the model indicate that road users travelling north behave with the goal of minimizing their personal travel time, as expected, and that while SH16 is relatively uncongested on public holidays users still prefer to use the heavily congested SH1.

The reason for users choosing the congested SH1 over SH16 is due to SH1 being much shorter than the alternative SH16, which requires drivers to also travel cross-city and in the case of those travelling to Warkworth on SH16 requires them to use SH1 from Wellsford to Warkworth if they wish to stay on highway routes.

The users who do choose to use SH16 were found to be primarily from the far west region of Auckland, illustrated in Figure 15 and Figure 16, where the cross-city travel time penalty is moved from SH16 to SH1. The importance of this is that the users SH16 appeals to consist only of those living very close by, and that the extra distance incurred by those living away from SH16 combined with the already long length of the highway renders attracting additional users through improving SH16 infeasible.

Looking forward into future improvements to the road network north of Auckland, with the goal of reducing congestion on SH1, the primary candidates are either upgrading SH1 itself to raise its capacity and hence reduce congestion, or to introduce a new highway north of Auckland that either runs parallel to SH1, or is based in the west with the goal of cutting off long sections of SH16 with a more straight alternative.

9 Acknowledgements

I would like to thank my supervisors Dr Andrea Raith and Dr Michael O'Sullivan of the University of Auckland for their advice and guidance on this project. I would also like to thank the people of the New Zealand Transport Agency for their interest and assistance, in particular Mieszko Iwaskow and Paul Glucina who provided the contacts necessary for the completion of the project. In addition I would also like to thank Richard Hancy, Minesh Lal, and Jarrod Darlington of Sinclair Knight Merz consulting group for their help providing necessary transport data. Thanks also to the Auckland Regional Transport Authority, whose development of Auckland transport models made this project possible, and to John Davies from the Auckland Regional Council for permission to use the Auckland transport models.

References

ART3 Documentation. 2010. Retrieved from website http://www.arc.govt.nz/transport/transport---strategies-and-documents/transport-model-development.cfm

Beckmann, M., McGuire, C.B., & Winsten, C.B.. 1956. *Studies in the economics of transportation*. Yale University Press.

Black, A. 1990. The Chicago Area Transportation Study: A Case Study of Rational Planning. *Journal of Planning Education and Research*, vol. 10 no. 1 p. 27-37 doi: 10.1177/0739456X9001000105

Dafermos, S., & Sparrow, F.T. 1969. The traffic assignment problem for a general network. *Journal of Research of the National Bureau of Standards-B. Mathematical Sciences*, 73B:91-118.

NZTA Website. 2010. Retrieved from website http://www.tollroad.govt.nz/

Ortuzar, J., & Willumsen, L. 2001. *Modelling Transport*, 3rd Edition, John Wiley.

Bi-objective Cycle Route Finding

Simon Bull, Denis Helm
Department of Engineering Science
University of Auckland
New Zealand
s.bull@auckland.ac.nz
dhel010@aucklanduni.ac.nz

Abstract

Selecting a route to cycle from one point to another requires considering several often conflicting factors. We considered two objectives – the distance of a certain path and a measure of its "attractiveness", which we compiled as a linear combination of several factors. Using freely available data and bi-objective minimization techniques we developed a web application that gives users a range of different options for selecting a cycle route representing the different trade offs between as short a route as possible and as attractive a route as possible.

Key words: Bi-objective, multi-objective, shortest path, cycling.

1 Introduction

Cyclists in Auckland and throughout many cities in the world are a rarity. This is often due to safety concerns as well as a lack of information about cycle routes. High traffic volumes, irresponsible vehicle drivers and the general lack of cycle ways are some of the factors that keep cyclists off the road (Harkey, Reinfurt, and Knuiman 1998). The need for safe cycle journeys is a requirement before cycling can be promoted as a commuting alternative. Hence, Auckland City Council must identify methods that improve cycling safety if it wants to see an increase in people choosing cycling as an alternative travel option (City 2009).

This paper discusses the implementation of a cycle web application that aids users with finding suitable routes from A to B within the Auckland region. It utilizes operations research techniques to do so. This is a practical application of the bi-objective optimization algorithm. Instead of finding the shortest path several factors are taken into consideration to find multiple best routes. The cyclist can then choose the best route based on his skills and experience. Factors that define the suitability of a route such as safety have been quantified within an index. This index is based on several road features which make up for the attractiveness of a route. The bi-objective algorithm then optimizes the suitability of a route and minimizes the route distance at the same time (Raith et al. 2009). The resulting routes are not necessarily the safest or the shortest, but have a mathematically significant compromise between the index and the distance. The results are then displayed in layers on top of a map. This allows the users to easily select the route that they think is best for them.

This approach will give existing cyclists and new cyclists a larger set of information about cycling routes. The information gives them the option to improve their safety by choosing routes deemed more appropriate, even though there might be a compromise in distance. Safer route options will lead to a more cycle-friendly environment and a possible increase in the proportion of cyclists (City 2009). Further to this, information about route choices can be of value to city councils. Evaluating the attractiveness of cycle routes in different areas of a city could help identify areas that only have unattractive routes and need better cycling facilities. This information could be of value for an economic evaluation for cycle network improvements (Raith et al. 2009).

The development of the cycle route finding application for the Auckland region will be explained. Initially the quantification of road suitability will be made clear and the data sourcing necessary for the algorithm will be laid out. Further, the graphical representation of the cycle network as well as the implementation of the algorithm is described.

2 Road Attractiveness Index (RAI)

The routes cyclists choose differ from route choice drivers of private vehicles have. Commuter drivers tend to choose routes with low travel times and low vehicles operating costs. Commuter cyclists on the other hand have multiple objectives when a route is chosen (Aultmann-Hall, Hall, and Baetz 1997). The travel time and the road suitability are factors influencing the cyclist the most. In order to obtain optimal route choices with the aid of an algorithm these factors have to be quantified (Raith et al. 2009).

The travel time of a cyclist is mostly influenced by the travel distance; typically shorter distances will lead to shorter travel times. The suitability of the road however is in itself an index of multiple factors. There are multiple papers that discuss the important factors influencing cyclist road choices. The development of the bicycle compatibility index (BCI) is a method that allows quantifying the suitability of roads (Landis et al. 1997). The Level of Service (LOS) assessment is another method that helps to quantify attractiveness. Each of these methods has its advantages and disadvantages. One of the disadvantages of BCI is that low traffic volume environments as well as significant intersections are not accounted for, but under the LOS they are. Hence a mix of both these methods has been applied to find attractiveness data in this assessment (Zealand 2005). The bi-objective optimization algorithm tries to minimize the distance as well as a term defined as the "un-attractiveness" of a road. This term is in fact an inverse of the attractiveness of a road. Several factors have been identified in literature to influence the attractiveness of a road. Cyclists are concerned with safety and safer routes tend to be more attractive. Factors such as the roadway traffic volume, the total width of the outside through lane, speed limit, driveway density, type of road surface, sidewalks, number of lanes and the size of the intersection will influence cyclists choices (Harkey, Reinfurt, and Knuiman 1998). Further to this the existence of cycle ways and bus lanes that can be shared with cyclists will make for a more attractive route (Raith et al. 2009). Each of these factors is considered for a road segment and given a ranking. As an example the width of a side lane of a road is measured. 20 points are given for a width greater than 2 metres and zero points for a width of zero metres. All widths in between these distances are scaled linearly. Every road segment is analysed in this manner. Similar procedures allow obtaining points for the traffic volume and fixed amount of points are given for the existence of cycle lanes or bus lanes. Each of the factors is scaled in proportion to their importance of the overall attractiveness. A summation of these factors results in the RAI (Harkey, Reinfurt, and Knuiman 1998).

In future work a survey of cyclists and commuters wanting to become cyclists could allow identifying factors Auckland citizens find especially important in determining route choices. The studies that have been used to identify the important factors are based on US as well as Canadian cities. New Zealands perception of attractiveness might however differ due to different driver behaviours and geography.

3 Data Sourcing

A sample application has been developed for the Auckland region. Data needed to be collected to compute the attractiveness for the road segments. OpenStreetMap (OSM) had been used to obtain road information. OSM is a collaborative project to create a free editable map of the world (http://www.openstreepmap.org). Compared to other websites such as Google Maps, OSM data can be freely downloaded and used to create applications. Road segments have street names as well as additional information such as road types (primary, secondary, residential, motorway etc.). This information has been provided by contributors in a similar fashion to Wikipedia contributors. However additional information such as the traffic volume density is not provided by OSM. In order to obtain a working application this additional data had been collected in a more manual way. The Auckland City Council has provided data for traffic volumes of major roads within the Auckland region and could be used (Council 2009). Further to this Auckland City Council tools such as a high quality aerial map of Auckland (Council 2010) have helped to measure such things as road width and number of lanes of a road. These measurements however often had to be completed on a manual basis and as a result of this only the Auckland region has been considered with the Ponsonby area having the largest set of details of road information. Other areas have partly been grouped into residential, primary and secondary roads and given a respective attractiveness score. All the extracted information is then used to calculate the RAI and store it within a database for further use in the optimization algorithm.

The acquisition of data from multiple sources proved to be manually intensive and in order to obtain accurate attractiveness values for other cities a more streamlined method and easily accessible data will be required.

4 Problem Formulation

For any given section of road there are two distinct measures – the length of the section and its attractiveness. It would be possible to combine these into one objective and find a single best path through the graph, but the manner in which they are combined will vary from user to user, and even for a given user with certain preferences the combination that gives the best route may differ for different start and end points in the same network. To address these issues and keep the choice with the users themselves we used bi-objective optimization to find multiple best paths, rather than a single best path.

Consider a path P, made up of segments. Each segment in P has a length and an attractiveness. The length of path P is defined as the sum of the lengths of the segments making up the path:

$$d(P) = \sum_{i \in P} d_i$$

The attractiveness of the path made up of distinct segments is taken to be a weighted sum of the attractiveness scores for those segments in such a way that the attractiveness is interpreted as a measure per unit length of road. For a path P the attractiveness is:

$$A(P) = \frac{\sum_{i \in P} A_i \times d_i}{\sum_{i \in P} d_i}$$

In a graph G given a start and end vertex, if there are paths connecting those vertices then it is possible to find the single shortest path, or the single most attractive path. The shortest path is found to be the one with smallest $d(P)$ among all paths connecting the start and end vertices, and the most attractive path is that with maximal $A(P)$. There are, however, paths that are neither the most attractive nor the shortest but which may represent an interesting trade-off between these objectives.

If there are two paths connecting the same vertices, one is said to dominate the other if it is better in both measures. We wish to find all paths with length and attractiveness that are not dominated by any other path. Such paths are said to be efficient; there exist no other paths with both better attractiveness and length. When considering all of the potential paths from the start to end vertices the following is true:

Path P is efficient $\Leftrightarrow \nexists P\prime : (d(P\prime) < d(P) \wedge A(P\prime) > A(P))$

We used an algorithm to find all of these efficient paths given a start end end node (Raith and Ehrgott 2009). However, before it could be applied we had to make two alteration to the attractiveness objective. Firstly, the definition of attractiveness given above is unsuitable for combining. Two attractiveness scores can not be combined without knowing the components that make each up. Secondly, the algorithm we used requires a problem with two objectives to be minimised. We used a modified attractiveness with the denominator removed, and the reversed in it a way that resulted in a minimization problem

$U(P) = A_{max} - \sum_{i \in P} A_i \cdot d_i$

That is, we subtracted each attractiveness from the maximum possible attractiveness to get the unattractiveness. We were then able to use the algorithm on the following problem:

$$\text{minimize} \left\{ \sum_{i \in P} d_i \right\} \qquad \text{minimize} \left\{ A_{max} - \sum_{i \in P} A_i \cdot d_i \right\}$$

This new formulation is not equivalent to the original formulation. However, solutions to this new problem are still relevant, and unlike the original problem the new problem can be solved in a reasonable amount of time (see (Raith and Ehrgott 2009) for details of this formulation and comparison with the original formulation).

With this formulation, the algorithm and an appropriate graph representation of a road network we were able to find all such efficient paths through the road network.

5 Graph Representation

5.1 Full Graph Representation

Road networks are represented in Open Street Maps as nodes, ways and relations. Each node represents a particular location in the world on a road or path, and has a latitude and longitude, but not a height. It is possible for multiple nodes to exist in the same location (for example when a road bridge crosses a different road). Roads and footpaths are represented as ways, which are lists of nodes with accompanying information such as the road or path type.

Together these nodes and ways define a graph, with the nodes representing the vertices of the graph and each way representing a list of edges. Using the latitude and longitude we were able to calculate the distance between two nodes and therefore construct a graph with edge lengths. Though roads can be curved and these were straight-line distances the vertices were close enough together (spaced every few metres along roads) that this method was adequate.

As some of the roads in a road network are one way the network was represented as a directed graph. Therefore most road segments were represented as pairs of forward and backward edges in the graph, as most roads are not one way roads.

A route on the road network is represented as a path in this graph, with the start and end points of the route identified as the vertices in the graph with the closest latitude and longitude.

5.2 Reduced Graph Representation

As the nodes are placed very close together (in some places as close as every metre) the created graph can be very large. We found that for the Auckland road network the resulting graph contained hundreds of thousands of vertices. For our purposes this was too large, and we came up with a method for transforming the graph into a smaller graph representing the same network by collapsing certain vertices. Vertices that are suitable for removal are in a section of road and have exactly two unique neighbours, and can be removed and the edges adjacent to them combined. See Figure 1 for an example of a single vertex on a one way street being removed, and edges adjacent to it being replaced. Note that for distance, the values are simply added. For attractiveness, however, they can not be added together. Instead the RAI are both required and combined (see Section 4).

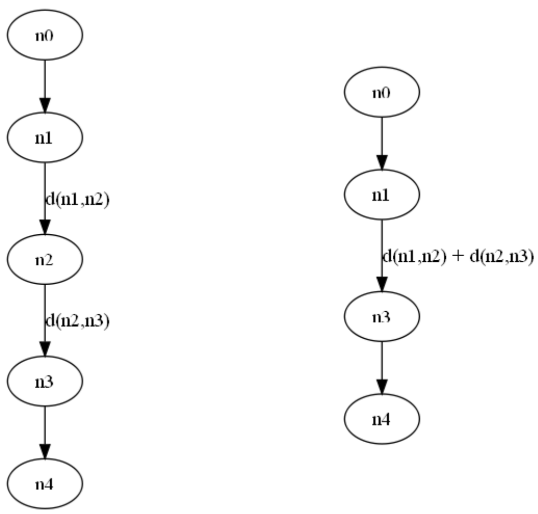


Figure 1: Collapsing a vertex and combining edges

By performing this collapsing in an iterative way all such nodes can be removed. As the removal is done edge lengths are summed. The second objective, attractiveness, is also combined in a similar way when this reduction is performed. As this measure is normalized for length, pieces are combined by calculating their average in terms of length.

Figure 2 shows a segment of road network. All the smaller nodes are candidates for removal, and after those nodes are removed and the edges combined the only nodes left will be the ones marked as larger circles.

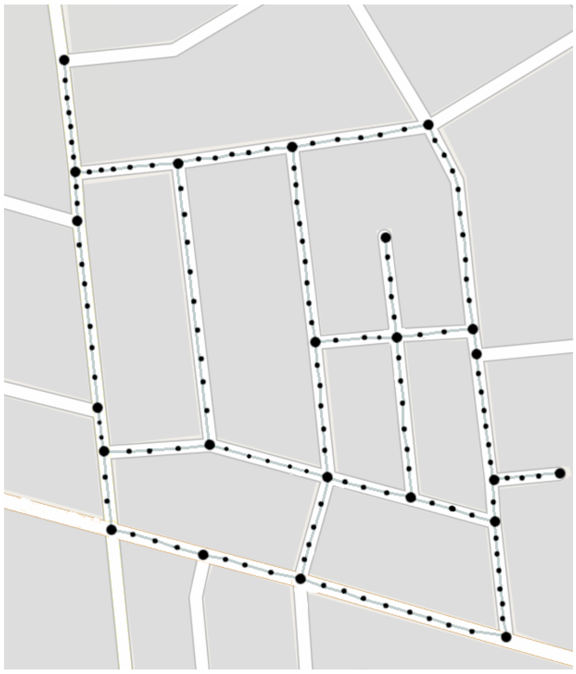


Figure 2: Only those vertices with larger circles remain after reduction

The full graph and the reduced graph are homeomorphic; the full graph is a subdivision of the reduced graph.

Finding a path in the full graph is done as follows:

- Find the closest vertices to the start and end points (in terms of latitude and longitude) in the full graph;
- search from these vertices along edges to the closest vertices that appear in the reduced graph;
- find a path through the reduced graph connecting these identified vertices;
- unpack the removed vertices from the edges to obtain the equivalent path in the full graph;
- add the vertices searched along from the original start and end vertices.

This means that a path can be found in the reduced graph (which has an order of magnitude fewer vertices than the full graph), and the full path can be reconstructed. The path in the full graph is necessary to fully describe the path geographically, but the path in the reduced graph is sufficient to uniquely identify the path.

6 Application

With the algorithm and a method of creating a graph from Open Street Maps data and calculating the distance and attractiveness scores, we were able to find all efficient paths given a start and end location. As a practical application we created a web application using the entire Auckland road network. The application has an interface similar to Google Maps, allowing zooming and panning. Users indicate the beginning and end of their journey and several potential paths are overlayed on the map. There is a small delay between the user requesting the routes and the results being displayed, but it is not significant when compared with other similar route finding websites such as Google Maps.

A problem we identified with this large network was that there are a very large number of efficient paths when the start and end vertices are far apart. If they are close to each together there may be between one and ten efficient paths, depending in the location, but when they are at opposite ends of the city there may be hundreds of efficient paths. Presenting this many paths to a user is unreasonable, as there is too much information to present in an accessible way.

6.1 Limiting The Number Of Paths

When the number of paths found is small, the web application returns and displays them all. However when it is more than a certain number (which is customizable) the number of displayed paths is limited.

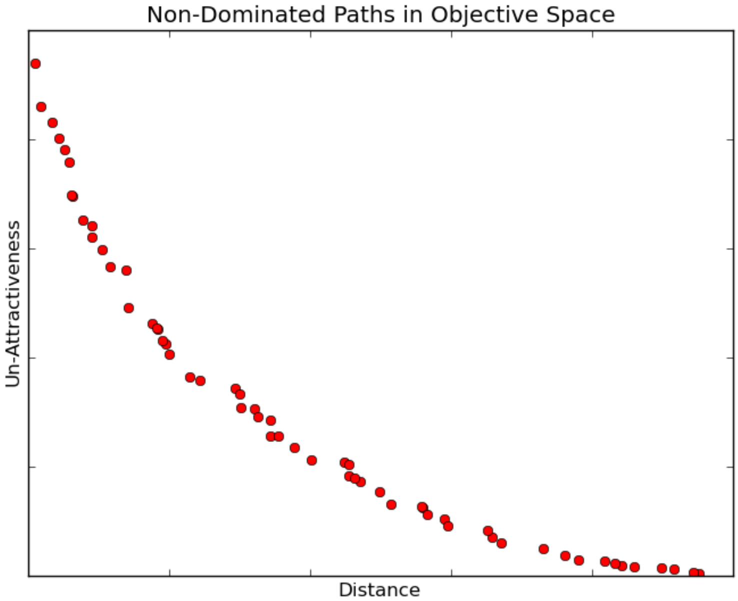


Figure 3: Example of the Range of paths found in a large network.

Figure 3 shows an example of the range of solutions returned when the start and end vertices are far away. We observed that in most cases there were two or three distinct major routes, each with many variations. For example, a major route may consist of a series of segments such that for each segment there are two choices of road to take. If there are n segments then there are 2^n different variations on the route. It is unlikely that they are *all* efficient (though it is possible), but in some cases many of them are efficient. Ideally our application would select just one or two different paths for each of these major routes and present those to the users.

The method we used to do this was select a subset of the efficient paths based on their objective functions, so that the returned paths are evenly spaced in objective space. In some cases this works well; all of the variations on one major route are very short but unattractive, all of the variations on another are long but attractive etc.

However, in other cases these variations on major routes can result in significantly different objectives (an example is a straight main road and a parallel winding minor road; the straight road is short but unattractive, while the parallel road is long but more attractive). As we did not have a precise definition of what a good subset of paths would be, we used the method described above involving objectives to select some of the paths. In practice it gave results that were adequate, but more research could be done in this area.

7 Future Work

There are a lot of expansions and improvements possible to the current version of the application. A number of points that can be considered in future versions of the application are described here.

7.1 Geographical Height Data

New Zealand and especially Auckland terrain has a lot of hills. Most commuting cyclists would prefer to take a route that does not contain any steep roads. Further, routes that are shorter in distance, but contain unnecessary hill crossings are undesirable by cyclists. Hence, a trade off between height differences and route distances is required to find the optimal route.

The current version of the application does not take heights into consideration. If appropriate data were available this function could be integrated into the model and be part of the definition of attractiveness. Hence the bi-objective optimization model would take into consideration the slope of roads when choosing routes and would provide users with more appropriate route choices.

7.2 Spatially Different Paths

As described above in Section 6.1 a different method of selecting a subset of the paths to display to the user could be investigated. Ideally the routes displayed would be spatially different to provide the user with a good range of choices. It may be possible to select these from the potential routes returned, or it may be necessary to approach the problem in a different manner and search the graph for spatially graphs directly.

Alternatively or additionally ways of displaying more of the information to the user may be investigated. Of the hundreds of efficient paths found currently only a few are used. Potentially the user could select one of the few routes returned and be given more routes that are spatially similar to that one but different in objective, allowing them to explore the solution space and refine the available options.

7.3 Automating Data Collection

As described above in Section 3 a more streamlined data collection method would be desirable. Currently public data is only available sparsely. Easily accessible data would allow automating the data collection and attractiveness calculation procedures. Some public data is available at City Councils. However if this data is not in electronic formats or is hard to access, voluntary data collection may be required. Methods similar to those used by Open Street Maps that rely on entirely voluntary data collection could be used to obtain relevant cycling data.

7.4 City Council Information

Expansions to the current application can be of great benefit to city councils. The application can be used to gather cycle information useful for cycle network improvements. As an example the application could store a history of the most desirable routes chosen by cyclists as well as the volume of cyclists using these routes. This information can then be used by city councils to make decision about future expansions of the cycle network in Auckland. This method will be a cost efficient indicator of favourable cycle routes. The attractiveness data could also indicate which city areas would need more improvements in order to create more attractive route choices.

8 Conclusion

Cycling is an activity councils such as the Auckland City Council try to promote as an alternative and sustainable method of transportation. However safety concerns and a lack of information about safe routes are often reasons commuters opt not to cycle. Hence, city councils try to find methods to improve safety or inform cyclist about safe routes that are available.

In this report, a practical application of the bi-objective optimization algorithm has been presented in the development of a web application that allows cyclists to find suitable routes from A to B. By using both the distance as well as a set of road information called attractiveness the cyclist is given multiple route choices. Each of them provides a compromise between distance and attractiveness and allows cyclists to make the best cycle decision from the choice of routes available.

The web application is easily accessible and has a user interface similar to Google Maps. Expansions of the application to include geographical data and a better choice of spatially different paths would allow the algorithm to show more relevant optimal routes. Further, possibilities to work with city councils could give access to better sets of data and the inclusion of application tasks that will gather useful information for councils will allow for improvements to the actual cycle networks.

Acknowledgements

The support of Dr. Andrea Raith and Assoc. Prof. Matthias Ehrgott from the University of Auckland Department of Engineering Science during the project is greatly appreciated. Their knowledge and guidance has helped to successfully complete the project.

References

Aultmann-Hall, L., F.L. Hall, and B.B. Baetz. 1997. "Analysis of Bicycle Commuter Routes Using Geographic Information Systems, Implications for Bicycle Planning." *Transportation Research Records.*

City, North Shore. 2009. North Shore City Cycling Stragetgy Plan 2009. `http://www.aucklandtransport.govt.nz/improving-transport/strategies/TransportStrategies/Documents/Original/nscc-cycling-strategy-plan-2009.pdf`.

Council, Auckland City. 2009. Traffic Flow. `http://www.aucklandcity.govt.nz/auckland/transport/flow/default.asp`.

Council, Auckland Regional. 2010. ALGGI Map Portal. `http://maps.auckland.govt.nz/Alggi/`.

Harkey, L.H., D.W. Reinfurt, and M. Knuiman. 1998. "Development of the Bicycle Compatibility Index." *Transportation Research Record 1636. North Carolina.*

Landis, B.W., V.R. Vattikuti, R.M. Ottenberg, T.A. Petritsch, M. Guttenplan, and L.B. Crider. 1997. "Intersection Level of Service for the Bicycle Through Movement." *Transportation Research Record.*

Raith, A., C. Van Houtte, J. Wang, and M. Ehrgott. 2009. "Applying Bi-objective Shortest Path Methods to Model Cycle Route-choice."

Raith, A, and M. Ehrgott. 2009. "A comparison of solution strategies for biobjective shortest path problems." *Comput. Oper. Res.* 36 (April): 1299–1331.

Zealand, Land Transport New. 2005. "Cycle network and route planning: 9 Evaluation Cycle Route Options."

Selecting a Portfolio of Cycling Projects

Uttara Nataraj
Department of Engineering Science
University of Auckland
New Zealand
unat002@aucklanduni.ac.nz

Abstract

At present, Auckland's cycling infrastructure is being developed not only to address problems associated with global warming, but also to promote healthier forms of transportation. Currently, the New Zealand Transport Agency (NZTA) employs a six stage decision making process when deciding which proposed cycling projects should be implemented. Here, the primary focus is to calculate the benefit to cost ratio for all proposed feasible projects and then to select those which perform well in this decision criterion within the constraint of a given cycling budget. Two weaknesses in this approach are the use of an unreliable demand forecasting model and the failure to capture interdependencies between proposed projects. Thus, a new methodology was developed to overcome these weaknesses and a focus area within the Auckland region was selected to demonstrate this improved process. A vital component in this methodology was to perform a cycling assignment on projects both individually and in conjunction with each other to determine the number of existing and new cyclists that are expected to use the new facilities. These forecasts were then used to calculate the benefit to cost ratios and finally, a linearised quadratic knapsack formulation was used to select the optimal portfolio of cycling projects.

Key Words: Benefit to cost ratio, Cycling assignment, Quadratic knapsack problem.

1 Introduction

The issue of global warming has assumed significant proportions in the world today. Great concern surrounds the large quantities of greenhouse gases such as carbon dioxide which are trapping infrared radiation in our atmosphere. Recent studies have shown that the carbon dioxide that is produced globally from the burning of fossil fuels such as petrol, diesel and ethanol is increasing by three percent every year (World Resources Institute, 2010). Considering that one of the main users of such fossil fuels are automobiles, sustainable alternative modes of transportation that do not rely on fossil fuels must be considered in order to prevent this statistic from increasing.

The New Zealand Transport Agency (NZTA) has recognised that active modes of transport such as walking and cycling should be promoted. Not only do these modes have atmospheric benefits, but they are also believed to alleviate congestion, help improve travel times for all road users and improve the reliability and resilience of the transport networks (NZ Transport Agency, 2009). Additionally, there are also clear health benefits which are associated with these active modes of transport.

In 2008, the NZTA set a target to double walking and cycling modes of transport to 30% of total trips by 2040 by building and maintaining quality facilities within the

Auckland urban and peri-urban areas (Hinton & Teh, 2008). This was to be achieved primarily by the addition of cycling facilities to state highways throughout Auckland as these highways are viewed to be vital in connecting and binding communities and facilitating economic development. As a result of this, potential cycling projects were analysed in the Southern, Western and Northern regions of Auckland. However, implementing the entire list of these projects is not possible given a cycling budget provided by the government.

Thus, the aim of this research project is to select a portfolio of potential cycling projects within the Auckland region that are beneficial to the cycling infrastructure, using an improved portfolio selection strategy that overcomes flaws in the current portfolio selection strategy.

2 Potential Cycling Projects

According to the Auckland Region Walking and Cycling Strategy (Hinton & Teh, 2008), in 2008 NZTA officials along with representatives from the Auckland Regional Transportation Authority (ARTA) developed a list of 79 proposed walking and cycling projects within the Auckland region. These projects were compiled from the individual project lists that were considered in the Auckland, North Shore, Waitakere, Manakau, Rodney, Papakura and Franklin regions.

3 Economic Evaluation Manual

The Economic Evaluation Manuals (Volumes 1 and 2) (EEM1 and EEM2), are documents created in 2009 by the NZTA which cover the economic efficiency evaluation of demand management and transport services activities for land transport (*EEM2*, 2009). For this research project the main section that was taken into consideration was Chapter 8 from EEM2 which includes the evaluation of walking and cycling facilities. Additionally, the procedure that was used to evaluate the new proposed projects by calculating a facilities benefit to cost ratio (BCR) was the Simplified Procedure 11 (SP11), which is specific to the evaluation of walking and cycling facilities. The BCR underpins the basis of the current portfolio selection strategy that the NZTA presently uses, and thus must be explored further.

4 Benefit to Cost Ratio Components

As mentioned in Section 3, the main component of the SP11 procedure is the BCR calculation of a new facility. The BCR value is used when determining the feasibility of a project and whether it should proceed to later stages in the evaluation process. The BCR for a new facility overall is defined as the following:

$$BCR(facility) = \frac{Benefit\ (facility)}{Cost\ (facility)}.$$

Formula 4.1: Benefit to Cost Ratio for a Facility.

In order for the BCR to be calculated as a ratio it is essential that we are able to express the benefits of a new facility in terms of monetary values. For this purpose, the Consumer Surplus Methodology is employed when determining the benefits with monetary value (*EEM2*, 2009). In this methodology, each of the types of benefits that a new facility will produce is monetised according to a consumers' willingness to pay for this benefit. The components of the benefits of a new facility that are thus considered for BCR calculations (as they can be monetised) are the travel time, vehicle operating, crash cost, pedestrian and cyclist, seal extension and passing lane, carbon dioxide and other tangible benefits.

In regards to the costs of a new facility, two main costs that are considered in the BCR calculations are the capital and maintenance costs. The capital costs are those which are required to bring the cycling facility to a commercially appropriate standard while maintenance costs are those which are incurred in the day to day running of the new facility. The costs are estimations and primarily based on the size of a new facility.

5 Demand Forecast

A common component in determining the travel time, the vehicle operating, pedestrian and cyclist and carbon dioxide savings is the demand forecast for a new facility. Since this is an important component of the benefit calculations, it is essential that we fully understand how these forecasts by the NZTA are currently estimated. The demand forecast is the number of cyclists that are forecasted to use the new cycling facility and consist of two components: existing cyclists and new cyclists.

Existing cyclists are frequent Auckland cyclists who are expected to use a facility if it is constructed. According to the SP11, the existing cyclists are estimated using observational data where the current numbers of cyclists are manually counted between the times of 7:00am-9:00am and 4:00pm to 6:00pm.

New cyclists are those who are assumed will start to cycle due to the construction of the new facility. These new cyclists are made up of new commuter cyclists (those who cycle as a mode of transportation to work) and new other cyclists (those who will cycle for other purposes, e.g. recreation). The numbers of new cyclists for a facility are estimated using the “Buffer Zone” method. The basis of this method is that buffer zones with a certain radius around a facility are analysed and from this a forecast is made for the number of new commuter and other cyclists based on population densities.

6 Current Portfolio Selection Strategy

So far, the fundamentals of the BCR calculations have been explored, where the various components of the benefits and costs have been defined. Following this, we can now examine the NZTA’s current portfolio selection strategy in order to identify any weaknesses so that improvements that can be made. An overview of the current six stage selection process is as follows.

Stage 1 (Identification): This stage involves listing all possible walking and cycling projects that adjoin or cross the state highway network. This list is compiled with the Auckland City Council’s support and has connectivity with the local network.

Stage 2 (Consultation): In this stage, there is a consultation with Auckland’s seven local authorities and ARTA in order to discuss specific projects that should be taken into consideration for each of the seven different areas. This is when the list of projects for the various regions are collated.

Stage 3 (Assessment): In this stage, all of the projects within the different areas are ranked according to defined assessment criteria. There are a total of six assessment criteria for which each of the projects is awarded a score between 1 and 3. These scores are then summed to give a total assessment score for a project.

Stage 4 (Ranking and Urgency): In this stage, the remaining projects are are tagged as urgent (U), under investigation (I) or pending (P) depending on their urgency.

Stage 5 (Prioritisation): In this stage, a shortlist of projects is determined based on the assessment criteria and urgency scores. Here projects which score above 9 that are also deemed ‘U’ are priority projects for investigation while projects which score above 12 that are also deemed ‘I’ are projects for investigation.

Stage 6 (Feasibility and Selection): In this stage, for each of the shortlisted projects, the benefits, costs and hence the BCR is determined using the SP11 procedure. This list of projects is then ranked according to the largest to smallest BCR values. A final portfolio of projects is selected for implementation by choosing the projects with the largest BCRs within the constraints of the cycling budget.

7 Weaknesses in Current Portfolio Selection Strategy

The analysis of the NZTA’s current portfolio selection strategy has highlighted some weaknesses that must be addressed in order to obtain a more thorough portfolio selection strategy. The main weaknesses have been identified in Stages 5 and 6 of the current strategy and are as follows.

The first major weakness identified was that the current methodology does not fully capture interdependencies between different projects. While the benefits, costs and the BCR are calculated for all individual projects in Stage 6, there is no real investigation into the benefit of placing two or more projects in conjunction with each other. While there is an attempt to consider these interdependencies in Stage 5 (i.e. ‘Urgent’ projects potentially are those which could provide missing regional cycle network links), there is currently no established mathematical method of modelling this.

The second main weakness that was found in the current methodology was the approach used to forecast the demand. Both the observational data used to forecast the number of existing cyclists and the ‘Buffer Zone’ method used to forecast the number of new cyclists are not particularly accurate and do not truely represent the number of cyclists expected to use a facility. As mentioned in Section 5, the number of cyclists that are estimated to use a new facility is a large contributor to the benefits for a facility and thus must be estimated as accurately as possible.

8 New Portfolio Selection Strategy

Given that the main weaknesses of the current methodology have been established, a new portfolio selection strategy which addresses these issues must be developed. Since the identification and consultation stages of the current strategy involve primarily determining the list of proposed projects, there is not much change that can be suggested there. Similarly, the assessment, ranking/urgency and prioritisation stages involve primarily short listing the large list into an ‘urgent’ list, and thus not much alteration can be imposed there either. Therefore, the main stage which will be explored is the feasibility and selection stage, where from the list of ‘urgent’ projects, a final portfolio can be selected. Thus a new portfolio selection strategy has been developed (see Figure 8.1). A description of these stages is as follows below.

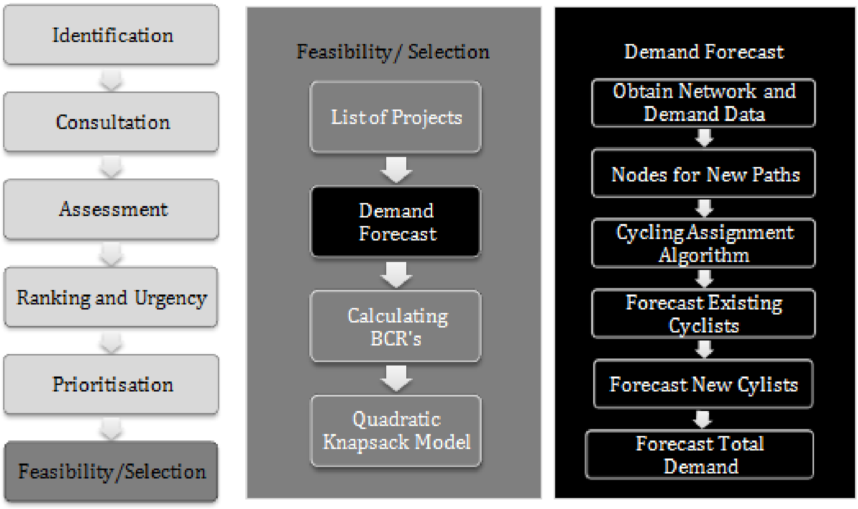


Figure 8.1: New Portfolio Selection Strategy with Demand Forecast Methodology.

8.1 Stage 1: Obtaining a List of Projects

The first stage of the new portfolio selection strategy involves obtaining a list of projects from which the optimal portfolio of cycling projects will be chosen. As mentioned previously, Stages 1-5 of NZTA's current portfolio strategy involves primarily obtaining this short list from the entire list of projects, and thus in reality, this stage will already be near completion at the end of Stage 5 of the original methodology.

8.2 Stage 2: Demand Forecast

The second step in the new methodology is to obtain the demand forecasts individual projects and for different combinations of projects. The reason for considering combinations of adjacent projects is because this is where the interdependency flaw is overcome. By forecasting demand and calculating the BCRs for projects alone and in conjunction with each other we can identify the benefit or disadvantage of projects implemented together.

Obtain Network and Demand Data: The first step in forcasting the demand is to aquire the data sets that are required in the cycling assignment. The first data input required is a network representation of the area that the projects are within. This network extracted contains all roads on which bikes are allowed. The second major data input which is required for the cycling assignment algorithm is the cycling demand matrix. This demand data is presented in a matrix and represents the information on how many cyclists currently travel from various origins to various destinations in the Auckland network. The demand matrix is required as these are the cyclists that will be used to determine the number of existing cyclists on each of the new proposed paths.

Nodes for New Paths: The second step is to create the new proposed projects using nodes from the current network which can be inserted into the Auckland network when required. The reason for this is that the new cycle paths obviously do not exist in this network and hence they must be added in.

Cycling Assignment Algorithm to Forecast New Cyclists: The third step was to develop an algorithm that performs the cycling assignment to estimate the number of existing cyclists. The code that was implemented for this algorithm was written in C and was compiled and run using the operating system Linux. The initial algorithm for the cycling assignment is as follows (based on orginal code written by Dr. Andrea Raith).

Cycling Assignment Algorithm:

1. *Read in the demand matrix from a data file which contains information about the total number of origin-destination (OD) nodes and the demands between all OD pairs.*
2. *Read in the Auckland network from a data file.*
3. *Convert the Auckland network to the forward star representation.*
4. *Start the clock.*
5. *Insert the new arcs which represent the new projects from a data file into the forward star representation.*
6. *Set a modal share (percentage of cyclists) and a cycling tolerance (number of cyclists for which a demand less than this is ignored).*
7. *For each OD pair in the demand matrix, do the following:*
    - 7.1. *If the source node and destination node are not the same, do the following:*
        - 7.1.1. *Set the source node, destination node and the demand for this pair.*
        - 7.1.2. *Determine the cycling demand using the cycling multiplier.*
        - 7.1.3. *If the cycling demand is greater than the cycling tolerance, do the following:*
            - 7.1.3.1. *Use the Bi-objective Label Setting Algorithm to find the set of efficient paths (those which are better in at least one component in comparison to the inefficient solutions, and are not equal) from the source node to the destination node.*
            - 7.1.3.2. *Determine the flow on each efficient path by distributing demand evenly amongst all efficient paths.*
            - 7.1.3.3. *Backtrack along all arcs which make up each efficient path and allocate the flow as determined above.*
8. *Print out the total flow on the new arcs.*
9. *End the clock.*

Forecast New Cyclists and Total Demand: Performing this cycling assignment algorithm on the required network will yield the number of existing cyclists expected to use the new facility. In order to obtain the number of new cyclists expected to use the facility, the existing demand figures and appropriate multipliers are used. The sum of the existing demand and the new demand yields the total demand for a facility.

8.3 Stage 3: Calculating Benefits and Costs

The third stage in the overall portfolio selection strategy is to obtain the benefits and costs for each of the paths individually and combined, which will in turn be used in the optimisation model. Both the benefits and the costs are calculated using an updated EXCEL workbook, which is based on an original workbook which was developed by the NZTA and contains the procedures for calculating the benefits and costs for a proposed new facility as shown in SP11.

8.4 Stage 4: Quadratic Knapsack Model

The final stage in the new portfolio selection strategy is to select the optimal portfolio of cycling projects using the Quadratic Knapsack Problem linearised (QKPL) as shown below. Here, the objective is to maximise the sum of the individual benefits as well as the additional benefits from two projects in conjunction with each other. The only constraint is that the sum of the costs of the implemented projects are within a defined budget.

Decision Variables:
x_i = *Implementing proposed project i.*
x_j = *Implementing proposed project j.*
y_{ij} =*Implementing proposed projects i and j in conjunction.*
Parameters:
b_j = *Benefit of implementing proposed project j.*
bij = Additional benefit of implementing proposed projects i and j in conjunction.
w_j = *Total cost of implementing proposed project j.*
c = Total cycling budget.

$$(QKPL): maximise \sum_{j=1}^{n} b_j x_j + \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij} y_{ij}$$

$$subject\ to \sum_{j=1}^{n} w_j x_j \le c,$$

$$y_{ij} \le x_i$$

$$y_{ij} \le x_j$$

$$x_{j,}\, x_j\, , y_{ij} \in \{0,1\},\ \ i,j = 1 \dots n.$$

Formula 8.1: Linearised Quadratic Knapsack Formulation.

9 Selected Focus Area

Given that the new portfolio selection strategy has now been described, the next step is to demonstrate that this methodology performs well in practice. In order to show this, a focus area within the Auckland region was selected. This section performs the four step methodology as described in Section 8 on the focus area in order to obtain the optimal cycling portfolio.

9.1 Stage 1: Obtaining a List of Projects

The focus area that was selected to demonstrate the new selection strategy is in the Auckland City region, and is the corridor along the Southern motorway which runs from Princes Street in Otahuhu to Carlton Gore Road in Newmarket.

The next step was to split this focus area, keeping in mind already proposed projects. Thus, the area was divided according to three projects namely Auckland City Proposed Cycling Project 8 (A08), Auckland City Proposed Cycling Project 7 (A07) and a New Project (NP). The NP was added in order to add connectivity to the Central Business District (CBD), as this is a major destination for many commuter cyclists. Since not much analysis can be done by simply considering three projects, A08, A07 and NP have been split into a total of seven paths that will be analysed when looking at interdependencies

9.2 Stage 2: Demand Forecast

Given that the focus area and the different paths have been selected to demonstrate the new methodology, the next step is to forecast the number of existing cyclists and the number of new cyclists which are expected to use the paths individually and in conjunction with adjacent paths. As mentioned previously, the number of existing cyclists are forecast using the cycling assignment algorithm. The Auckland network data set that was used was extracted from OpenStreetMap which is an online representation of geographical data and maps. The demand matrix data set used contains origins and destinationsa and values from the Auckland Transport Authority Model 3 (ART3) model, which was developed by ARTA. For the number of new cyclists, the number of new commuter cyclists are taken to be 19% of the existing cyclists while the number of new other cyclists are taken to be 15% of existing cyclists.

The final demand forecast figures which consist of the sum of the existing and new cyclists are as shown in Table 9.1. Here we see that the first run consists of inserting all seven paths into the network, runs 2 to 8 consist of inserting each of the paths individually and runs 9 to 14 consist of inserting combinations of two adjacent paths.

	Run													
New Arc	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	**66**	**59**	0	0	0	0	0	0	**65**	0	0	0	0	0
2	**163**	0	**124**	0	0	0	0	0	**128**	**136**	0	0	0	0
3	**334**	0	0	**138**	0	0	0	0	0	**157**	**273**	0	0	0
4	**294**	0	0	0	**157**	0	0	0	0	0	**218**	**209**	0	0
5	**435**	0	0	0	0	**185**	0	0	0	0	0	**312**	**209**	0
6	**702**	0	0	0	0	0	**355**	0	0	0	0	0	**448**	**535**
7	**881**	0	0	0	0	0	0	**767**	0	0	0	0	0	**833**

Table 9.1: Final Demand Forecasts for Individual ad Combinations of Paths.

From these results two observations can be made. Firstly, the largest flows are produced when the entire corridor is considered, the second largest flows are produced when two paths are considered in conjunction with each other and the smallest flows are produced when paths are considered alone. This clearly shows that examining combinations of paths (and thus projects) is a vital component that must be considered when forecasting demands. Secondly, as the paths get closer to the city, the demands also progressively increase. This is also to be expected as the CBD is a major destination for most cyclists.

9.3 Stage 3: Calculating Benefits and Costs

Now that all the demand forecast figures have been obtained for the focus area, the next step is to obtain the benefits (these include the individual benefits and the additional combined benefits) as well as the costs of the individual paths using the updated SP11 EXCEL workbook. The values for the benefits of individual projects, the additional benefits of projects in conjuction with each other are as follows:

Comb. Number	**Arcs**							**Existing Demand**	**New Demand**	**Benefits**				**Total Benefit**
	1	**2**	**3**	**4**	**5**	**6**	**7**			**TTS**	**VOS**	**PCS**	**CDS**	
2								44	15	33099	1701	38444	68	73312
3								92	31	57398	2951	81141	118	141608
4								103	35	31622	1626	90690	65	124003
5								117	40	75627	3889	102851	156	182523
6								138	47	63595	3269	121259	131	188254
7								265	90	82224	4227	233179	169	319799
8								572	195	340173	17489	503300	700	861662

Table 9.2: Final Benefits for Individual Paths in Focus Area.

Comb. Number	Arcs							Add. Existing Demand	Add. New Demand	Additional Benefits				Additional Total Benefit
	1	2	3	4	5	6	7			TTS	VOS	PCS	CDS	
9								8	3	11260	580	7180	23	19043
10								23	8	21638	1114	20489	45	43286
11								146	50	139575	7175	128760	287	275797
12								135	46	149137	7667	118385	307	275496
13								88	30	67788	3486	77289	139	148702
14								183	62	165397	8504	160812	340	335053

Table 9.3: Final Benefits for Combinations of Paths in Focus Area.

In regards to the calculations for the cost of implementing the individual projects, firstly it has been assumed that the capital costs considered are identical for all paths. Secondly, it has been assumed that the maintenance costs considered are linearly related to the total length of the path. The costs of the individual projects are as follows:

Path	Length	Capital Cost	Maintenance Cost	Total Cost
1	2405	1000000	3608	1003608
2	1976	1000000	2964	1002964
3	974	1000000	1461	1001461
4	2054	1000000	3081	1003081
5	1465	1000000	2198	1002198
6	985	1000000	1478	1001478
7	1888	1000000	2832	1002832

Table 9.4: Total Costs for each Path in Focus Area.

9.4 Stage 4: Quadratic Knapsack Model

Based on the results obtained from the quadratic knapsack model linearised which was implemented in AMPL using the benefits and costs obtained above, the recommended optimal portfolio of cycling paths for the focus area consists of path 4,5,6 and 7 (as highlighted in Figure 9.1) and has an optimal objective of $2311489.40. This portfolio clearly supports the theory of considering interdependencies between projects as all the paths selected form a continuous corridor. Additionally, these paths are closest to the CBD and had the largest flows, which support the theory that the demand forecast is a vital component in the selection of the cycling portfolio.

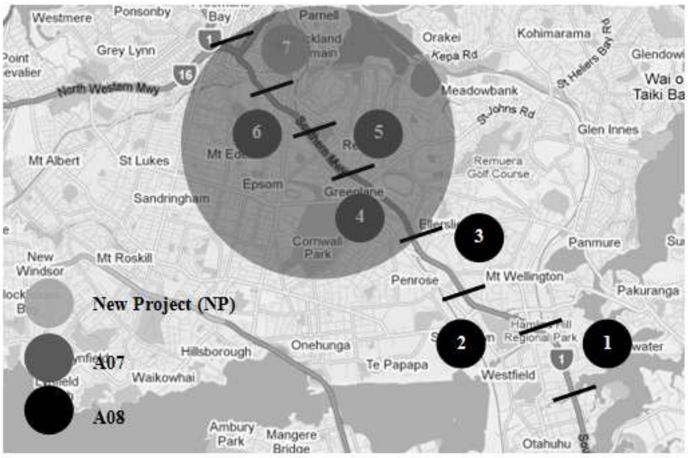


Figure 9.1: Optimal Portfolio of Cycling Projects for Focus Area (Google Maps, 2010).

10 Future Work

This research project has involved primarily altering the NZTA's current portfolio strategy in order to develop a new strategy which has a more accurate demand forecast and takes into consideration interdependencies between projects. While this has been proved to be an effective strategy, there are five major opportunities for future work:

1. Testing projects which have a reduced additional joint benefit.
2. Exploring the concept of induced demand. Induced demand is where due to the introduction of new facility, there will be additional demand for that facility.
3. Consider larger combinations of adjacent projects.
4. Applying this new methodology to a larger project set.
5. Explore methods used to split the demand amongst the efficient paths in the cycling assignment algorithm other than evenly.

11 Conclusions

In this research project we have:

- Fully understood the NZTA's current portfolio selection strategy, its various components and the list of potential cycling projects.
- Critically identified flaws in the current portfolio selection strategy and thus derived a more thorough portfolio selection strategy.
- Focused attention on a particular area within Auckland from the given list of potential cycling projects, applied the improved portfolio selection strategy and obtained an optimal portfolio of cycling projects in order to verify that the new strategy is appropriate.

Acknowledgments

I would like to thank my supervisors, Dr. Andrea Raith, Associate Professor Matthias Ehrgott and Garry Miller for their continuous support and invaluable expertise throughout the duration of this research project. I would like to thank Mr. Keith Pauw, Mr. Mieszko Iwaskow and Dr. Ranjan Pant from the NZTA for their very useful insight into the cycling world. I would like to thank ARTA and Mr. John Davies for the ART3 network model that used in the development of this research project. I would like to thank Dr. Michael O'Sullivan for providing me with help when required with AMPL. I would like to thank Dr. Judith Wang from the Business School for her general help and guidance throughout the duration of the project. I would like to thank Mr. Percy Barboza for his technical support during the initial stages of the project.

References

EEM1. (2010). NZ Transport Agency.

EEM2. (2009). NZ Transport Agency.

Google Maps. (2010). *Google*. Retrieved 13 September, 2010, from http://maps.google.co.nz/

Hinton, M., & Teh, C. (2008). *Auckland Region Walking and Cycling Strategy*. Auckland: Maunsell.

NZ Transport Agency. (2009). *Planning for Walking and Cycling*. Retrieved 3 March, 2010, from http://www.nzta.govt.nz/planning/process/walking-cycling.html

World Resources Institute. (2010). Retrieved 2 August, 2010, from http://www.wri.org

Proceedings of the 45th Annual Conference of the ORSNZ, November 2010

A Multi-commodity Flow Formulation for the Integrated Aircraft Routing, Crew Pairing, and Tail Assignment Problem

S. Ruther
School of Mathematical and Physical Science
University of Newcastle
Newcastle
Australia

Abstract

The integrated aircraft routing, crew pairing, and tail assignment problem consists of simultaneously finding a minimum cost set of aircraft routes and crew pairings such that each flight is covered by one aircraft and one crew.

A common problem when integrating airline planning stages is the long planning horizon of the crew pairing problem. We propose an approach in which crews initially are only told when they work. This enables us to generate an overall schedule much closer to the start of the planning horizon. Therefore, along with a short planning horizon, much more detailed and accurate overall schedules can be generated.

Due to the tail assignment aspect of the problem maintenance requirements have to be satisfied for each aircraft. Robustness of solutions is increased by using penalties on short connections.

We propose a mixed integer multi commodity flow formulation and report results for small instances.

Key words: Aircraft Routing, Crew Pairing, Tail Assignment, Muli-commodity Flow Formulation

1 Introduction

Airlines have been using operations research methods to tackle their complex planning and operational problems for decades. Due to the size of their operations not all planning decisions can be made simultaneously. In practice a sequential approach is used in the planning stage (Klabjan 2005). After a *flight schedule* is developed aircraft types are assigned to each flight leg in the *fleet assignment* problem. The goal is to match the capacity of the aircraft and the estimated number of passengers closely, thereby maximizing overall profit. Only few resources such as the number of available aircraft in each fleet are considered.

The following *aircraft routing* problem selects a minimal cost set of routes such that all flight legs are covered. Generic aircraft routes are generated by selecting a number of flights that have to be flown in sequence. While general maintenance requirements like days between maintenance checks have to be respected, the actual state of an aircraft, for example time spent flying so far, is ignored.

In the *crew pairing* problem generic pairings are generated such that the crew costs are minimized. A pairing is generated by selecting a number of flight legs that are serviced in succession by a crew while obeying several restrictions such as days off, limits on time spend working or layover requirements.

The pairings are put together to monthly rosters in the *crew rostering* problem. They are then assigned to actual crews based on personal preferences and needs. The crew rostering problem is solved approximately one month before the day of operations and only minor changes in the crew schedule are made after this step.

A few weeks before the day of operations the *tail assignment* problem is solved. Individual aircraft are assigned to routes generated in the aircraft routing problem. The planner needs to ensure that routes are feasible with respect to maintenance, taking into account the actual aircraft location and resource consumption at the beginning of the time horizon.

Because of the sequential nature it is very unlikely that the overall optimal or even a good solution can be obtained. Decisions in early planning stages like flight schedule generation or fleet assignment restrict the choices in later stages. For example in the fleet assignment problem the main issue is that no detailed information about aircraft and crews are available so the fleet assignment problem usually does not take into consideration that aircraft are temporarily unavailable due to maintenance checks. This may result in infeasibility in the aircraft routing problem, in which case flight legs have to be assigned to different aircraft types manually, increasing cost significantly. Also interdependences between stages, for example the issue of short connections in the aircraft routing and crew pairing problem, will result in higher cost when not considered in an integrated model.

The outline of this paper is as follows. In section 2 we give the motivation of our approach. Section 3 reviews integrated models that have been published. In section 4 we present a multi commodity flow formulation. We finish the paper with an outline of future work in section 5.

2 Motivation

A major disadvantage of the sequential approach is that it includes solving the aircraft routing and the crew pairing problem usually weeks or even months before the day of operations. Any decision making process is limited by the accuracy of the information available at the time of the decision. In the case of aircraft routing, information about the status and location of aircraft become significantly more accurate as the day of operations approaches. For this reason, it would be beneficial to delay any decisions as long as possible, while ensuring that subsequent planning stages can be carried out.

However, crews want to know their schedules some time in advance. Therefore the crew rostering problem is a major limiting factor to prolonging any decisions. It is usually solved approximately one month before the day of operations. This has two implications: first, the decisions of the aircraft routing problem and those of the crew pairing problem have to be made at least one month before the day of operations. Second, the minimum duration of a schedule is one month.

We propose an approach that can overcome this issue. Weeks before the day of operations an airline tells its crews on which days they have to work but not on which flights, i.e. only tells them the beginning and end of their pairings. Crew members then are able to manage their private lives while the airline can prolong aircraft routing and crew pairing decisions. Because crews are told when they have to work weeks before the final schedule is generated, it is the airline's responsibility to later generate pairings with durations close to the scheduled durations. If the airline fails to do so, the crew will have to get payed, despite not actually working on those days.

We develop the idea further by noting that there may be multiple crews stationed at the same base having the same start and end date. These crews can be considered identical in the planning process and are therefore represented by a *crew block*. Within a crew block the airline is free to make any assignments closer to the day of operations as long as the block start and end times are respected.

This approach impacts both issues raised above: the aircraft routing and the crew pairing problems do not need to be solved at least one month before the day of operations and the length of the schedule does not need to be one month. Instead, the length of the schedule now depends on negotiations between the airline and its employees. Crews still prefer knowing in advance on which days and which flights they have to work but may be more flexible if a higher financial reward is payed. We believe that the increased cost associated with having more flexible crews is more than offset by higher revenue resulting from more accurate operational schedules.

The duration of the schedule to be generated should not exceed 7 days as deviations from a longer schedule are to be expected. However, another 5 days worth of flying is included so that it can be guaranteed that crews are able to finish their pairings and that aircraft routes are feasible with respect to maintenance. Most published models solve a daily problem and then simply repeat the schedule for every day, neglecting the fact that the demand and schedule can be substantially different on weekends. Instead, we consider the actual flights scheduled in the next 12 days. The model is to be solved 4 days before the day of operations. At that time it should be possible to project location and status of crews and aircraft accurately. On the other hand sufficient time will be available to evaluate the solution and possibly resolve the problem.

Solving an integrated model this close to the day of operations facilitates considering individual aircraft, i.e. tail numbers. This addresses another major disadvantage of models that are solved a long time before the day of operations. In recent integrated models (see section 3) the aircraft routing problem ensures that maintenance feasible routes are generated. However, these routes are generic and do not consider the current state of an aircraft on the day of operations. Often the schedule has to be changed because aircraft are at a different location than expected at the time of the planning process or they require maintenance earlier than anticipated. Four days before the day of operations information will be available that allows accurate prediction of state and location of an aircraft. Therefore, instead of generating generic routes we propose to generate routes for each individual tail number. These routes can represent specific maintenance requirements, eliminating the need to assign aircraft and possibly having to change the schedule because a feasible assignment is not possible.

In summary, we integrate aircraft routing, tail assignment and crew pairing generation. The model is to be solved daily with a rolling horizon, enabling us to consider current developments in resource consumption or deviations from earlier schedules.

3 Literature review

In scholarly publications a number of integrated models have been developed in the past. (Desaulniers et al.) and (Barnhart et al. 1998) semi-integrate fleet assignment and aircraft routing. In both cases the models are solved using the branch-and-price method. In the former no maintenance is considered, therefore feasibility can not be guaranteed. Flights are allowed to depart within a certain time window, resulting in lower cost compared to fixed departure times. The latter considers strings of flights with maintenance opportunities attached to the end. The cost of assigning an aircraft type to a flight leg is considered.

(Klabjan et al.) semi-integrate aircraft routing and crew pairing. They reverse the order by first solving the crew pairing problem instead of the aircraft routing problem. Plane-count constraints are introduced to the crew pairing problem, ensuring feasibility of the aircraft routing problem under the assumption that maintenance is performed over night when all aircraft are on the ground. In addition, flight departure times are allowed to vary within time windows.

A model that fully integrates aircraft routing and crew pairing is developed by (Cordeau et al. 2001). The aforementioned issue of short connections is considered here for the first time. The resulting large number of constraints are handled by Benders decomposition, with aircraft routing as the master problem. The solution process iterates between the master problem and a subproblem that solves the crew pairing problem. Both the Benders master and subproblem are solved by Column Generation. No cost are associated with aircraft routes, reducing the aircraft routing problem to a feasibility problem. This neglects the fact that with more frequent maintenance aircraft are unavailable more often and higher maintenance costs are incurred.

(Cohn and Barnhart 2003) also present a model for the integrated aircraft routing and crew pairing problem. As in (Cordeau et al. 2001) aircraft routing cost are neglected and the aircraft routing reduces to a feasibility problem. However, they develop this idea further by realizing that routes that differ only by the order of the legs flown can be represented by a single column in the so called extended crew pairing problem. This reduces the number of variables in the model significantly but comes at the cost of having to solve multiple aircraft routing problems that generate these columns. However, the approach is inapt if maintenance cost need to be considered because in this case the routes differ not only by the order of legs flown. The extended crew pairing problem is solved by a branch-and-price algorithm. The crew pairing and aircraft routing solutions are generated in pricing problems.

The model of (Cordeau et al. 2001) was developed further by (Mercier, Cordeau, and Soumis 2005) who increased solution robustness by penalizing connections that are likely to cause delays if they are not performed by the same aircraft and crew. The idea is a generalization of the concept used for short connections. The authors also show that solving the aircraft routing problem as the Benders master problem and the crew pairing problem as the subproblem is beneficial because fewer Benders cuts need to be generated. This can be attributed to the fact that with an aircraft routing subproblem mostly feasibility and only little optimality information is transferred to the master problem.

The latter model is further enhanced by (Mercier and Soumis) by introducing time windows to the formulation. A solution algorithm combining Benders decomposition, column generation and a dynamic constraint generation procedure is developed and proved to be very efficient for their problem.

(Sandhu and Klabjan 2007) integrate fleet assignment and crew pairing while considering some aircraft routing aspects. The model ensures that a pairing is assigned to a single fleet type only, thus permitting different sizes and qualifications of crews. Only plane-count constraints are included in the formulation which means that maintenance feasibility can not be guaranteed. The model is solved by Benders decomposition as well as a combination of Lagrangian relaxation and branch-and-price. The first method finds good solutions quickly but is outperformed by the latter if more solution time is available.

(Weide 2009) uses an iterative procedure in which the aircraft routing and the crew pairing problem are solved alternately. Short and restricted connection rules are considered but because of the iterative nature not all of these connections are included. A daily flight schedule is generated and maintenance is assumed to be carried out over night when aircraft are on the ground.

(Papadakos 2009) presents a model that is based on (Cordeau et al. 2001) but fully integrates fleet assignment, aircraft routing, and crew pairing. Similar to (Sandhu and Klabjan 2007) crew pairings are fleet dependent. The model is solved using a combination of Benders decomposition and accelerated column generation.

The literature suggests column generation or Bender's decomposition or even a combination of both as suitable algorithms to solve integrated airline problems. However due to the advances in general mixed integer programming in the last decade we decided to explore if such standard methods have become suitable as well.

4 Mathematical model

The model is based on a muli-commodity flow formulation where each crew block and each aircraft are represented by one commodity. The nodes in the network represent flight legs, while the arcs represent connections between flight legs. Additionally, one source and one sink node is added for each commodity.

Sets

B	The set of all crew blocks
R	The set of all individual aircraft
Ω	The set of all types of maintenance checks
Δ	The set of all resources under consideration
Δ_ω	The set of resource which apply to the check type $\omega \in \Omega$
N	The set of all nodes
N_{Legs}	The set of nodes which represent actual flight legs
N^+_{AC}	The set of aircraft source nodes r^+
N^-_{AC}	The set of aircraft sink nodes r^-

C_{Crew}	The set of all crew connections
$C_{\text{Crew}}^{\text{Duty}}$	The set of duty connections for crews
$C_{\text{Crew}}^{\text{Restr}}$	The set of restricted connections for crews
$C_{\text{Crew}}^{\text{Lay}}$	The set of layover connections for crews
C_{AC}	The set of aircraft connections

Parameters

$\text{Block}(i)$	The block hours for leg $i \in N_{\text{Legs}}$
$\text{Wait}(i,j)$	The total crew waiting time at an airport associated with connection $(i,j) \in C_{\text{Crew}}$ including briefing time, waiting and transit time, but not including debriefing time
MaxDuty	The maximum allowable duty hours in a duty
$\text{Penalty}(i,j)$	A penalty for assigning a crew block to a restricted connection $(i,j) \in C_{\text{Crew}}^{\text{Restr}}$ while not assigning an aircraft to the same connection
$\text{MaintCost}_{ij}^{\omega}$	Cost of carrying out a maintenance check of type $\omega \in \Omega$ on connection $(i,j) \in C_{\text{Craft}}$
$\text{LayCost}(i,j)$	The cost of a layover on connection $(i,j) \in C_{\text{Crew}}^{\text{Lay}}$, which includes transport, hotel, and meal allowance cost.
$\text{Credit}(i)$	The applicable credit for leg $i \in N_{\text{Legs}}$
MinCredit	The minimum amount of credit for a duty
CreditCost	The cost of one unit of credit for one crew
DeBriefTime	The duration of a debriefing at the end of a duty
$\text{Used}_{\text{turn}}^{\delta}(i,j)$	The amount of resource $\delta \in \Delta$ used on connection $(i,j) \in C_{\text{Craft}}$
$\text{Used}_{\text{leg}}^{\delta}(i)$	The amount of resource $\delta \in \Delta$ used on leg $i \in N_{\text{Legs}}$
$\text{StartVal}^{\omega\delta}(r)$	The amount of resource $\delta \in \Delta_{\omega}$ used since the last maintenance of type $\omega \in \Omega$ for aircraft $r \in R$ at the start of the planning period
$\text{MaxMaint}^{\omega\delta}$	The maximum amount of resource $\delta \in \Delta_{\omega}$ which can be accumulated before carrying out a maintenance check of type $\omega \in \Omega$

Decision variables

$$y_{ijb} = \begin{cases} 1 & \text{if crew block } b \in B \text{ uses connection } (i,j) \in C_{\text{Crew}} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ijr} = \begin{cases} 1 & \text{if aircraft } r \in R \text{ uses connection } (i,j) \in C_{\text{Craft}} \\ 0 & \text{otherwise} \end{cases}$$

$$w_{ij} = \begin{cases} 1 & \text{if both plane and crew use connection } (i,j) \in C_{\text{Crew}}^{\text{Restr}} \cap C_{Craft} \\ 0 & \text{otherwise} \end{cases}$$

$$z_{ij}^{\omega} = \begin{cases} 1 & \text{if maintenance } \omega \in \Omega \text{ is scheduled on connection } (i,j) \in C_{\text{AC}} \\ 0 & \text{otherwise} \end{cases}$$

d_i The number of duty hours accumulated within a duty by a crew block at the end of leg $i \in N_{\text{Legs}}$

a_i The number of credit hours accumulated within a duty by a crew block at the end of leg $i \in N_{\text{Legs}}$

$\overline{a_{ij}}$ The number of credit hours accumulated by the end of a duty for a crew block that finishes its duty at the end of leg i and then connects to j, $\forall(i,j) \in C^{\text{Lay}}_{\text{Crew}}$ or $j \in N^{-}_{\text{AC}}$

$m_i^{\omega\delta}$ The amount of resource $\delta \in \Delta$ accumulated since last maintenance of type $\omega \in \Omega$ at the end of leg $i \in N_{\text{Legs}} \cup N^{+}_{\text{AC}}$

4.1 Objective function

The objective is to minimize the cost associated with a flight schedule and to increase the robustness of the solution:

$$\begin{aligned}
\text{Minimize} \quad & \sum_{\substack{\omega \in \Omega, \\ (i,j) \in C_{\text{AC}}}} \text{MaintCost}^{\omega}_{ij} * z^{\omega}_{ij} \\
& + \sum_{(i,j) \in C^{\text{Lay}}_{\text{Crew}}} \text{LayCost}(ij) * \sum_{b \in B} y_{ijb} \\
& + \sum_{(i,j) \in C_{\text{Crew}}} \text{CreditCost} * \overline{a}_{ij} \\
& + \sum_{(i,j) \in C^{\text{Restr}}_{\text{Crew}}} \text{Penalty}(i,j) * w_{ij}
\end{aligned} \tag{1}$$

The first term represents the maintenance cost if maintenance is carried out on a connection. The second term counts the layover cost, while the third captures the cost incurred by scheduling a certain duty. The last term is incurred every time an aircraft and a crew do not stay together on a restricted connection. All of these costs and penalties are only applied to flight legs and connections that occur in the first seven days of the planning horizon.

4.2 Constraints

We require that every flight has to be covered by exactly one aircraft and one crew block. This is achieved by using standard set partitioning constraints. Additionally we need flow conversation constraints and constraints that ensure that each commodity leaves its source. Another set of constrains deals with the issue of short connections, see (Cordeau et al. 2001). They require that if a crew uses a short connection an aircraft has to use the same connection as well. In other words, the same crew and aircraft are assigned to consecutive flights, i.e. no aircraft swap occurs.

To increase robustness a penalty term is incurred every time a crew or an aircraft uses a restricted connection without the other one doing so as well. This increases robustness as fewer aircraft changes occur. Restricted connections have a duration of 45 to 90 minutes.

$$\sum_{r \in R} x_{ijr} \leq w_{ij}, \qquad \forall\ (i,j) \in C^{\text{Restr}}_{\text{Crew}} \cap C_{\text{AC}} \tag{2}$$

$$\sum_{b \in B} y_{ijb} \leq w_{ij}, \qquad \forall\ (i,j) \in C^{\text{Restr}}_{\text{Crew}} \cap C_{\text{AC}} \tag{3}$$

4.2.1 Constraints relating to the length of a crew pairing

Regulations in Australia require that a crew only works up to 8 *block hours* and up to 11 *duty hours* per duty. Block hours are defined as time spent flying plus taxi time, while duty hours are the block hours plus the time spent at an airport between flights including briefing time. Duty hour restrictions are modeled using constraints:

$$d_i \leq \text{MaxDuty} - \text{DeBriefTime}, \qquad \forall i \in N_{\text{Legs}} \tag{4}$$

$$d_i \geq \text{Block}(i) + \sum_{\substack{b \in B,\, j \in N| \\ (j,i) \in C_{\text{Crew}}}} \text{Wait}(j,i) * y_{jib}, \qquad \forall i \in N_{\text{Legs}} \tag{5}$$

$$d_j - d_i \geq \text{Block}(j) + \text{Wait}(i,j) - M(1 - \sum_{b \in B} y_{ijb}), \qquad \forall (i,j) \in C^{\text{Duty}}_{\text{Crew}} \tag{6}$$

The first constraint limits the number of duty hours, while the second ensures that the book-keeping variable starts with the correct start value. The last constraint increments the variable along an arc if a crew uses the corresponding connection. Valid big M values are $M = \text{MaxDuty} - \text{DeBriefTime}$. Block hour requirements are modeled in a similar way, however they don't include a Wait term in the constraints.

Other regulations limit the number of block hours to 30 in a 7 day period for every crew. Because we are not considering detailed crew scheduling, we approximate this rule by applying the limit to every pairing, which have a maximal duration of 5 days. This rule is implemented in a similar fashion as the duty hour restrictions, constraints (4) - (6). The only difference is that the total block hours value increments along all crew connections including layovers and that no Wait term exists.

4.2.2 Constraints relating to crew cost

Applicable credit is a concept used to describe the amount of pay that a crew member receives. A *credit* value is associeated with every flight leg depending on its duration. The actual crew payments are then based on a fixed salary, with overtime paid if the number of credit hours exceeds a given value. For the time frame considered here, an approximation is used where applicable credit is charged at a fixed rate per hour per crew. Similar constraints as above are required:

$$a_j - a_i \geq \text{Credit}(j) - M(1 - \sum_{b \in B} y_{ijb}), \qquad \forall (i,j) \in C^{\text{Duty}}_{\text{Crew}} \tag{7}$$

$$a_i \geq \text{Credit}(i), \qquad \forall i \in N_{\text{Legs}} \tag{8}$$

The credit values at the end of each duty are required for costing purposes. Constraint (9) assigns the accumulated credit to variables representing the end of duties.

$$\overline{a_{ij}} \geq a_i - M(1 - \sum_{b \in B} y_{ijb}), \qquad \forall (i,j) \in C^{\text{Lay}}_{\text{Crew}} \text{ or } j \in N^{-}_{\text{AC}} \tag{9}$$

If an airline were to schedule too many short duties or pairings, it could become infeasible to schedule all flights including leave, training and days off. One method currently employed by airlines to ensure a sensible set of trips is to apply a minimum amount of applicable credit to each duty. Each duty will incur cost equal to this value or its true amount of credit, whichever is greater, thus penalizing the creation of too many duties. The following constraint is imposed on every arc that represents the end of a duty.

$$\overline{a_{ij}} \geq \text{MinCredit} * \sum_{b \in B} y_{ijb}, \qquad \forall (i,j) \in C^{\text{Lay}}_{\text{Crew}} \text{ or } j \in N^{-}_{\text{AC}} \tag{10}$$

4.2.3 Constraints relating to maintenance decisions

Maintenance requirements are modeled in a similar way as well. The book keeping variables are initialized with the correct values for each aircraft at the beginning of the time horizon

$$m^{\omega\delta}_{r^+} = \text{StartVal}^{\omega\delta}(r), \qquad \forall r \in R,\ \omega \in \Omega,\ \delta \in \Delta_\omega \tag{11}$$

The maintenance variables have upper limits.

$$m^{\omega\delta}_{i} \leq \text{MaxMaint}^{\omega\delta}, \qquad \forall i \in N_{\text{Legs}},\ \omega \in \Omega,\ \delta \in \Delta_\omega \tag{12}$$

Variables are incremented along any arc that an aircraft travels. However, the variables are not incremented if maintenance occurs at the same time, i.e. $z_{ij} = 1$. A sufficient big-M value is $M = \text{MaxMaint}^{\omega\delta}$.

$$m^{\omega\delta}_{j} - m^{\omega\delta}_{i} \geq \text{Used}^{\delta}_{\text{turn}}(i,j) + \text{Used}^{\delta}_{\text{leg}}(j) - M(1 - \sum_{r \in R} x_{ijr} + z^{\omega}_{ij}), \tag{13}$$
$$\forall (i,j) \in C_{\text{AC}},\ \omega \in \Omega,\ \delta \in \Delta_\omega$$

Scheduling maintenance resets the book-keeping variables to the value of the current connection.

$$m^{\omega\delta}_{j} \geq \text{Used}^{\delta}_{\text{turn}}(i,j) + \text{Used}^{\delta}_{\text{leg}}(j) - M(1 - z^{\omega}_{ij}), \tag{14}$$
$$\forall (i,j) \in C_{\text{AC}},\ \omega \in \Omega,\ \delta \in \Delta_\omega$$

Maintenance can only be carried out on connections that have a sufficient duration and that occur at a station that has appropriate equipment. No maintenance must be scheduled on a connection which is not used by an aircraft as well.

$$\sum_{\omega \in \Omega} z^{\omega}_{ij} \leq \sum_{r \in R} x_{ijr}, \qquad \forall (i,j) \in C_{\text{AC}} \tag{15}$$

The constraint also restricts the number of maintenance checks per connection to one. This aims at distributing the maintenance checks so that not too many are scheduled at one station at a time.

5 Results and future work

Results are pending and will be presented at the conference. However initial experiments have shown that constraints (9) and (10) make the problem harder to solve, possibly destroying the multi-commodity flow problem structure. Therefore if the results will be discouraging excluding constraints (7) through (10) may be considered.

The authors expect the problem to be too hard to be solve in reasonable time for medium and large instances, i.e. up to 2000 flight, 30 aircraft, and 100 crew blocks. Therefore other algorithms have to be explored. An obvious choice would be methods that were proposed in the literature (see Section 3), namely column generation and Bender's decomposition.

References

Barnhart, C., N.L. Boland, L.W. Clarke, E.L. Johnson, G. Nemhauser, and R.G. Shenoi. 1998. "Flight string models for aircraft fleeting and routing." *Transportation Science* 32:208–20.

Cohn, A.M., and C. Barnhart. 2003. "Improving crew scheduling by incorporating key maintenance routing decisions." *Operations Research* 51:387–96.

Cordeau, J-F., G. Stojkovic, F. Soumis, and J. Desrosiers. 2001. "Benders decomposition for simultaneous aircraft routing and crew scheduling." *Transportation Science* 35:375–88.

Desaulniers, G., J. Desrosiers, Y. Dumas, M. Solomon, and F. Soumis. "Daily aircraft routing and scheduling." *Management Science*, vol. 43.

Klabjan, D. 2005. "Large-Scale Models in the Airline Industry." In *Column Generation*, edited by G. Desaulniers, J. Desrosiers, and M.M. Solomon, 163–195. Springer US.

Klabjan, D., E. Johnson, G. Nemhauser, E. Gelman, and S. Ramaswamy. "Airline crew scheduling with time windows and plane-count constraints." *Transportation Science*, vol. 36.

Mercier, A., J-F. Cordeau, and F.A. Soumis. 2005. "A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem." *Computers & Operations Research* 32:1451–76.

Mercier, A., and F.A. Soumis. "An integrated aircraft routing, crew scheduling and flight retiming model." *Computers & Operations Research*, vol. 34.

Papadakos, N. 2009. "Integrated airline scheduling." *Computers & Operations Research* 36:176–95.

Sandhu, R., and D. Klabjan. 2007. "Integrated Airline Fleeting and Crew–Pairing Decisions." *Operations Research* 55:439–56.

Weide, O. 2009. "Robust and integrated airline scheduling." Ph.D. diss., University of Auckland.

Kick Strength and Online Sampling for Iterated Local Search

Thomas Liddle
School of Mathematics, Statistics and Operations Research
Victoria University of Wellington
New Zealand
thomas.liddle@msor.vuw.ac.nz

Abstract

A key decision when using Iterated Local Search (ILS) as an optimisation metaheuristic is the determination of the kick's nature and strength. This paper investigates methods of improving ILS through discovering the nature of the neighbourhoods in the search using two methods: adjusting the kick strength *before* the search and sampling the neighbourhood of local optima at each iteration to discover the best search direction to explore. In addition a new method of dynamically adjusting the kick strength *during* the search is developed to avoid needing to test multiple kick strengths separately. These methods are tested on the Travelling Salesman Problem using two-opt and dynasearch local search, and it can be concluded that: increasing the kick strength can significantly improve the performance of ILS but the best strength is problem instance dependent; dynamically sampling the kick strength during the search can provide better performance than a single kick and is an effective way of getting better results without explicitly testing multiple kick strengths; online sampling can improve the performance of ILS, but the best method of sampling is problem instance dependent; and dynasearch significantly outperforms two-opt local search for small kick strengths.

Key words: Metaheuristics, iterated local search, travelling salesman problem, dynasearch, online sampling.

1 Introduction

Iterated Local Search (ILS) (Lourenço, Martin, and Stützle 2003) is a metaheuristic search technique which aims to find good solutions to difficult combinatorial optimisation problems in a reasonable amount of computational time. It is an iterative procedure which performs a biased sampling of local optima (hilltops) in the search space by randomly changing the current solution at each iteration (performing a "kick") and then using a local search heuristic to find a new local optima. Thus ILS prevents getting stuck in local optima (the main disadvantage of heuristics). ILS can operate with "black-box" components; it does not need to know the local search heuristic or how the kick is performed.

The current solution at each iteration in ILS has a *neighbourhood*, which is the set of solutions that all possible locally optimised kicks can reach (the set of all neighbouring hilltops). Standard implementations of ILS perform a single kick blindly without searching for the "best" kick to perform (to reach the best neighbouring hilltop). In addition, the *strength* of the kick (how much change is made to the current solution) determines the solutions that are contained within the neighbourhood (hilltops closer or further away).

In this paper we set out to discover whether adjusting the kick strength in ILS can improve the performance; whether sampling the neighbourhood at the current solution to find a better search direction can improve the performance; whether a method of dynamically adjusting the kick strength *during* the search can improve or maintain the performance without needing to test multiple kick strengths; and whether using a more powerful local search method affects the outcome of neighbourhood sampling and dynamic kick strength choice.

The remainder of this paper is structured as follows: Section 2 provides some background information, Section 3 proposes two new variations of ILS, Section 4 outlines the experimental setup, Section 5 gives results and discussion, and Section 6 offers some conclusions and suggestions of further work.

2 Background

We use the Travelling Salesman Problem (TSP) to test the proposed variations of ILS, as it is an NP-hard combinatorial optimisation problem which is used as a benchmark for testing new optimisation techniques (Lawler et al. 1985). The TSP is the problem of finding the shortest-distance tour around n cities, and can be informally described by imagining a salesman who starts out at his home city and needs to visit a set of other cities exactly once before returning home, wanting to find the best visiting order to minimize his distance travelled.

ILS is a metaheuristic that has been applied to many different optimisation problems, including the TSP and many scheduling problems. The general procedure can be seen in Algorithm 1 (Lourenço, Martin, and Stützle 2003).

Algorithm 1 Iterated Local Search

s_0 = GenerateInitialSolution
s^* = LocalSearch(s_0)
repeat
  s' = PerformKick($s^*, history$)
  s'^* = LocalSearch(s')
  s^* = AcceptanceCriterion($s^*, s'^*, history$)
until TerminationCriterion

ILS begins with a local optima from using the local search heuristic on an initial solution (generated by a construction heuristic such as nearest neighbour construction). It then iteratively performs a kick on the current solution (by randomly changing some elements of the solution), obtains a new local optima (using local search) and then determines if this new solution should be used as the current so-

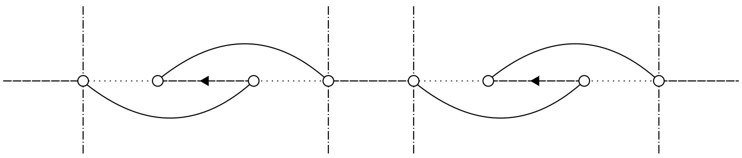


Figure 1: Dynasearch with two-exchanges.

lution for the next iteration by testing it against some *acceptance criterion.* The search is terminated when a *termination criterion* is reached.

Two local search methods are considered in this paper, two-opt local search and dynasearch (with two-exchanges). *Two-opt* search is a simple local search method which iteratively makes the best two-exchange until no improving exchange can be found (Lin 1965). A two-exchange selects and removes two non-adjacent links, joining the tour back up in the only other way possible. A single iteration of two-opt search evaluates all possible two-exchanges and performs the exchange that provides the largest improvement in length over the current tour. *Dynasearch* (Congram, Potts, and van de Velde 2002) is a local search method which uses dynamic programming to find the best set (largest total improvement in distance) of non-overlapping two-exchanges on the current tour (see Figure 1), performing all of them in a single iteration. It performs exactly the same number of exchange evaluations as two-opt local search per iteration, but will find at least as much (but likely more) improvement than two-opt can.

We adopt the "double-bridge" move (a particular four-exchange) as the kick component, as it has been found to be very effective for the symmetric TSP and has been commonly used since (Martin, Otto, and Felten 1991). The *kick strength* is the number of sequential random double-bridge moves, k, performed as a single kick, e.g. if $k = 5$, the kick component performs five random double-bridge moves.

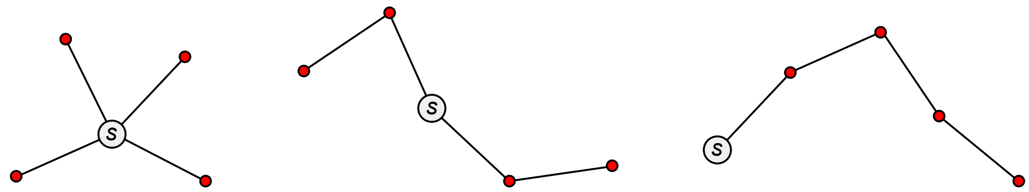
Two *acceptance criteria* are considered: *better* acceptance, where the new solution is only accepted if its tour length is less than that of the current solution; and *probability* acceptance, where in addition to better solutions, if the new solution is worse than the current, it may be accepted with a probability p if there has been a delay of d iterations without finding an improving solution. Different balances of d and p produce a search that is closer to either better acceptance or a *random restart* search. The search terminates if either an optimal solution has been found (if the tour length is known beforehand) or a maximum time limit has been reached.

3 Proposed Variations of ILS

We propose two variations of ILS: sampling the neighbourhood of the current solution at each iteration and dynamically adjusting the kick strength at each iteration.

3.1 Sampled Iterated Local Search

We propose a variation of ILS, called Sampled Iterated Local Search (S-ILS), that samples the neighbourhood of the current solution, by performing four kicks in three different branching combinations (sampling types), each of which is locally optimised before performing the next. The best local optima found (intermediate or final) is presented as the new solution to the acceptance criterion. The three different sampling types can be seen in Figure 2 and are briefly described below.



(a) (b) (c)

Figure 2: Different sampling types for S-ILS: Four branches of a single kick (a), two branches of two kicks each (b), and one branch of four kicks (c).

- *Type 1: Four branches of one kick.* A single kick of strength k is performed away from the current solution four independent times. Local search is performed on each to produce four local optima.
- *Type 2: Two branches of two kicks each.* Two independent sampling branches are followed, each of which consists of two consecutive kicks of strength k (each locally optimised before performing the next). This type looks "ahead" further than Type 1, but looks "around" less. The best of the four local optima (two intermediate and two final) is proposed as the new solution, giving the same amount of improvement opportunity as Type 1.
- *Type 3: One branch of four kicks.* A single branch of four consecutive kicks of strength k (each locally optimised before performing the next) is followed. This type consists entirely of looking ahead, and does not look around the current solution. It may be able to discover a search direction that looks promising in the short term but is in the long term.

3.2 Dynamic Iterated Local Search

We propose another variation of ILS, called Dynamic Iterated Local Search (D-ILS), which avoids the need to test multiple kick strengths to find the best one; instead it dynamically samples from a set of available kick strengths K, choosing more often those kick strengths in K that seem to be performing well so far. For ILS, $|K|$ experiments (one for each kick strength) are needed to find the best $k \in K$, but D-ILS has the potential to achieve the same or similar results using only a single experiment. It can also be incorporated into S-ILS as it only affects how the kick is performed.

We use *weights* w_k for each $k \in K$ in a probabilistic selection method. These w_k are updated during the search. This concept is similar to the use of *pheromones* in Ant Colony Optimization (Dorigo and Stützle 2004). K is specified by lower and upper kick strength limits k_l and k_u, thus $K = \{k_l, k_l + 1, \ldots, k_u - 1, k_u\}$, with $m = k_u - k_l + 1$ possible kick strengths to sample from.

A simple implementation of dynamic sampling is to use a single weights vector $\mathbf{w}$, where the next kick strength is chosen according to $\mathbf{w}$ only, that is, no memory is incorporated. Alternatively, we incorporate one level of memory into the kick strength choice, in order to evolve a balance between intensification and diversification. In this case a weights *matrix* W is kept, where each row j of W $(j = 1, \ldots, m)$ corresponds to the weights vector $\mathbf{w}_j$: the weights for choosing the next kick strength given that the previous kick strength was j.

Using a small kick strength has the potential for getting stuck in a local optima (Lourenço, Martin, and Stützle 2003). We introduce a *non-improvement bias* (NIB), which biases the strength of the kick increasingly towards larger values as the number of non-improving iterations increases. The NIB (set before the search) is used as a power to exponentially increase the bias for higher values of NIB in this paper.

The proposed method for probabilistically selecting a kick strength j from the set of possible kick strengths K, when performing a dynamically sampled kick in D-ILS is as follows. First determine the row vector of weights $\mathbf{w}$. If not using memory, this is just $\mathbf{w}$, the vector of weights. If using memory assign $\mathbf{w} = \mathbf{w}_i$, row i of the weights matrix W, which corresponds to the previous kick strength $i \in \{1, \ldots, m\}$. Then calculate the probability vector $\mathbf{p}$ from $\mathbf{w}$ as follows. Normalise the weights to be non-negative:

$$v_j = \begin{cases} w_j - w_{min} & \text{if } w_{min} < 0 \\ w_j & \text{otherwise} \end{cases}, \qquad \text{for } j = 1, \ldots, m \tag{1}$$

where $w_{min} = \min_j w_j$. Calculate the normalisation increment a as:

$$a = \frac{w_{max} - w_{min}}{k_u - k_l} \tag{2}$$

where $w_{max} = \max_j w_j$. Calculate the non-improvement biases b_j:

$$b_j = \begin{cases} 0 & \text{if NIB} = 0 \\ (\beta j)^{\text{NIB}} & \text{otherwise} \end{cases}, \qquad \text{for } j = 1, \ldots, m \tag{3}$$

where β is the number of non-improving iterations that have elapsed. Add the normalisation increment and non-improvement biases to $\mathbf{v}$:

$$v_j = w_j + a + b_j, \qquad \text{for } j = 1, \ldots, m \tag{4}$$

Finally, calculate p_j for each $j \in K$ by normalising the v_j between 0 and 1:

$$p_j = \frac{v_j}{\sum_{i=1}^{m} v_i}, \qquad \text{for } j = 1, \ldots, m \tag{5}$$

The probability vector $\mathbf{p}$ can then be used to sample a kick strength to use from K. After a kick strength j is selected, the kick is performed, a local optima is found using local search, and the weight corresponding to the chosen kick strength is then updated via

$$w_j \leftarrow w_j + length_{last} - length_{new} \tag{6}$$

where $length_{last}$ and $length_{new}$ are the tour length of the current (before the kick) and new solutions' local optima respectively. If selection with memory is used, then replace w_j with w_{ij} in Equation (6) where i is the kick strength chosen the previous iteration.

Studies of Ant Colony Optimization (ACO) have shown that initialising the pheromones (which serve a similar purpose to the weights in D-ILS) to an arbitrary value can adversely affect the algorithm's performance (Dorigo and Stützle 2004). Performance increases when the pheromones are initialised to a value that is indicative of the average change in pheromone level. In the context of D-ILS, we propose to estimate this value (as it cannot be found without running a full search) by performing a kick and local search of each available kick strength on the generated initial solution of ILS independently of each other, and then setting the each weight to be the average of the improvement found by each kick strength. The general procedure for D-ILS can be seen in Algorithm 2.

Algorithm 2 Dynamic Iterated Local Search

s_0 = GenerateInitialSolution
s^* = LocalSearch(s_0)
$\mathbf{w}$ = InitialiseWeights(s^*)
repeat
  s' = PerformDynamicKick($s^*, \mathbf{w}, \mathit{history}$)
  s'^* = LocalSearch(s')
  $\mathbf{w}$ = UpdateWeights($s^*, s', \mathbf{w}$)
  s^* = AcceptanceCriterion($s^*, s'^*, \mathit{history}$)
until TerminationCriterion

Table 1: Problem instance properties and maximum time limits for all experiments.

	berlin52	eil101	tsp225	a280	pcb442
Number of cities	52	101	225	280	442
Optimal tour length	7542	629	3919	2579	50778
Search time limit (s)	10	20	120	120	300

4 Experimental Setup

This section describes the experimental setup for testing the proposed methods, including the problem instances tested and the configurations of the parameters in each method used in the experiments.

Five instances of the symmetric TSP with Euclidean distances varying in size have been chosen from (Heidelburg University 2010) for testing the proposed variations (see Table 1).

Initial experiments of ILS with two-opt local search using better acceptance and different combinations of $p \in \{0.05, 0.25, 0.5, 1\}$ and $d \in \{1, 5, 10, 20\}$ for probability acceptance showed that better acceptance generally performed better than probability acceptance; of the 16 probability acceptance configurations, the best was the one with smallest p and largest d ($p = 0.05$ and $d = 20$); and ILS performed much better than random restart search (increasingly so as the problem instance size increased). Hence, each problem instance was tested using better acceptance and probability acceptance with $p = 0.05$ and $d = 20$, and each of two-opt and dynasearch local search. ILS and each S-ILS sampling type were tested on each of $k \in \{1, 2, \ldots, 10\}$ for the berlin52 and eil101 instances, and each of $k \in \{1, 2, \ldots, 20\}$ for the other instances. D-ILS used these same ranges for the available kick strength set K, and each of NIB $\in \{0, 1, 2\}$ with and without memory, a total of 24 D-ILS configurations. Each experiment was run on the same 50 different random seeds. Table 1 shows the maximum time limits allowed for all experiments for each instance.

5 Results and Discussion

Table 2 shows results graphs for ILS and each type of S-ILS on three of the problem instances (eil101, tsp225 and pcb442), using better acceptance and each of two-opt and dynasearch local search heuristics. The distribution of results is shown by using the median and lower and upper quartile values of the percentage extra distance travelled over the optimal solution. The graphs show a banana-like shape in the distribution for two-opt, indicating that adjusting the kick strength can significantly reduce the extra distance travelled over the optimal solution. There is a much steeper decrease in performance for larger kick strengths for dynasearch, indicating that optimising the kick strength may just be a matter of trying a few small kick strengths. For small kick strengths, dynasearch performs much better than two-opt (achieving the optimal solution for the tsp225 instance), but significantly worse for large kick strengths.

Across all kick strengths, S-ILS Type 1 (S-ILS(1)) performed similarly to ILS, and S-ILS(3) performed generally worst (with a narrower range of good kick strengths and a steeper decrease in performance for increasing kick strengths). The best kick strength varies between problem instances, sampling types and local search methods, but the performance of the best kick strength for each of ILS and the three S-ILS types is similar for each instance (though slightly worse for S-ILS(3) in each case). The results for D-ILS (not shown) generally sat somewhere within the "average" distribution of each sampling type, giving better results than standard ILS but not as good as ILS with the best kick strength.

Table 3 shows that these results generally hold for probability acceptance as well, with S-ILS(1) performing the best in most problem instances and S-ILS(3) the worst. This table shows the best two configurations of D-ILS for each instance and acceptance criterion, with a prevalence of Type 1 sampling being among the best configurations. It is inconclusive whether memory is useful or not, or which value of NIB to use (though other results suggest that NIB $= 2$ performs slightly worse than the others). At least one D-ILS configuration achieved the overall best performance out of all ILS and S-ILS configurations without needing to test multiple kick strengths for the berlin52, eil101 and a280 problem instances.

Table 4 shows that in four out of five problem instances, all 24 D-ILS configurations improved on ILS (the other instance gave 22 out of 24 improving configurations), which is a promising result. The amount of improvement achieved varied among configurations, however, but the best configurations gave significant reductions in extra distance travelled (up to 75% for the larger instances). The 100% reduction for the berlin52 problem instance corresponds to the optimal solution being found (at least 50% of the time) by S-ILS(2), S-ILS(3) and all D-ILS configurations, where standard ILS's median result was 0.23% from the optimal solution. Which configurations gave the best results were not consistent, but the worst performers were those with Type 3 sampling or NIB $= 2$, and removing these in future would reduce the number of D-ILS configurations to just 12. In addition, D-ILS was rather sensitive to the kick strength set, indicating that further research is required to improve the selection method. Because dynasearch's best kick strength was very low and showed significant performance degradation, the results of D-ILS when using dynasearch indicate that is unlikely to be necessary if using dynasearch as the local search heuristic. Table 4 also shows that the three S-ILS types were able to match the performance of ILS, and in some cases improve on it, though only by a small amount. The sampling types are of most use within D-ILS.

Probability acceptance generally produced worse results than better acceptance for ILS across all instances, but interestingly produced similar results for all S-ILS types, indicating that the neighbourhood sampling may be able to "weed out" the worse solutions that ILS is forced to propose.

Table 2: Distribution of the percentage extra distance travelled over the optimal solution across 50 seeds (y axis) for the eil101, tsp225 ad pcb442 problem instances using better acceptance, testing different configurations of S-ILS against ILS.

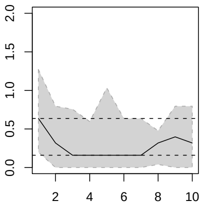
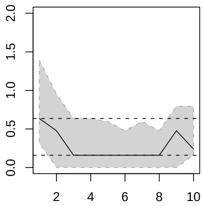
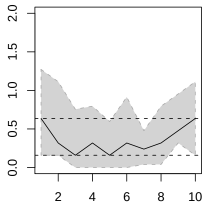
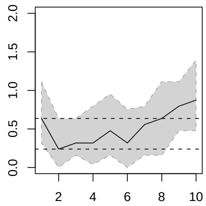
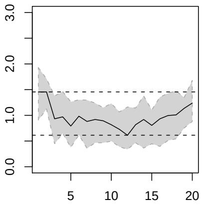
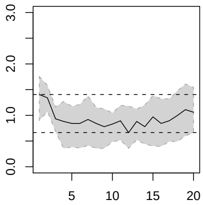
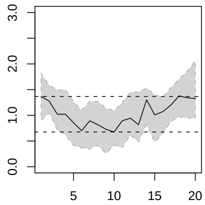
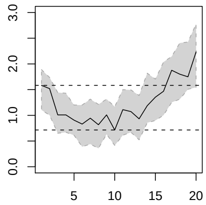
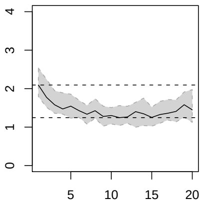
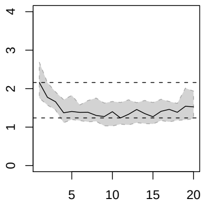
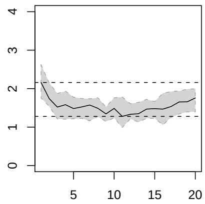
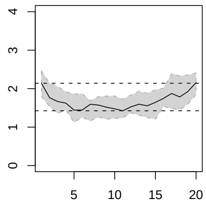
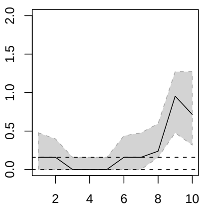
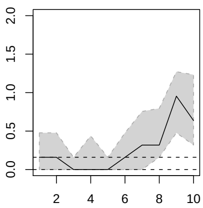
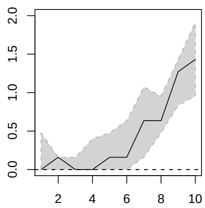
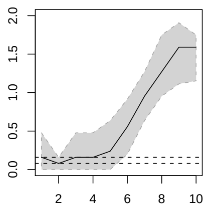
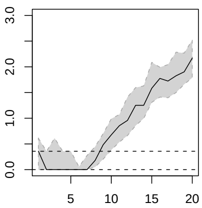
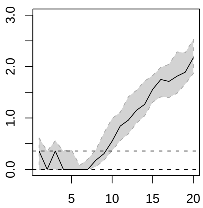
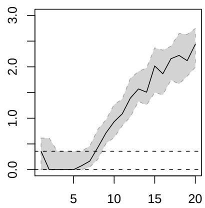
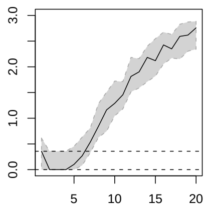
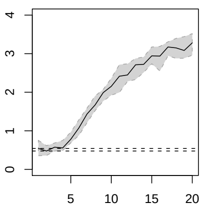
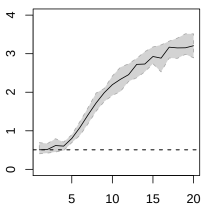
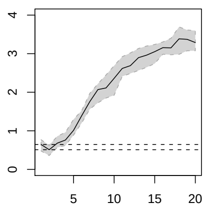
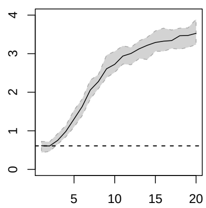
Instance	ILS	S-ILS(1)	S-ILS(2)	S-ILS(3)
		two-opt		
eil101				
tsp225				
pcb442				
		dynasearch		
eil101				
tsp225				
pcb442				

Table 3: Summary of experimental results from testing ILS, S-ILS, and D-ILS using two-opt local search, showing the best and worst performing method from ILS and the three S-ILS types, as well as the best two D-ILS configurations and whether they achieved the overall best median result.

Problem instance	Acceptance criterion	ILS/S-ILS Best	ILS/S-ILS Worst	Best D-ILS	Sampling	Memory?	NIB	Overall Best?
berlin52	Better	S-ILS (3)	S-ILS (1)	1.	None	Yes	2	Yes
				2.	None	No	2	Yes
	Prob	ILS	S-ILS (1)	1.	None	Yes	2	Yes
		S-ILS (3)		2.	None	No	2	Yes
eil101	Better	S-ILS (1)	S-ILS (3)	1.	Type 2	Yes	0	Yes
				2.	None	Yes	2	Yes
	Prob	S-ILS (1)	S-ILS (3)	1.	Type 2	Yes	0	Yes
		S-ILS (2)		2.	Type 2	No	1	Yes
tsp225	Better	ILS	S-ILS (3)	1.	Type 1	No	2	No
		S-ILS (1)		2.	Type 1	No	0	No
	Prob	S-ILS (2)	ILS	1.	Type 1	Yes	0	No
		S-ILS (1)		2.	Type 1	No	2	No
a280	Better	ILS	S-ILS (3)	1.	None	No	2	Yes
		S-ILS (1)		2.	Type 2	No	1	No
	Prob	S-ILS (1)	ILS	1.	Type 1	Yes	2	No
		S-ILS (3)		2.	Type 2	No	1	No
pcb442	Better	S-ILS (1)	S-ILS (3)	1.	Type 1	Yes	1	No
		ILS		2.	None	Yes	0	No
	Prob	S-ILS (1)	ILS	1.	Type 2	No	0	No
			S-ILS (3)	2.	Type 1	Yes	1	No

Table 4: The number of configurations (of S-ILS or D-ILS) that performed strictly better than (% distance from the optimal solution), and better than or equal to each of standard ILS ($k = 1$) and ILS with the discovered best kick strength ($k = k_{best}$) shown as ($<$ ($\leq$), % reduction) triples. Comparisons are made against the median percentage distance from the optimal solution. Results are shown for two-opt local search and better acceptance.

Instance	Configuration	S-ILS (/3) $k = 1$		S-ILS (/3) $k = k_{best}$		D-ILS (/24)	
berlin52	ILS, $k = 1$	2 (2)	100%			24 (24)	100%
	ILS, $k = k_{best}$			0 (3)		0 (24)	
eil101	ILS, $k = 1$	0 (3)				22 (23)	75%
	ILS, $k = k_{best}$			0 (2)		0 (8)	
tsp225	ILS, $k = 1$	2 (2)	6%			24 (24)	45%
	ILS, $k = k_{best}$			0 (0)		0 (0)	
a280	ILS, $k = 1$	1 (3)	5%			24 (24)	73%
	ILS, $k = k_{best}$			0 (1)		0 (1)	
pcb442	ILS, $k = 1$	0 (0)				24 (24)	58%
	ILS, $k = k_{best}$			1 (1)	1%	0 (0)	

6 Conclusions

Optimizing the kick strength in ILS can significantly improve performance, although it requires multiple experiments of different kick strengths. Sampling the neighbourhood at each iteration can also improve on (or at least match) the performance of ILS, but the best method of sampling is problem instance dependent. The method of dynamic kick strength choice we developed appears quite promising and can improve on ILS when using two-opt search, but is sensitive to the range of available kick strength it takes and thus does not perform as well as using dynasearch. Dynasearch performs better than two-opt search for small kick strengths but (counter-intuitively) significantly worse for large kick strengths for ILS and S-ILS.

There is much further research to be done, including testing these methods on a wider set of problem instances and more local search methods; investigating other methods for dynamic kick strength choice to reduce its sensitivity to the available kick strengths; investigating why dynasearch exhibits extreme fluctuations in performance; development of other neighbourhood sampling types and methods; and testing of these methods on other problem types to see if the observed results hold.

References

Congram, R. K., C. N. Potts, and S. L. van de Velde. 2002. "An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem." *INFORMS Journal on Computing* 14 (1): 52–67.

Dorigo, M., and T. Stützle. 2004. *Ant Colony Optimization.* MIT Press.

Heidelburg University. 2010. TSPLIB. `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95`, accessed October, 2010.

Lawler, E. L., J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. 1985. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization.* Wiley.

Lin, S. 1965. "Computer solutions of the traveling salesman problem." *Bell System Computer Journal* 44:2245–2269.

Lourenço, H. R., O. Martin, and T. Stützle. 2003. "Iterated local search." In *Handbook of Metaheuristics*, edited by F. Glover and G. Kichenberger, 321–353. Kluwer.

Martin, O., S. W. Otto, and E. W. Felten. 1991. "Large-step markov chains for the traveling salesman problem." *Complex Systems* 5:299–326.

Optimal Public-Transport Transfer Synchronization Using Operational Tactics

Albert Ang, Michael McIvor
Department of Civil and Environmental Engineering
University of Auckland
New Zealand
a.ceder@auckland.ac.nz

Abstract

Transfers in public-transport are used to create a more efficient network, by reducing operational costs and allowing more flexible route planning. However because of the stochastic nature of traffic, scheduled transfers do not always occur, thus increasing the total passenger travel time and reducing the attractiveness of the public-transport service. This work analyzes how to use selected operational tactics in public-transport networks for increasing the actual occurrence of scheduled transfers. A model is developed to determine the impact of instructing vehicles to either hold at or skip certain stops, on the total passenger travel time and the number of simultaneous transfers. The model is comprised of two components. First, a simulation of public-transport network examines the two tactics for maximizing the number of transfers. Second, an ILOG optimization model is used for optimal determination of the combination of the two tactics to achieve the maximum number of simultaneous transfers. An Auckland bus network was created, as a case study, to verify the impact of the model's application. The results show that applying online operational tactics dramatically improved the frequency of simultaneous transfers. The concept has large potential for increasing the efficiency and attractiveness of public-transport networks which involve scheduled transfers.

Keywords: Operational tactics, real time tactics, public transport networks, transfer, transfer synchronization, transfer optimization.

1 Introduction

In any public-transport (PT) network, it is impractical to have routes between every conceivable trip origin and every conceivable trip destination. There are too many possible routes and the service cannot be economically provided. Transfers in the network allow routes to complement each other meaning fewer routes are able to provide the same level of coverage. This in turn enables higher frequency services and an easier network to understand and remember, increasing the attractiveness as a whole (Mees, 2000).

Transfers in general allow more flexibility in route planning and more effective use of services which results in a more efficient PT network, associated environment

(Waterson et al., 2003), economic (Jakob et al., 2006), health and social benefits (Barton and Tsourou, 2000) (Frank et al., 2006).

Conversely transfers are cited as a key reason for PT being less attractive than cars (Knoppers and Muller, 1995). Due to the stochastic nature of travel times, dwell times and passenger demands in PT networks, two vehicles which are scheduled to arrive simultaneously at the same stop (a scheduled transfer) have a small encounter probability of (a direct transfer). This can cause frustration, longer passenger waiting times and a less efficient system as a whole.

Transfer synchronization aims to increase the number and probability of bus encounters. Some researchers e.g. Ceder, Golany et al. (Ceder et al., 2001), Domschke (1989) and Fleurent, Lessard and Séguin (2007) have used mathematical relationships to generate timetables with the maximum opportunity for direct transfers.

Another way of improving the occurrence of transfers is by using “operational tactics” in real time, first outlined by Ceder (2007). Hadas and Ceder (2008) evaluated the specific tactics of holding vehicles at stops in anticipation of connections and instructing skip-stops and shortcuts of routes to meet subsequent connections. Use of these tactics was assessed with simulation on synchronized transfers, in a complex although contrived, discrete model. The stochastic nature of transfers was taken into account with estimates of probabilities of encounters, based upon the difference in arrival times. It was shown that large increases in the number of direct transfers and small reductions in the total user travel time could be achieved. They suggested that this would dramatically increase the comfort of passengers and the attractiveness of the network (Hadas and Ceder, 2008).

This research analyses how instructing vehicles to either hold at or skip certain stops can increase the occurrence of direct transfers and improve the efficiency as measured by total passenger travel time in the PT network as a whole. It develops a model and methodology to determine the effect of various tactics at various stops and how these tactics should be best applied.

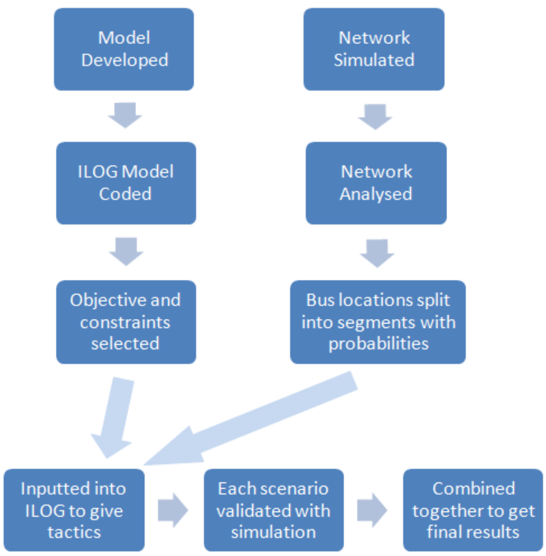
A simulation of an appropriate PT network in Auckland was developed in order to determine the effect of the network’s stochastic nature on direct scheduled transfers, with and without operational tactics.

These were integrated in order to assess the impact of operational tactics for the simulated Auckland network.

2 Methodology

2.1 Model Derivation

A model was derived from first principles in order to assess what the impact of operational tactics was on the network. This impact was measured by two criteria, the number of direct transfers, and the change in total passenger travel time (ΔTPTT). It is intended for use with deterministic data.



Parameters:

- R is the set of all bus routes
- N is the set of all bus stops
- $Y_{X\mu}^{n}$ = 1 if bus X is late causing a missed direct transfer with bus μ at stop n, before any tactics and n is a transfer point between X and μ
  = 0 otherwise
- f the ratio of average bus to average pedestrian travel speed through urban areas
- e_X^n number of passengers entering the network onto bus X at stop n (i.e. excl. transferring passengers)
- l_X^n number of passengers leaving the bus network from bus X at stop n (i.e. excl. transferring passengers)
- $t_{X\mu}^{n}$ number of passengers wishing to transfer from bus X to bus μ at stop n
- d_X^n dwell time of bus X at stop n
- p_X^n number of passengers riding bus X as it arrives at stop n.
- c_X^n the estimated time between bus X

Figure 1: Methodology flow chart

  arriving at stop n and arriving at the previous stop (without tactics)
- Q_X^i vector listing all stops of route X in order defined by a natural number position, i
- V_X^n vector listing the positional index, i, of stop n on route X (0 if not on route)
- h_X headway between buses route X
- T_X initial time difference that bus X is behind schedule
- m_X total number of stops on route X
- k_X positional index, i, of the next stop on route X that the bus will arrive at
- g_X elapsed time since bus X arrived at the previous stop on the route to its current position

Decision Variables

- W_X^n time to hold bus X at stop n
- S_X^n = 1 if bus X skips stop n
  = 0 otherwise
- $Z_{X\mu}^{n}$ = 1 if bus X is late causing a missed direct transfer with bus μ at stop n, after any tactics and n is a transfer point between X and μ
  = 0 otherwise
- $DT_{X\mu}^{n}$ = 1 if a direct transfer occurs for bus X with μ at stop n after any tactics
  = 0 otherwise

Objectives:

(1) Minimise ΔTPTT = $\sum_{X\in R} \quad \sum_{n\in N}$

(a) $W_X^n \ \{ \ p_X^n - l_X^n + e_X^n \ + (\sum_{\mu\in R,\mu\neq X} \ (1 - Y_{\mu X}^n)\, t_{\mu X}^n - t_{X\mu}^n) + \ \sum_{i=V_X^n+1}^{m_X} [\ e_X^{Q_X^i} + \sum_{\mu\in R,\mu\neq X} \ t_{X\mu}^{Q_X^i} (1 - Z_{\mu X}^{Q_X^i})] \ \}$

(b) $+ \ S_X^n \ \{ l_X^n \, f c_X^{Q_X^{V_X^n+1}} + e_X^n (h_X - \ T_X + \sum_{i=k_X}^{V_X^n-1} [S_X^{Q_X^i} \, d_X^{Q_X^i} - W_X^{Q_X^i}]) - d_X^n (p_X^n + \sum_{i=V_X^n+1}^{m_X} [\ e_X^{Q_X^i} + \ \sum_{\mu\in R,\mu\neq X} \ t_{X\mu}^{Q_X^i} (1 - Z_{\mu X}^{Q_X^i})])\}$

(c) $+ \sum_{\mu\in R,\mu\neq X} \ [t_{X\mu}^n \{ \ Z_{X\mu}^n \ (h_\mu - T_X \ + \sum_{i=k_X}^{V_X^n-1} [S_X^{Q_X^i} \, d_X^{Q_X^i} - W_X^{Q_X^i}]) \ - Y_{X\mu}^n \ (h_\mu - T_X)\}]$

(2) Maximise = $\sum_{X\in R} \sum_{\mu\in R,\mu\neq X} \sum_{n\in N} \ DT_{\mu X}^n$

Subject to:

(3) $\sum_{i=k_A}^{V_A^n} c_A^{Q_A^i} - g_A - \sum_{i=k_A}^{V_A^n-1} \left[S_A^{Q_A^i} d_A^{Q_A^i} - W_A^{Q_A^i}\right] - \sum_{i=k_B}^{V_B^n} c_B^{Q_B^i} + g_B - d_B^n - W_B^n + \sum_{i=k_B}^{V_B^n-1}[S_B^{Q_B^i} d_B^{Q_B^i} - W_B^{Q_B^i}] \leq 9999Z_{AB}^n$

(4) Y_{AB}^n = 1 if true: $\sum_{i=k_B}^{V_B^n} c_B^{Q_B^i} - g_B + d_B^n \leq \sum_{i=k_A}^{V_A^n} c_A^{Q_A^i} - g_A$
= 0 otherwise, or any variables are undefined

(5) $DT_{\mu X}^n$ =1 if true: $Z_{\mu X}^n + Z_{\mu X}^n = 0$, $t_{\mu X}^n + t_{\mu X}^n > 0$
=0 otherwise

(6) $S_X^n \ (\sum_{\mu \in R, \mu \neq X} \ [t_{X\mu}^n + t_{\mu X}^n]) = 0$

(7) S_X^n , $W_X^n = 0$ when $V_X^n < k_X$

(8) $S_X^n + S_X^{Q_X^{V_X^n+1}} \leq 1$

(9) $S_X^n \ W_X^n = 0$

Assumptions:

- Foreknowledge of the route information, including travel times, passenger demands, transferring passengers and dwell times. In this way the model was designed to work with deterministic data.
- Passenger demands do not change with a varying bus arrival time.
- Transfers are scheduled and the next buses are on time.
- Any waiting passengers skipped, or those that miss their transfer connections will wait for the next bus which is also on time.
- Transfers can only occur at a single physical bus stop, this means that the time for passengers to transfer is ignored.
- Stops where people want to transfer cannot be skipped, nor can more than one stop in a row be skipped.

Equation 1 represents the change in total passenger travel time (ΔTPTT). This consists of three parts, the holding, skip stop and a late to transfer effect, 1a, 1b and 1c respectively.

Holding a vehicle in (1a) increases the travel time for the passengers on the bus and those waiting for the bus further along the route. Determining the number of passengers on the bus is complicated by whether or not passengers transferring onto the bus have boarded and made the transfer. By assuming that the bus would not be holding if these passengers had already boarded, and will only hold until they board, these passengers can be ignored. This is improved by assuming they would have made the transfer without the application of any tactics at all and increasing the passengers held on board by $(1 - Y_{X\mu}^n)\ t_{X\mu}^n$.

Those advantaged by the skip stop (2a) with the time saving d_X^n are those who are currently on the bus $p_X^n$and those that will get on the bus in the future.

Those that wanted to alight and those that wanted to board are now disadvantaged.

The approximation of the extra travel time for these passengers is$f c_X^{Q_X^{V_X^n+1}}$. It is assumed that the extra dwell time at the subsequent bus stop from extra passengers alighting is negligible; it is unlikely a stop would be skipped if large numbers were alighting.

The wait for those that wanted to board skipped is $h_X - T_X$ if no tactics were applied, but if tactics had been applied at previous stops the wait is $h_X - T_X + \sum_{i=k_X}^{V_X^n-1} [S_X^{Q_X^i} d_X^{Q_X^i} - W_X^{Q_X^i}]$.

For (1c) if a bus is late to a scheduled transfer, the passengers on board that wanted to transfer are disadvantaged. Transferring passengers on the earlier bus will only have to wait for the other bus to arrive, and any tactics to this bus affecting these passengers are taken into account in 1a and 1c.

For transferring passengers on bus X missing the connection to bus μ at stop n, the wait is $h_\mu - T_X$ pre-tactics and $h_\mu - T_X + \sum_{i=k_X}^{V_X^n-1} [S_X^{Q_X^i} d_X^{Q_X^i} - W_X^{Q_X^i}]$ post-tactics. If a direct transfer is made then there is no transfer waiting delays at all. The difference between the two delays before and after tactics is the ΔTPTT relating to making/missing direct transfers that occurs from applying tactics. All Xμ combinations are summed.

Equation 2 represents the total number of buses that make a direct transfer. (3) and (4) determine Y and Z respectively. If either route does not travel to stop n, c_B^n, c_A^n is undefined and Y, Z is taken as 0. Z was taken as a decision variable to enable use with IBM ILOG. (5) determines if a bus makes a direct transfer while (6) states stops cannot be skipped where passengers want to transfer. (7) ensures there is no tactics on stops that a bus is not going to or is already passed. Finally (8) allows only one skipped stop in a row and (9) provides for no skipping and holding at the same stop.

2.2 IBM ILOG Optimization

This model was then inputted into the optimization software IBM ILOG to enable the best operational tactics for any situational input or objective to be computed.

The CP programming tool was used, which required the decision variables to be discrete. Hold time was calculated to the second, with 0 indicating no holding. A maximum hold time constraint was added of no more than half the headway of the route, $W_X^n \leq ½h_X$

The ILOG model was integrated with excel spreadsheets so data initialization was easier for a wide range of network types and situations.

3 Application to Auckland

3.1 Auckland PT routes

To assess the effectiveness of the model and potential for implementation, it was applied to an Auckland PT network. Three routes with two transfer points from the Northern bus network were chosen as an Auckland PT network to evaluate. These routes and the Northern Busway in general have been designed around the concept of transfers. There are no current synchronized or scheduled transfers, but there is future potential for their implementation here.

The routes were the 155 route on the Northern Busway, the 880 route which runs north-south (east of the busway) and the 913 feeder loop which transfers with both (see Figure 2 below).

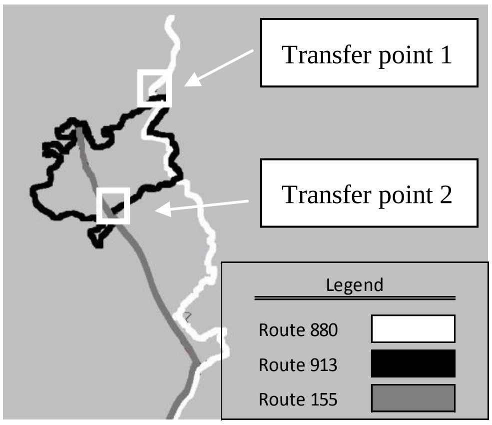


Figure 2: North Shore City with three bus routes. Transfer point 1 and 2 located in white squares. Source: Google Maps.

ARTA provided the information for these three routes. Route 880 starts from Long Bay Regional Park, Route 913 starts from Massey University, Albany and Route 155 starts from Albany Station Platform 2a.

Route 880 and 913 intersect at transfer at point 1 on Bute Road, Browns Bay. Route 913 and 155 connect at transfer point 2 at the Constellation Bus Station, Albany.

3.1 Microscopic simulation model

The versatile microscopic traffic simulation software used was TransModeler 2.6 created by the Caliper Corporation in order to simulate transit vehicles in a stochastic network.

The simulation consisted of carriageways that are used by the bus route only. Due to the complexity of the model and time limitations, the model did not include external traffic, intersection control and side streets. A function within TransModeler however allowed the 'road class' to be selected to suit the speed-density function of a particular segment of road. Route 155 (Northern Busway) was classified as an Expressway. Routes 880 and 913 are a mixture of minor and major arterial roads.

This software allowed the stochastic transit routes to be simulated as the derived model used deterministic data.

3.2 Data collection

Bus route data for the different routes were collected from the Auckland Transport Authority (ARTA). Important information such as the road layout and physical stops were used for the construction of the simulation. Other data had to be assumed.

Each road segment was given a road class. Route 155 in particular was classified as an expressway as there was no traffic variability on the busway.

The travel time of buses from one stop to another could be obtained from the outputs produced by TransModeler. Parameters which affected the speed of buses such as the acceleration function were left to their recommended default values.

The dwell times of buses depended on the parameters in TransModeler. The fleet size of the buses were set to having a capacity of 40 sitting and 20 standing in order to take into bus 'crowding'. The dead time, passenger boarding time and passenger alighting time was set to the default values of 4, 3.5 and 2.2 seconds respectively. These times were consistent with a study by (Dueker et al., 2004).

Passenger demand was assumed by using the known numbers of stops for each route. The passenger demand was set to be higher at transfer stops in order to simulate the event of passengers transferring.

3.3 Analysis

The tools developed were then used together in the application to the selected North Shore bus network to assess the potential impact of operational tactics. This was done by first simulating the network with no operational tactics used. The departure times of each bus was adjusted so that the mean arrival time at each transfer point was the same. The percentage of direct transfers at each transfer point was evaluated.

Also from this simulation the different bus locations before the transfer point was evaluated. When the 913 bus arrived at a stop four stops before the transfer point, the other bus could be in many different locations. By dividing the other bus's route up into small segments the probability of it being present was found.

The 'situation' of the buses in each of these segments was plugged into the ILOG optimization model, assuming that any bus in a particular segment was in the middle of that segment, and using deterministic data estimated from the average quantities already witnessed in the network. The selected objective criterion was the number of direct transfers. This was coupled with the constraint that no tactics could be implemented unless both the total passenger travel time of the network was decreased and a direct transfer was predicted.

Finally each of these 'situations' was simulated in TransModeler with the tactics employed. This enabled the deterministic model to integrate and be evaluated with a stochastic environment. This enabled a validation percentage to be obtained outlining the percentage of times applying those tactics to a particular segment actually resulted in a direct transfer.

The probabilities for each segment were then multiplied by the percentage of the time that the operational tactics actually enabled the direct transfer and summed. This resulted in a final percentage for the amount of scheduled direct transfers that actually resulted in a direct transfer to be ascertained, for each headway spacing and transfer point.

4 Results

It was shown that because of the stochastic nature of the network that direct transfers occurred infrequently without tactics (see Figure 3 right).

The occurrence of direct transfers without tactics was rare. This was expected due to the stochastic nature of the network. The different headway saw no statistically significant difference between the percentages of direct transfers at each transfer points. Transfer point two had significantly lower percentages of direct transfers than transfer point one. This was because it was further from the start of the routes, so there was greater variability in the travel time, the arrival time and

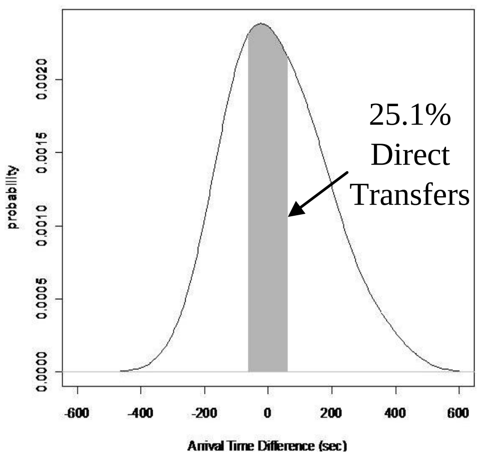


Figure 3: Percentage of successful transfers and arrival time difference at transfer point 1 with 5 minute headways.

therefore the arrival time differences.

The tactics implemented by the model made a substantial improvement to the percentage of direct transfers for all headways and transfer points (see Table 1 below). The tactics implemented also improved the efficiency of the network, reducing the total passenger travel time substantially.

Table 1: The percentage of direct transfers achieved for each scenario

Percentage of direct transfers		Without tactics	After Tactics	After Validation	Ratio of increase
Transfer1	5 min hw	25.1%	55.2%	51.9%	2.1
	20min hw	23.1%	77.6%	48.7%	2.1
Transfer2	5min hw	9.7%	20.3%	20.3%	2.1
	20min hw	13.7%	53.4%	50.4%	3.7

The 20 minute headways, especially at transfer point two, had greater potential for implementation of tactics. This was because of two reasons. Firstly there was less variability in the travel times for 20 minute headways, as bus bunching did not occur as often or to the same magnitude. Secondly the time delay for passengers that miss transfers with larger headways was higher, so more often are tactics implemented to make the transfer and improve the total passenger travel time.

Overall the validation showed that when the optimization model indicated certain operational tactics to be employed to make a direct transfer, this actually occurred 95.2% of the time. Most of the times that the tactics did not result in the direct transfer occurring, the buses only just missed each other. A second application of tactics in this case should increase the validation percentage to close to 100%.

5 Discussion

There are several limitations to the model developed, and these relate to the assumptions used in its derivation. The assumptions of the next bus being on time and the passenger demands not changing for small variations in bus arrival times are not always valid. This was a reasonable assumption for small deviations of the bus from schedule. As the deviation from schedule increases the model will be less accurate. Significant deviations could be due to heavy traffic conditions or other delays upstream which will affect subsequent buses.

As headways increase so does the likelihood that passengers have consulted a timetable prior to travelling and the effect of arrival time variations on passenger demand was reduced.

Therefore it was suggested that the deviation from the schedule be compared to the headways to analyze the validity of this assumption.

Another limitation of the model was its reliance on deterministic data. Although in a real time application some of the parameters will be known, such as the passengers on the bus, other parameters such as the downstream passenger demands and travel times may not. The model was best used with the expected values for these quantities and will be less accurate with greater variability in them.

The simulation package used was not able to process transferring passengers. The boarding and alighting rates were increased to compensate, but it was not possible to actually model the interchange of passengers between buses. If this had been included it would have affected the dwell times of buses at transfer points, however this difference would be small and was deemed inconsequential compared to other variability's in the executing of the simulation. Also affected would have been the passenger demands for on buses after missed transfers, but again this was ignored.

In practice, different routes will have different types of buses which have different passenger capacities. However, TransModeler did not allow different passenger capacities of different routes. This means that routes with buses that generally carry more passengers will have to carry less resulting in slightly biased results.

Calculating travel time was based on the simulated time for the bus to get from one point to another. The route and speed at which the buses travel at are limited to the class of the road and the geographic GIS layer. The assumption made was that it was accurate given that traffic signals and traffic of the real world are ignored

The derived model defined routes as a specified sequence of stops from the set of all stops and calculated the tactic effects of all routes on all stops, constrained if the stop was not part of the route. Changing of the notation of the problem to have each route and its stops individually defined and the tactics effects calculated thus could improve the processing efficiency of the model, for application to larger networks. It is also suggested that heuristics could be used for this type of problem in real time.

For the application to the North Shore example network the model and ILOG tool worked well. The lack of integration between the optimization and simulation programs limited the evaluation of the tactic effectiveness. However the method of evaluation used could be implemented to assess the impact of operational tactics on different types of networks.

6 Conclusion

Without the use of tactics successful transfers based on a synchronized timetable do not occur often. The further from the start of the route the transfer was, the lower the probability of the vehicles achieving a successful transfer.

Implementing tactics using deterministic optimization software and stochastic simulation software improves the synchronization of bus transfers thus increasing the number of successful transfers. With higher headways, there was larger potential to use tactics to make transfers successful to reduce the total travel time of passengers. This gives rise to the potential to apply online real-time tactics in the real world make PT more reliable and attractive.

The derived model and associated ILOG-based processor are powerful tools for implementing these operational tactics with deterministic figures and this could be used in real time with continuous and expected data.

7 Further Studies

Application to continuous data – The logical next step would be applying the model in "real time" in a simulated environment; allowing the model to instruct the network whether to apply operational tactics to correct the variability's that arise using known data for what would actually be available in practice and expectancies for other quantities such as passenger demands and travel times.

Multi-objective optimization – Other factors which could be evaluated are operator cost and bus arrival time reliability. Further exploration of the relationship between these and the total passenger travel time and the number of direct transfers achieved could be interesting.

Application to different networks – Other bus networks can be analyzed using the same optimization model. While this model involved three routes and two transfer points, more complex with more routes and transfers can be easily implemented to compare the effectiveness of the model.

Assessment of the effect on long term ridership – Holding buses and skipping stops negatively affects some passengers, while making direct transfers advantages others. A study on how these factors relate to the overall attractiveness of the PT network would be crucial before any implementation could take place.

8 Acknowledgements

The authors would like to thank Avi Ceder and Yuval Hadas for their constant guidance and supervision throughout the project and also ARTA for the bus route data.

9 References

BARTON & TSOUROU 2000. Healthy Urban Planning. New York: World Health Organization.

CEDER, A. 2007. *Public Transit Planning and Operation: Theory, Modeling and Practice,* Oxford, UK, Butterworth-Heinemann, Elsevier.

CEDER, A., GOLANY, B. & TAL, O. 2001. Creating bus timetables with maximal synchronization. *Transportation Research Part A: Policy and Practice,* 35**,** 913-928.

DOMSCHKE, W. 1989. Schedule synchronization for public transit networks. *OR Spektrum,* 11**,** 17-24.

DUEKER, K., KIMPEL, T., STRATHMAN, J. & CALLAS, S. 2004. Determinants of bus dwell time. *Journal of Public Transportation,* 7**,** 21-40.

FLEURENT, C., LESSARD, R. & SÉGUIN, L. 2007. Transit Timetable Synchronization: Evaluation and Optimization. GIRO Inc.

FRANK, L., KAVAGE, S. & LITMAN, T. 2006. Promoting Public Health through Smart Growth: Building healthier communities through transportation and land use policies and practices. Vancouver: Smart Growth British Columbia.

HADAS, Y. & CEDER, A. 2008. Public transit simulation model for optimal synchronized transfers.

JAKOB, A., CRAIG, J. L. & FISHER, G. 2006. Transport cost analysis: a case study of the total costs of private and public transport in Auckland. *Environmental Science & Policy,* 9**,** 55-66.

KNOPPERS, P. & MULLER, T. 1995. Optimized transfer opportunities in public transport. *Transportation Science,* 29**,** 101-105.

MEES, P. 2000. *A very public solution : transport in the dispersed city,* Carlton South, Vic., Melbourne University Press.

WATERSON, B. J., RAJBHANDARI, B. & HOUNSELL, N. B. 2003. Simulating the impacts of strong bus priority measures. *Journal of Transportation Engineering-ASCE,* 129**,** 642–647.

Optimisation of Small-Scale Ambulance Move-up

Lei Zhang
Department of Engineering Science
University of Auckland
New Zealand
lzha174@aucklanduni.ac.nz

Abstract

This paper proposes a multi-time scale Markov Decision Process (T-MDP) model for relocating a small fleet of ambulances to better respond to emergency calls. A set of temporally abstract states are used to approximate the intractable state space in the real world. A T-stage look-ahead scheme is developed to approximate the temporally accrued rewards and discounted probabilities for the T-MDP. An example is given to compare its performance with five other ambulance locating models.

Key words: dynamic programming, healthcare, ambulance relocation.

1 Introduction

The ambulance problem we consider is characterised by the response process summarised as follows. When an emergency call is received, a dispatcher chooses an available ambulance to dispatch. A typical dispatch policy will look at those vehicles waiting idle at a base or returning to their base and find the vehicle that is closest to the accident scene. (The time required for this dispatching process is typically small, and so will be ignored here.) This dispatched ambulance travels to the scene of the emergency call. Once the ambulance reaches the scene, the ambulance officers perform an initial at-scene treatment of the patient. If no more medical care is required then the ambulance becomes free at the scene, and returns to its base. More typically, however, transportation is required to a hospital and the ambulance then becomes free at the hospital after completing a patient hand-over.

The elapsed time between the receipt of the call and the vehicle arriving at the scene is termed the *response time*. An ambulance organisation's performance will often be measured by the percentage of calls having a response time no greater than some target time W. When trying to maximise their performance, ambulance operators typically refer to their readiness to respond to the next call in terms of *coverage*, where a suburb is considered covered if its centroid is no further than W minutes drive from the closest available ambulance. They also refer to *call coverage* which is the probability that the location of the next call is no further than W minutes drive from the closest available ambulance.

In many ambulance organisations, each ambulance is assigned a base to which the vehicle returns after each call; determining the best base for each vehicle gives us a static location problem. These problems are typically solved using integer programming (IP) models which seek a vehicle-to-base assignment that maximises some simple coverage-based model of expected system performance.

In an attempt to improve their response times, some ambulance operators operate a *redeployment* policy in which they move idle ambulances from one base to another, or even to street corners, as they seek to improve their call coverage. This vehicle movement is an example of a *move-up*. A common redeployment approach is System Status Management (SSM) which, for any given number of free vehicles n_{free}, specifies a pre-defined vehicle *configuration* $C(n_{\text{free}})$ that gives a standby location (i.e. a base or a street corner) for each of the idle vehicles; e.g. see Bryan et al. 2010. Whenever the number of free vehicles changes, the dispatchers are required to determine a set of move-ups that efficiently move vehicles into the appropriate SSM configuration.

An alternative approach, which is the focus of our work, is to adopt dynamic relocation models to determine optimal or near optimal move-ups for the available ambulances. Unlike SSM plans, these solutions do not enforce a single configuration $C(n_{\text{free}})$ for each $n_{\text{free}} = 1, 2, 3, ...$, but instead allow the target configuration to depend on the current vehicle locations. For a broad overview on ambulance location and relocation models, see Brotcorne, Laporte, and Semet 2003.

In this paper, we propose a multi-time scale Markov Decision Process (T-MDP) formulation to relocate a small fleet of ambulances. Traditional Markov Decision Processes (MDPs) assume a single fixed time step: actions take one step to complete, and their immediate consequences become available after one step. This often leads to intractable state space for real problems. Moreover, in many applications, we are more interested in common-sense, higher-level actions such as the destination for each vehicle, rather than next location each vehicle goes to in one time step. These higher-level actions are similar to Artificial Intelligence's classical macro operators (Richard 1985), in that they can take control for some period of time using a sequence of primitive actions (next location for each vehicle in one time step) until some termination condition is met, at which point a new macro-action can be applied. Macro-actions can be useful when solving MDPs with large state and action spaces by focusing on a subset of states. The key is to treat these macro-actions just like primitive actions, which have associated temporally accrued rewards and transition probabilities during execution of each macro-action. These macro-actions can be combined with primitive actions to improve rate of convergence, or used in temporally abstracted MDPs to find (near) optimal solutions to the original problems. T-MDP chooses a subset of states from the original MDP and defines a set of macro-actions between these states along with temporally accrued rewards and transition probabilities. Then the standard value iteration or policy iteration for a dynamic programme can be used to solve the problem. For related theoretical analysis and applications, see Sutton 1995 and Hauskrecht et al. 1998.

The remainder of this paper is organised as follows. In section 2, we briefly discuss multi-time scale MDPs (T-MDPs). In section 3, we present the T-MDP formulation for ambulance move-up. In section 4, we provide computational experiments. Section 5 summarises our findings.

2 Background

A finite MDP consists of four elements: A finite state space S; A finite set $A = A_1 \cup A_2 ... \cup A_s$ where A_s is a set of actions that can be performed at state s; A bonded reward function $R : S \times A \rightarrow \mathbb{R}$ such that $R(s, a)$ denotes the immediate reward associated with action a in s; A transition distribution P: $S \times A \times S \rightarrow [0, 1]$ such that $P(s, a, w)$ denotes the probability of moving to state w when action a is performed at state s on one time step. The objective is to find a *policy* that maximises the expected accumulated reward over an infinite horizon: $E(\sum_{t=0}^{\infty} \gamma^t r^t)$ where r^t is a reward obtained at time t and $\gamma \rightarrow (0, 1)$ is a discount factor. This objective can also be formulated as a Bellman optimality equation (Bellman 1957):

$$V(s) = \max_{a \in A}(R(s, a) + \gamma \sum_{w \in S} P(s, a, w)V(w))$$

where $V(s)$ is the value function at state s.

This equation represents the dynamics of the system in one time step. Sutton (1995) viewed such an action $a \in A$ as a primitive action which leads to immediate consequences and generalized the equation for multi-time scale MDPs, which summarise several time scales and have the ability to predict events that can happen at various unknown moments. In a multi-time scale MDP, an action is viewed as a macro-action which takes control of the system for some period of time using a sequence of primitive actions until some termination condition is met and a new macro action takes over. Note that primitive actions also qualify as macro-actions in the general form: they are initiated in a state, take control for a while (one time step), and then end. Let s be a state that initiates a macro action a. Let τ be the time at which some termination condition is met. The reward for this action can be written as:

$$R(s, a) = E_\tau(\sum_{t=0}^{\tau} \gamma^t r(s^t, a(s^t))|s^0 = s, a)$$

where $a(s^t)$ specifies the primitive action taken at time t (state s^t) under the macro-action a with an immediate reward $r(s^t, a(s^t))$. Similarly, the discounted transition probabilities according to the expected termination time for this macro-action can be written as:

$$\begin{aligned} P(s, a, w) &= E_\tau(\gamma^{\tau-1} Pr(s^\tau = w|s^0 = s, a)) \\ &= \sum_{t=1}^{\infty} \gamma^{t-1} Pr(\tau = w, s^t = s'|s^0 = s, a) \end{aligned}$$

Now the general form of the Bellman optimality equation can be written as

$$V(s) = \max_{a \in A(s)} (R(s, a) + \sum_{w \in S} P(s, a, w)V(w)) \quad (1)$$

where $A(s)$ is a finite set of macro-actions at state s.

3 Model Description

In this section, we define the set of temporally abstract states for the T-MDP model and show how to approximate their temporally accrued rewards and transition probabilities for move-up actions using a look-ahead scheme.

3.1 Temporally Abstract State Space

Let M represent the total number of ambulances in the EMS system. Let G represent a network consisting of a set of nodes N and a set of undirected links L, where $(i, j) \in L$ is the undirected link joining nodes i and j, $i, j \in N$. The spacing of the nodes is such that each link requires a constant drive time Δt to traverse. The travel time from node i to j along the shortest path is denoted as $d(i, j)$. Call arrivals follow a Poisson process with a total arrival rate λ. The probability that the next call occurs at node k is p_k with $\sum_{k \in N} p_k = 1$. Let B represent a set of preselected nodes as ambulance bases. Let B_i represent the ith element in B. If an ambulance is available, it should be either traveling to or waiting at a base. The dispatch policy is to send the closest ambulance. We assume transport to hospital is always needed. (In reality, an average of 75% of calls requires transport to hospitals (Maxwell et al. 2010)). The service time from being dispatched to becoming free at hospital is assumed to be an exponential distribution with rate μ.

The state space of our model consists of two parts. The first part of state space consists of the set S^c of states in which idle ambulances are waiting at bases. We restrict the capacity at each base to be one vehicle. These states are viewed as candidate stable states since they are the configurations we consider for ambulance move-up in real time. Once ambulances are in one of these states, they 'do nothing' until an event (a call arrival or a completion of service) occurs. A candidate stable state s^c is a binary vector of length $|B|$ where the ith element (s^c_i) is 1 if an ambulance is waiting at base location B_i. For example, if B is $\{5, 12, 18\}$, $M = 3$, then $(0, 0, 1)$ indicates that there is one idle ambulance at location 18 and two ambulances being busy. The set S^c is created by enumerating every subset, including an empty subset, of B. For example, if $B = \{5, 12, 18\}$, then $S^c = \{(0, 0, 0), (0, 0, 1), (0, 1, 0), (1, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 0), (1, 1, 1)\}$. The state with all zeros represents that all vehicles are busy.

The second part of state space consists of set S^t of 'temporary' states which are the consequences of an event of either a call arrival or a completion of service occurring at any candidate stable state s^c. These are the states that require a move-up action in the model. A state $s^t \in S^t$ is represented using a binary vector of length $|B|+1$ where the ith element(s^t_i) is 1 if an idle ambulance is at base location $B_i, i = 1, 2, ..., |B|$ when the event occurs, otherwise 0 and the $|B| + 1$ th element is 1 if the event that leads to this state is a completion of service, otherwise 0 if the event is a call arrival. The set S^t is created by total enumeration over S^c. As an example of one iteration, given that $B = \{5, 12, 18\}$ and $s^c = (0, 1, 1)$, we then create states including $(0, 1, 1, 1)$, $(0, 0, 1, 0)$ and $(0, 1, 0, 0)$. Next we insert them to set S^t if any of them has not been inserted by previous iterations.

3.2 Model Formulation

We aim to develop a T-MDP model in order to maximise the discounted number of calls reached within the specified target time W in an infinite horizon. Ambulance responses are given a reward of 1 if the ambulance can reach the call within W and 0 otherwise. To model an ambulance movement, we use a 'wait-and-jump' discretisation in which the ambulance waits for time Δt at its current node k, then moves instantaneously to an adjacent node k'.

First we consider candidate stable states at which the only action is to 'do nothing' until the next event occurs. The optimality equation for a candidate stable state s^c is

$$V(s^c) = R(s^c) + \gamma P(s^c)^T V, \forall s \in S^c$$

where $R(s^c)$ is the conventional one-step reward for 'do nothing' and $P(s^c)$ is the associated one-step transition distribution over $S^t \cup S^c$.

To define transition probabilities and expected rewards, we introduce a few additional notations here. Let n_b be the number of ambulances being busy at the time. Let Q denote the set of occupied locations by idle ambulances, $|Q| \leq M$. Let $y \in N$ be any node of the network. Let $\overline{N}_y(Q)$ denote the set of nodes of the network that are as close to y as to any of the nodes of the set Q, i.e,

$$\overline{N}_y(Q) = \{l \in N; d(y,l) \leq d(Q,l)\}$$

where

$$d(Q,l) = \min_{i \in Q}\{d(i,l)\}$$

Consider state s^c at which all ambulances are busy serving calls ($|Q| = 0$), new call arrivals are lost under our assumption of no queuing up in the system. One possible transition is to the temporary state at which one ambulance becomes free at hospital:

$$P\{(\overbrace{0,...,0}^{|B|},1)|(\overbrace{0,...,0}^{|B|})\} = (1 - e^{-(\lambda+n_b\mu)\Delta t})\frac{n_b\mu}{\lambda + n_b\mu} \tag{2}$$

In (2), we can see that this transition needs two conditions: An event occurs at next time step and the event is a completion of service. Similarly, we have a transition to itself:

$$P\{(\overbrace{0,...,0}^{|B|})|(\overbrace{0,...,0}^{|B|})\} = e^{-(\lambda+n_b\mu)\Delta t} + (1 - e^{-(\lambda+n_b\mu)\Delta t})\frac{\lambda}{\lambda + n_b\mu} \tag{3}$$

and the expected immediate reward is zero.

Consider now any candidate stable state s^c at which $0 < |Q| = m < M$. Without loss of generality, let us assume the first m digits of s^c are 1, i.e, $s^c_i = 1, 1 \leq i \leq m$ and $s^c_i = 0, m < i \leq M$. The transition probability from s^c to itself after Δt is

$$P\{s^c|s^c\} = e^{-(\lambda+n_b\mu)\Delta t} \tag{4}$$

Expression (4) is obvious since only if no event occurs, vehicles stay in the same state s^c after Δt. Then we have the transition probabilities due to a call arrival:

$$P\{\underbrace{(\overbrace{1,...1}^{k-1},0,\overbrace{1,..1}^{m-k},\overbrace{0,..,0}^{n_b},0)}_{|B|+1}|s^c\} = (1 - e^{-(\lambda+n_b\mu)\Delta t})\frac{\lambda}{\lambda + n_b\mu}\sum_{i\in\overline{N}_{B_k}(Q)} p_i, \tag{5}$$

$$k = 1,2,3...,m$$

In (5) we take into account the fact that three conditions are necessary for this transition. First of all, an event must occur during Δt. Second, the event is a call arrival and third, the call occurs at one of the nodes that are closer to kth idle ambulance than to any other occupied locations by idle ambulances in set Q. Similarly, we have the transition due to a completion of service:

$$P\{\overbrace{1,...,1}^{m},\overbrace{0,...,0}^{n_b},1)|s^c\} = (1 - e^{-(\lambda+n_b\mu)\Delta t})\frac{n_b\mu}{\lambda + n_b\mu} \tag{6}$$

The expected immediate reward in state s^c is

$$(1 - e^{-(\lambda + n_b \mu)\Delta t}) \frac{\lambda}{\lambda + n_b \mu} \sum_{i \in N:\ d(Q,i) \leq W} p_i \tag{7}$$

Equation (4), (5) and (7) also apply to the candidate stable state with all vehicles available ($m = M, n_b = 0$) and equation (6) is not needed as no calls are being served.

Now consider a temporary state s^t. A move-up action is required to relocate idle vehicles into a candidate stable state. Let $A(s^t)$ denote the set of all the candidate stable states for the given number of idle ambulances. For each move-up action $a \in A(s^t)$, we first solve an assignment problem to decide which ambulance is going to which base such that the total travel time is minimised. We assume an ambulance always travels along the shortest path to a base. As move-up takes time to complete, new call arrivals and completions of service can occur during move-up and new move-up actions may be performed. It is impractical to track the system status in the future for every possible scenario. We instead use a T-stage look-ahead scheme which is a decision tree for a partial enumeration of the near future using a set of termination conditions . The optimality equation for a temporary state s^t can be written as:

$$V(s^t) = \max_{a \in A(s^t)} (R(s^t, a) + \sum_{s \in S^t \cup S^c} P(s^t, a, s)^T V(s)) \tag{8}$$

where $R(s^t, a)$ is the temporally accrued reward for move-up action a and $P(s^t, a, s)$ is discounted transition probability to either a candidate stable state or a temporary state defined in our model. Note that we assume ambulances will always move into one of the states defined in our model for a move-up action, which is not true in reality. We use these states to approximate the real world.

Next we describe the T-stage look-ahead scheme which implicitly computes $R(s^t, a)$ and $P(s^t, a, s)$. Each stage takes time Δt. The dynamics of this look-ahead scheme is driven by three elements: idle vehicle move-ups, the randomness of call arrivals and completions of service in the near future. We explore the benefit of performing one more move-up action if an event occurs before vehicles reach their destinations assigned by the move-up action a during look-ahead. This second move-up considers every possible candidate stable state after gaining or losing one vehicle due to the event. Six termination conditions are used to stop look-ahead at a state $s, s \in S^c \cup S^t$.

The *first condition* is that all vehicles reach their assigned destinations without any event occurring during a move-up action. We stop look-ahead at the candidate stable state defined by the most recent move-up action. Note that the termination state may be the result of the move-up action a or the follow-up move-up action after an event occurs during look-ahead.

The *second condition* is that idle vehicles are still travelling to their assigned destinations at the last stage of look-ahead. In this case, we stop look-ahead by instantly moving vehicles to the intended candidate stable state.

The next two conditions stop look-ahead from performing the second move-up action for the first event. The *third condition* is that when the first event occurs, we compute the travel time for each idle vehicle to its current destination, if the maximum travel time is less than a threshold ΔT, we stop look-ahead by instantly moving vehicles into the temporary state defined by the current destinations. Note that if the event is a completion of service at the hospital, we assume the current destination for this new idle vehicle is the hospital location.

The *forth condition* is that when the first event occurs, there is no more than three stages left to look ahead, we stop look-ahead by instantly moving vehicles into the temporary state defined by the current destinations.

The *fifth condition* is that if two events have occurred during the T-stage look-ahead, we stop look-ahead by instantly moving vehicles into the temporary state defined by the current destinations.

For termination conditions 2-5, the instant movement means an implicit assumption that vehicles will always reach the target state with probability one. In reality, more events can occur before vehicles reach the target state. However, we treat this state as the most likely scenario. To compensate for this assumption, we introduce Algorithm 1 to heuristically reduce this 'idealistic' probability to γ'. A few extra notations are needed for Algorithm 1. Let the states be numbered as 0,1,...,K from the moment that the specific termination condition for the instant movement is met to reaching the target state. Let $p'(k), k = 0, 1, ...K - 1$ be the probability of not reaching the next call on time at state k. We compute γ' by considering two factors: (1) The call arrival rate λ. The $\lambda^{\max}$ is a large number to scale down λ to be within (0,1).(2) The coverage for next call at each state k. In general, the higher the call arrival rate, the smaller γ' gets and the smaller the coverage for next call at each state k, the smaller γ' gets as well.

We do not specify the expressions for transition probabilities and immediate rewards at each time step in the look-ahead as they are similar to those defined for the candidate stable states.

Algorithm 1 A heuristic approach to computing γ'.

1 Initialisation: $\gamma^{\max} \leftarrow 0.99999, \lambda^{\max} \leftarrow 50,\ \theta \leftarrow 0.0125, \gamma' \leftarrow 1,\ k \leftarrow 0$

2 While $k < K$

  Compute $p'(k)$

  $\gamma' \leftarrow \gamma'(\gamma^{\max} - \theta\frac{\lambda}{\lambda^{\max}}p'(k))$

  $k \leftarrow k + 1$

3 Return γ'

At this stage we have all the information to apply the value iteration to find the optimal value functions $V(s), \forall s \in S^c \cup S^t$. In reality, idle vehicles can be at any locations when a move-up action is required. We use the T-stage look-ahead with these value functions when termination conditions are met to evaluate every possible move-up action given the current status of vehicles and choose the best one.

4 Computational Experiments

An example of 50 nodes on a line with a single hospital is used for computational experiments. The spacing between two adjacent nodes is two minutes. As shown in Figure 1, the probability of next call occurring at each node i is randomly generated. The hospital is at node 21. Set B contains 6 preselected ambulance bases: node 6, 14, 20, 23, 34 and 43. The call arrival rate is 2.2 calls per hour. The service rate is 1.1 calls per hour. The response target time W is 8 minutes. There are five vehicles in the system. The look-ahead horizon is 10 stages (20 minutes). The threshold ΔT

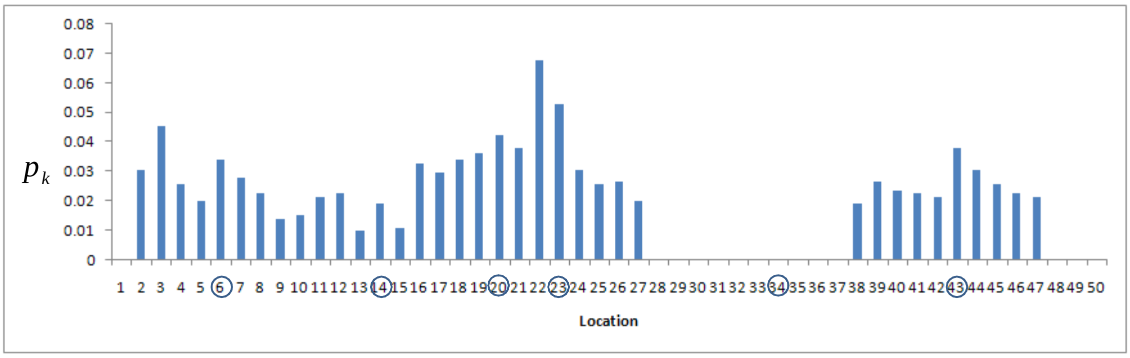


Figure 1: Plot of p_k, the probability of next call occurring at node k on a 50-node line, 6 bases marked by circles and one hospital at node 21.

Model	Description
(1) Return-to-base (RS)	Return to predetermined bases
(2) SSM	Move to a predetermined configuration
(3) Next-call (NC)	Maximise the probability of reaching the next call on time
(4) Look-ahead Next-call (LNC)	Modify Next-call by adding a look-ahead scheme
(5) Instant move-up (IM)	Instantly move into a predetermined configuration

Table 1: Five other models for locating ambulances

for the instant movement is 6 minutes. At most one vehicle can be located at each base.

30 data sets are randomly generated using the inputs we provided as above. Each data set contains two weeks of calls. We also use the same data sets to test five other models for locating ambulances. These five models are shown in Table 1. The first model is a 'return-to-base' policy. Each vehicle is assigned to a fixed base. Whenever it is free, it returns to its assigned base. Since we have five vehicles and six bases (at most one vehicle per base), we have five 'return-to-base' policies. We simulate all of them and choose the best one for comparison. The second model is a SSM approach. The predefined configurations for SSM are shown in Table 2. These configurations are selected such that the coverage for next call is maximised . The third model is a 'next-call' model. This model aims to maximise the probability of reaching the next call on time by moving idle vehicles into a candidate stable state. The same set of candidate stable states defined in our model is used. For each move-up action, an assignment problem for deciding the destinations for vehicles are solved to minimise total travel time and then the probability of getting to the next call on time can be computed by tracking each vehicle's position. The fourth model is a 'look-ahead next-call' model. We add a similar look-ahead scheme to the next-call model. In this look-ahead, we consider the possibility of one vehicle becoming free before the next call occurs in which case a follow-up move-up is performed. The last model is an 'instant move-up' model which is a modified version of SSM. Whenever the number of idle vehicles changes, we instantly move idle vehicles into the configuration defined by SSM. We view this 'unrealistic' model as an optimistic performance measure.

The results on the percentage of calls reached on time by these six models are summarised in Table 3. In the last row of Table 3, we show the average percent deviation from the best solution value. The best solution value is found by using instant move-up model as an optimistic performance measure. The CPU time to

Number of idle vehicles	Configuration
1	20
2	6,20
3	60,20,43
4	5,14,23,43
5	5,14,23,34,43

Table 2: Configurations for SSM

Data set	RS	SSM	NC	LNC	T-MDP	IM
1	59.61%	63.59%	65.09%	64.77%	67.02%	75.62%
2	58.43%	64.23%	65.09%	66.49%	66.60%	75.73%
3	59.83%	65.31%	65.84%	67.88%	68.53%	78.30%
4	58.22%	65.31%	66.49%	67.78%	67.35%	76.05%
5	60.04%	65.52%	66.38%	67.56%	69.07%	77.66%
6	59.83%	65.74%	67.67%	70.14%	67.67%	76.58%
7	58.97%	66.06%	64.55%	66.38%	67.78%	73.25%
8	61.33%	66.38%	66.27%	67.56%	70.14%	76.48%
9	59.61%	66.38%	67.13%	67.99%	67.78%	75.73%
10	58.43%	66.60%	66.49%	66.49%	66.70%	76.26%
11	60.04%	67.13%	66.06%	66.70%	66.70%	78.09%
12	59.29%	67.35%	68.64%	70.03%	69.07%	78.52%
13	63.16%	67.45%	70.14%	69.17%	70.25%	78.84%
14	61.22%	67.67%	69.07%	71.43%	72.29%	79.16%
15	61.33%	67.67%	69.60%	68.96%	69.71%	78.95%
16	61.65%	67.99%	68.10%	68.10%	69.39%	78.73%
17	59.72%	68.64%	70.78%	70.57%	70.57%	79.38%
18	60.58%	68.96%	68.31%	70.03%	71.00%	79.05%
19	59.40%	68.96%	69.17%	68.64%	69.28%	80.13%
20	58.22%	69.07%	69.82%	70.68%	71.32%	76.05%
21	63.59%	69.28%	70.78%	71.54%	70.68%	81.74%
22	60.15%	69.39%	67.78%	68.53%	71.32%	78.73%
23	62.62%	69.60%	69.92%	70.35%	71.00%	81.74%
24	63.16%	69.82%	67.88%	71.86%	71.21%	78.95%
25	62.30%	69.82%	70.25%	69.50%	70.14%	74.76%
26	61.33%	70.03%	72.72%	73.68%	73.04%	82.38%
27	62.62%	70.57%	71.11%	72.72%	72.07%	81.95%
28	64.12%	70.68%	71.97%	72.18%	70.46%	81.42%
29	65.20%	70.89%	70.78%	72.29%	74.33%	83.14%
30	63.05%	71.11%	71.75%	72.07%	73.90%	78.30%
Average	17.48%	10.48%	9.87%	8.99%	8.51%	0

Table 3: Proportion of calls reached on time for six ambulance locating models. The last row shows the average deviation from the solutions found by IM.

solve the T-MDP took 40 minutes using value iteration. The CPU time to find a move-up decision using the look-ahead with value functions found by T-MDP varied between 0.2 and 3 seconds. Calls were assumed lost in the simulations if no vehicle was idle for all these models. We observed that about 3% of total calls were lost. In practice, lost calls are typically responded by a backup system.

As Table 3 shows, the RS gave the worst performance for all 30 data sets. Without considering the IM, the T-MDP outperformed other four models in 18 data sets. The LNC gave the best performance in 9 data sets. The NC gave the best result in 2 data sets and SSM gave the best result in 1 data set. On average, the T-MDP gave the best performance with an average deviation of 8.51%. The LNC is 8.99% worse than the IM. The average devations for the NC and SSM were 9.87% and 10.48%. It seems that all the models that do not fix an ambualnce to a base (return-to-base) provide significant improvements on the number of calls covered. This is a good example that dynamic vehicle relocation can improve EMS performance.

5 Concluding Remarks

In this paper we developed a T-MDP model for generating dynamic vehicle deployments in order to better respond emergency calls. Five other ambulance locating models were compared with our model using randomly generated data. The experimental results showed that our model can generate good move-up policies. This approach can solve problems with a small fleet of ambulances (five to six) and a small set of bases within reasonable time. The values of parameters in Algorithm 1 are determined by experiments. It leaves a possibility of finding better values with further research. We are also conducting research on solving large-scale problems.

6 Acknowledgments

I wish to thank Andrew Mason and Andy Philpott for their supervision and untiring help. This work was supported by the New Zealand Institute of Mathematics and its Applications.

References

Bellman, R. 1957. *Dynamic Programming.* Princeton University Press.

Brotcorne, L., G. Laporte, and F. Semet. 2003. “Ambulance location and relocation models.” *Eupropean Journal of Operational Research* 147(3):451–463.

Bryan, E., DO. Bledsoe, FACEP, and EMT-P. 2010. EMS Myth 7: System Status Management lowers response times and enhances patient care.

Hauskrecht, M., P.L. Kaelbling, T. Dean, and C. Boutillier. 1998. “Hierarchical solution of Markov decision processes using macro-actions.” *Proceedings of Uncertainty in Artifical Intelligence.*

Maxwell, M. S., M. Restrepo, S. G. Henderson, and H. Topaloglu. 2010. “Approximate Dynamic Programming for Ambulance Redeployment.” *Informs Jurnal on Computing* 22 (2): 266–281.

Richard, E.K. 1985. *Learning to Solve Problems by Searching for Macro-Operators.* Pitman Publishing Ltd, London.

Sutton, R.S. 1995. “TD Models: Modeling the World at a Mixture of Time Scales.” *Proceedings of the 12th Int. Conf. on Machine Learning*, pp. 531–539.

Scale Advantage - Using Data Envelopment Analysis to Detect Economies of Scale in the Insurance Industry

Kane Harton
University of Auckland Alumni
New Zealand
kharton@gmail.com

Abstract

Insurer A is the largest insurer in its product category within New Zealand (Investment Savings and Insurance Association, June 2010), this led to the assumption that Insurer A has scale advantage. This assumption has an impact on Insurer A's business decisions so this assumption was examined further. An optimisation technique called Data Envelopment Analysis (DEA) was used to determine efficiency in the insurance market and therefore scale advantage.

It was found that Insurer A did not have scale advantage in the adviser market but did have scale advantage in the institutional market. It was also found that Insurer A is spending efficiency on commission. Potential reasons for these results were explored and their impact on business decisions was discussed.

Key words: Data Envelopment Analysis, Efficiency, Insurance, Scale, Scale Advantage, Economies of Scale.

1 The Problem

For the purposes of thir report I will refer to research I did within my own organisation, which is refered to as Insurer A due to commercial sensitivity reasons. Insurer A is the largest insurer in its product category within New Zealand (Investment Savings and Insurance Association, June 2010). As a result of its size the employees of Insurer A assumed this size advantage led to economies of scale, in effect scale advantage (Staff of Insurer A, 2010). This scale advantage would mean that Insurer A is the most efficient insurer and therefore could extract a higher return on capital then any other insurer. This scale advantage premise leads to numerous conclusions including that the company should seek to grow in order to increase efficiency further. Therefore understanding whether Insurer A does in fact have scale advantage is very important.

This leads to the following question, 'does Insurer A have scale advantage?'. This question effectively translates to; 'Is Insurer A more efficient then its competitors because of its size?'. In effect we are trying to prove the hypothesis that because Insurer A is larger then its competitors, Insurer A is more efficient.

Efficiency as defined in our question is a relative measure. To determine if Insurer A is efficient we need to determine how efficient A's competitors are. The method we chose to use to examine this problem was Data Envelopment Analysis (DEA), which is an operations research technique that explores efficiency.

1.1 What is Efficiency?

Efficiency is commonly defined as inputs / outputs (A. Emrouznejad, 1995 –2001). Where we seek to maximise our outputs from a given set of inputs. In the case of the insurance industry we would like to maximise our profit, or similarly we would like to minimise our costs.

2 Introduction to Data Envelopment Analysis (DEA)

DEA is an optimisation technique used to determine the efficient frontier of a group of decision making units, in this case insurance companies. DEA determines the efficiency of each insurer. This efficiency is relative to the performance of the insurers it is being compared to. If an insurer is considered efficient then it is at the highest level of efficiency of those being studied. More than one insurer can be considered efficient.

DEA requires that a set of inputs and outputs be defined, these inputs and outputs are then used to determine efficiency. The premise is that inputs are used to create outputs, and thus define efficiency. The insurer producing the most outputs with the least inputs is deemed to be efficient. (T. Coelli, P. Rao & G. Battese, 2005)

A simple implementation of DEA would involve solving the below mathematical formulation. This programme seeks to maximise each insurer's efficiency by altering the weight assigned to the inputs and outputs as shown below.

$$\textbf{Efficiency of Insurer}_{\mathbf{i}} = maximise \frac{\sum_{j=1}^{m} Outputs_{ji} Weight_i}{\sum_{k=1}^{n} Inputs_{ki} Weight_i}$$

If a particular insurer outputs a superior amount of output X, X will be weighted highly for this insurer, this means each insurer will achieve the highest efficiency possible based on that insurers strengths. An equivalent way to view the problem (the dual or envelopment form of the problem) is to see whether a more efficient insurer can be created by combining the outputs and inputs of other insurers. If this can be done then the insurer is not efficient.

The major advantage of DEA is that it quantifies the efficiency of insurers and produce targets for inefficient companies.

3 Insurance

For the purposes of this paper an insurance company is a company who 'manufactures' insurance policies. Insurance protects against risk, in exchange for a regular premium an insurer agrees to pay its customer an agreed amount under certain circumstances or risks. Insurance works because insured risks are based on historical information and are statistically small. The insurer is therefore able to calculate the expected claims and price insurance accordingly.

3.1 New Business

It is often said that 'insurance is sold and not bought'. Insurance is not tangible and is considered to be a luxury good. As a result of insurance is not prioritised by consumers and sales takes a very important and expensive role in insurance.

At the point of sale there is a mismatch between cashflows arising from an insurance policy. For an insurer a policy will have high upfront sale costs, but it will result in ongoing cashflows over the policy's lifetime. In cashflow accounting an instantaneous loss would be made when a new policy is sold because the value of its future cashflow is not considered. For this reason Actuaries are able to defer these upfront expenses and spread them over the expected life of a policy, avoiding an instantaneous loss and allowing profit to be recognised in the first year the policy is taken out. The profitability of an insurance policy is dependent on actual experience, in particular the variance from claims and policy longevity assumptions.

3.2 Distribution Channels

The cost of New Business varies according to how insurance is sold, there are three main distribution channels.

3.2.1 Adviser

This is where an adviser will meet with clients to discuss their needs and then sell them insurance. An adviser may be independent which means that they are not tied to any insurance manufacturer and are typically reimbursed for their efforts on a commission basis. An adviser may also be tied to an insurance company, this means the adviser will be paid a salary (although is also likely to be paid some commission as well) and the insurer will have to cover their costs of doing business.

In recent years the entrance of insurance aggregators, who bargain for commission on behalf of a group of advisers, has seen commission rise from ~100% of first year premium income to ~200% of first year premium (Staff of Insurer A, 2010). Despite the adviser channel being high cost, 70% of New Business is sold through this channel making it valuabe for insurers to maintain **(**Investment Savings and Insurance Association, June 2010).

3.2.2 Institutional

This is where the insurance product is distributed by a corporate partner. One example of this is through a bank – this is significantly cheaper than distributing through Advisers. 30% of New Business is generated through the Institutional channel **(**Investment Savings and Insurance Association, June 2010**)**.

3.2.3 Direct

This is where insurance is sold by the manufacturer online or via a call center, creating the lowest distribution costs. As a result of this a lesser level of advise is offered through the direct channel.

This is an emerging market with 0.05% of business distributed via this method **(**Investment Savings and Insurance Association, June 2010**)**.

3.2.4 Existing Business

Existing Business (as opposed to New Business) is business that the insurer already has on its books and is therefore relatively cheap to maintain. The longer a given policy remains with an insurer the more profitable it is.

4 Previous Efficiency Research Within Insurer A

Efficieny had been explored twice by consultants engaged by Insurer A since 2004. Both of these approaches will be examined below.

4.1 Consultant 1

Consultant 1 created a cost model for the industry based on Insurer A's cost structure. Existing Business and New Business were chosen as cost drivers, this means the model effectively assumes scale advantage. This model was applied to Insurer A's competitors and found that Insurer A had scale advantage. Consultant used this assumption of scale advantage to prove scale advantage – creating a logical fallacy. When the modelled costs were compared to the actual costs it was found that the actual costs were in fact 22% lower than the modelled costs and with a standard deviation of 48% and check. These substantial differences were written off by the consultant as reporting differences. (Consultant 1, 2004)

4.2 Consultant 2

Consultant 2 also touched on scale advantage in a larger piece of work. They did this by comparing the cost of acquiring New Business with the amount of New Business actually acquired. As well as this they compared the cost of maintaining Existing Business with the amount of Existing Business held. They then used these results to draw scale curves, which showed an interaction between scale and efficiency.

Unlike Consultant 2, Consultant 1 had avoided this approach as it is reliant on insurers reporting the breakdown between New Business and Existing Business costs accurately and in a consistent manner with other insurers. Separating these costs is more of an art then a science as it requires separating all fixed and direct costs. The way an insurer does this will depend on the system they use to split costs.

Other failings of this technique include that it is not able to quantify efficiency. There also appeared to be no underlying scientific reason for drawing the scale curves the way they did.

Consultant 2's message was that Insurer A did in fact have scale advantage however their findings did not seem to support this. (Consultant 2, 2004)

4.3 Proposed Technique

Both techniques have their failings, Consultant 1's example fails because it avoids using actual competitor data and Consultant 2's example fails because it relies too strongly on the accuracy of competitor data. Data Envelopment Analysis (DEA) will be used in this study to determine scale advantage.

5 Data envelopment analysis applied

5.1 Previous DEA Research

A large amount of research on the insurance industry has already been published. Typically this research seeks to compare the efficiency of insurers in one country against another or seeks to compare the efficiency of different insurance industries (M. Eling & M. Luhnen, July 2008). None of the research I identified is focussed on trying to determine scale advantage. This does not mean however this research is not useful as they are trying to determine the efficiency of each insurer; they just use this efficiency information in a different way.

The selection of both inputs and outputs are critical to determining efficiency, it is necessary to track the processes that add value to insurers. The majority of frontier efficiency research uses a value added approach to select outputs (M. Eling & M. Luhnen, July 2008). Value is added in insurance via three functions; risk-pooling, financial services related to insured losses and intermediation (F. Fiordelisi & O. Ricci, March 2010). Input and output choice reflects how value is added.

5.2 Input & Output Choice

This value-added technique requires a number of inputs and outputs, which presented a couple of problems for this research:

- Not all of this information was available for insurers in New Zealand via our limited data sources.
- Having many inputs and outputs creates too many degrees of freedom for a small market such as NZ. When an approximate value added method was trialled for New Zealand insurers it was found that it did not produce meaningful results.

As a result I approached the problem from a different angle; being primarily interested in the advantages that scale offers. It is likely that most of the benefits from scale are derived from policy administration. We are interested in how administration expenses are transformed to premiums. The more efficient an insurer is the more premium they can write for less cost. As discussed previously the premium can be New Business or Existing Business, which will affect how much it costs.

By making a number of assumptions this approach remains consistent with the value-added approach.

- Insurers are being run in a solvent manner. This is a fair assumption due to consistent regulation and professional actuarial standards and is evidenced by the fact that no NZ based insurer has failed recently. Therefore it can be assumed that a consistent proportion of premiums will be used to pay claims. There is also unlikely to be a scale advantage in this instance as underwriting standards will be defined by reinsurance companies who will work with all NZ based insurers.
- Investment returns should be the same across all insurers, as all insurers should pursue low risk investment strategies for risk management purposes. There should be no scale advantage here as all insurance companies should possess sufficient scale to make institutional investments.

By making these assumptions we are able to take a very simple approach to this model:

Input: Operating Expenses

Outputs: Change in Existing Business premiums (as a proxy for New Business)
Existing Business Premium

Both the input and outputs are derived from each insurer's Annual Financial Report and should be largely consistent between insurers.

5.3 Choice of DEA Methodology

There are many different variations of DEA, but the two most basic models are the CCR model developed by Charnes, Cooper and Rhodes in 1978 and the technique developed by Banker, Charnes and Cooper in 1984 (BCC) (T. Coelli, P. Rao & G. Battese, 2005). The key difference between these different formulations is how they handle returns to scale, the CCR is the more simplistic model and assumes return to scale are constant while BCC assumes variable returns to scale.

The focus of this research is scale, so the choice of model is crucial. Allowing variable returns to scale effectively means when defining efficiency of a given insurer it is only compared to other insurers of a similar size. Assuming scale advantage exists this is equivalent to compensating smaller organisations for their reduced efficiency.

Assuming that economies of scale exists within a BCC model a small organisation will appear efficient but within a CCR model the small organisation will appear inefficient.

Indices
o = insurer: 1,…,l
i = insurer: 1,…,l
j = outputs: 1,…,m
k = inputs: 1,…,n
Parameters
θ_o = efficiency of insurer o
x_{ik} = input k for insurer i
q_{ij} = output j for insurer i
Decision Variables
λ_{io} = weight of insurer i while finding the efficiency of insurer o
Model: CCR, Input Oriented, Envelopment Form
minimise θ_o

(1) $-q_{oj} + \sum_{i=1}^{l} q_{ij}\lambda_{io} \geq 0$ for o = 1,…,l

(2) $\theta_o x_{ok} - \sum_{i=1}^{l} x_{ik}\lambda_{io} \geq 0$ for j = 1,…,m

$\lambda_{io} \geq 0$ for k = 1,…,n

Explanation

The objective is to solve the dual of the efficiency maximisation problem, by finding the most efficient combination of insurers, for each insurer by selecting the weighting of the other insurers appropriately.

I chose the input oriented, envelopment form of the CCR technique, as shown above, to explore the effects of scale; the model should not compensate smaller companies for their lack of scale. Under the CCR model assuming economies of scale do exist, given A's scale advantage, Insurer A should be the only efficient insurer and there should be correlation between an insurer's size and its efficiency. If this trend does not exist then we should be able to conclude that there is no scale advantage in the market.

5.4 Distribution Channels

As mentioned previously there are 3 different distribution channels, the direct channel is currently emerging and at this point in time there are not enough competitors to perform an analysis, which leaves us with the Adviser and Institutional channels. These channels have different underlying cost structures and profitability so should be considered. By doing so each insurer, apart from A, operates within a single channel.

As discussed earlier, commission is paid to Advisers operating within the Adviser channel, this is a major source of cost for insurers. Each insurer will pay a different level of commission as part of their competitive offering. Commission costs are separated out, this means for the Adviser channel we have a view of efficiency including and excluding commissions. This allows us to explore the effect of commission on efficiency. We have defined the phrase "internal efficiency" to mean we are excluding commission and "total efficiency" to mean we are including commission.

5.5 Software Implementation

I used DEA.py, an implementation of DEA within Python (a programming language) using PuLP (a LP modeller written in Python) and Coin-OR (a solver accessible in Python) to solve the problem.

6 Results

6.1 Adviser

From Figure 1, we can see that Insurer A's Adviser operations are considered to be efficient by the model. However we can also see two smaller insurers G and H are considered efficient, this indicates that scale doesn't exist. The remaining insurers appear to be fairly inefficient when only internal efficiency is considered.

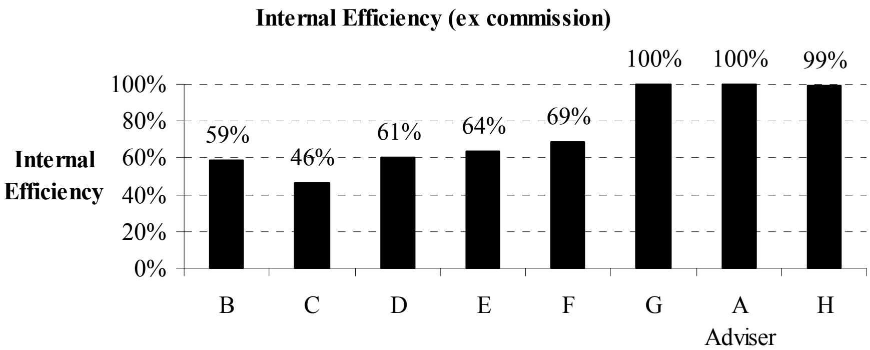


Figure 1: Internal efficiency of insurers distributing through advisers

Figure 2 shows that when commission costs are included the same insurers are considered efficient, however the efficiency of the remaining insurers improve.

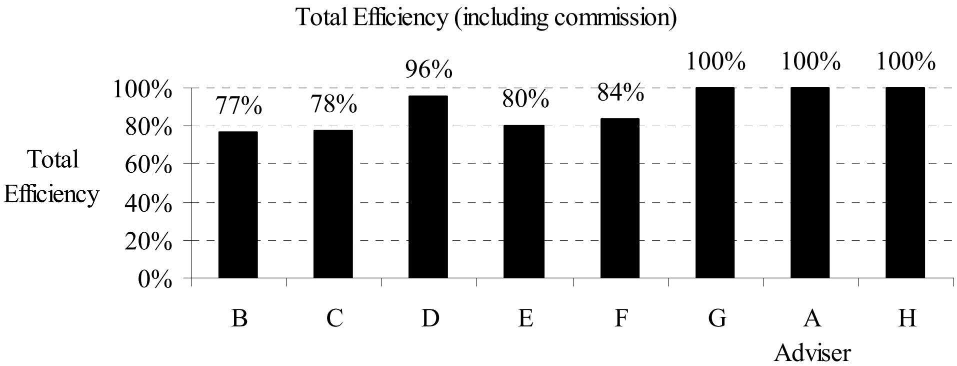


Figure 2: Total efficiency of insureres distributing through advisers

Figure 3 indicates that scale advantage doesn't exist, this is confirmed when we correlate the size of each insurer with their efficiency. A linear scale trend explained 0.0016% of the data.

An advantage of using such a simple DEA model (single input, 2 outputs) is that we can visualise our results, by normalising our outputs with respect to the input. The efficient frontier is defined in figure 4 by the grey lines. The further away an insurer is from the efficient frontier, the more inefficient it is considered.

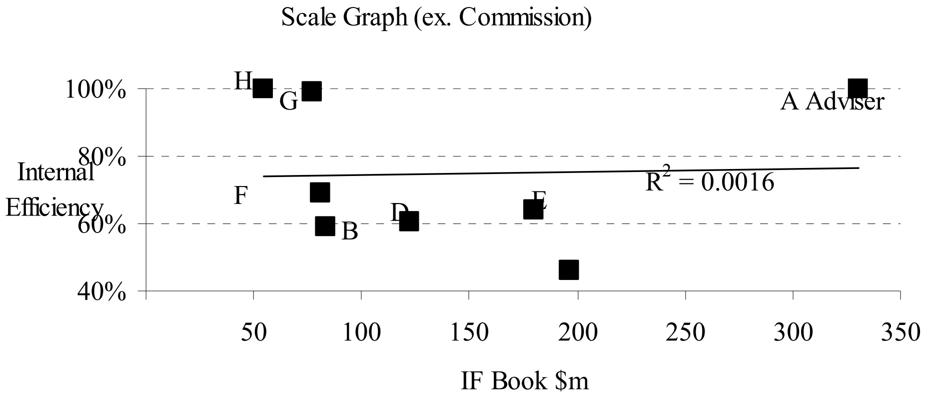


Figure 3: Scale graph for adviser channel

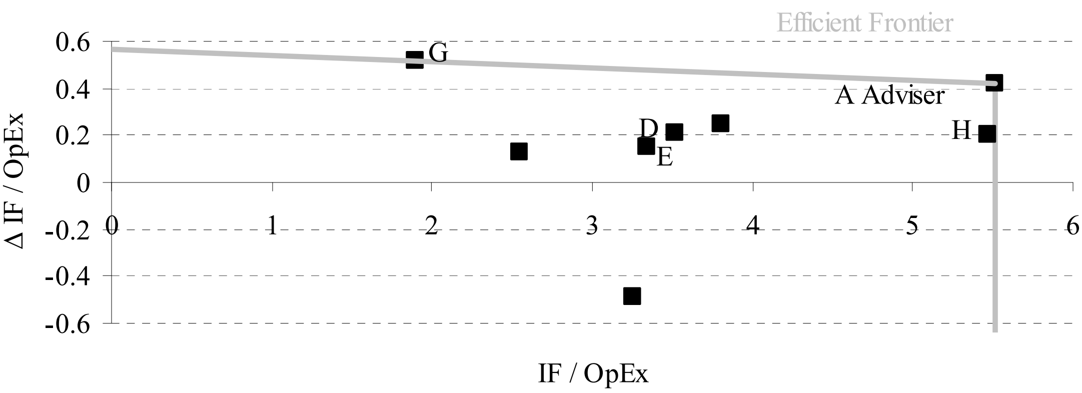


Figure 4: Internal efficiency frontier for adviser channel

What we can see from figure 4 and figure 5 is that when we include commission both Insurer A and insurer H move closer to the group of other insurers. If we were to consider super efficiency, where we allow efficiency to increase beyond 100%, we would see that Insurer A has lost super-efficiency when commission is considered. Therefore Insurer A is spending efficiency on commission.

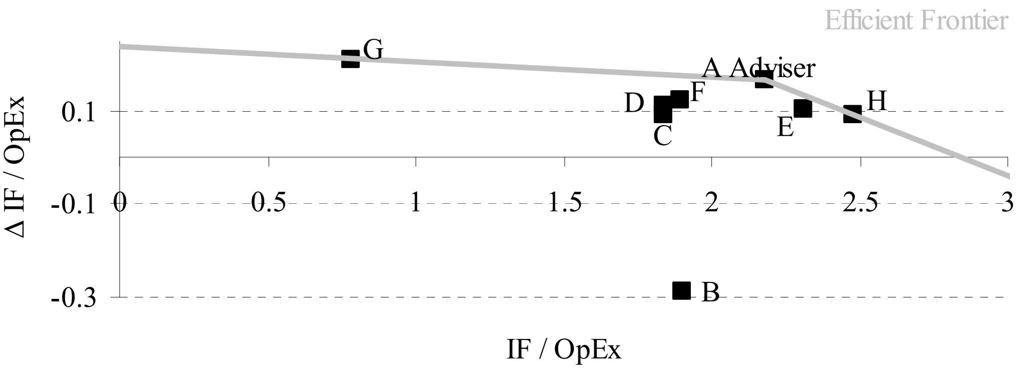


Figure 5: Total efficiency frontier for adviser channel

6.2 Institutional

Figure 6 shows that both large institutional insurers L and the institutional operations of Insurer A are considered efficient.

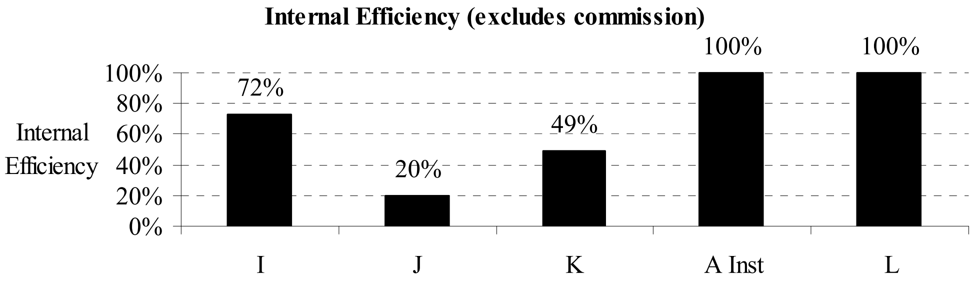


Figure 6: Internal efficiency for institutional channel

In figure 7 when we plot the size of the insurer against efficiency we find that a linear scale trend explains 65% of the data.

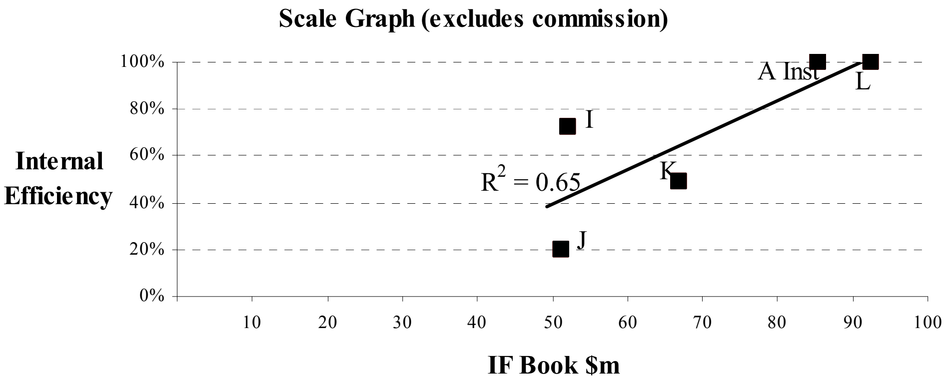


Figure 7: Scale graph for institutional channel

7 Conclusions

We have found that in the Adviser market there does not seem to be any scale advantage but there does appear to be scale advantage in the Institutional market.

Possible reasons for this difference are outlined below. The Adviser relationship is transactional as Advisers are not required to give all of their output to a single supplier, most Adviser's will choose their supplier each time they sell insurance. It may be that Adviser's value 'independence' and therefore to receive a greater share of an Adviser's business you have to offer them more than a competitor with a smaller share of their business. Or it may be a case of Insurer A not making the most of its potential scale advantage. Insurer A has grown over the years through a number of acquisitions and as a result has a number of inefficient Legacy computer systems it needs to maintain which could lead to this inefficiency.

The institutional market on the other hand is a relatively new market, most institutional business would have been written on efficient modern systems. The market that institutional is tapping into is different as well, once an insurer has signed up an institutional partner. The only insurer selling insurance to their partner's customer base is the insurer. The insurer has a monopoly on their partner's customer base. Therefore there is no need to pay an additional premium for a large share of those customers. The result of these conclusions are;

- That Insurer A should seek to grow their institutional business by acquiring more institutional partners.
- Insurer A should not pursue a strategy of acquisitions in the Adviser channel in order to increase efficiency further.
- Insurer A should focus on increasing internal efficiency, such as resolving issues with Legacy systems.

It was also found within the Adviser channel that Insurer A is spending efficiency on commission, effectively Insurer A is paying Adviser's their efficiency instead of taking it as profit. Insurer A should review their commission structure and see if they can reduce this without reducing market share.

8 Further work

8.1 Super-efficiency

We could implement super efficiency within DEA/pu, in order to quantify the efficiency being spent on Advisers.

8.2 Malmquist Productivity Index

By implementing a Malmquist Productivity Index we can track how efficiency changes over time and track the movements of technology and A's own efficiency. This would track A's improvements in efficiency as a result of an efficiency focus within the organisation arising from this study.

8.3 Exploring the Impact of Legacy Costs

We are in the process of calculating how efficient Insurer A would be if we were to remove the costs that Legacy systems impose on the organisation. These costs are being derived from business cases to remove these Legacy systems. This will allow us to explore whether by tackling these internal inefficiencies Insurer A will be able to capture some scale advantage.

9 References

A. Emrouznejad, 1995-2001, " Ali Emrouznejad's DEA HomePage", *Warwick Business School*

Consultant 1, 2004, Consultant 1's report to Insurer A

Consultant 2, 2009, Consultant 2's report to Insurer A

F. Fiordelisi & O. Ricci, March 2010, "Efficiency in the Life Insurance Industry: What are the Efficiency Gains from Bancassurance?", *Economics & Management of Financial Intermediation (EMFI) Working Papers*

Investment Savings and Insurance Association, June 2010, "Insurance Industry Statistics"

M. Eling & M. Luhnen, July 2008, "Frontier Efficiency Methodologies to Measure Performance in the Insurance Industry: Overview and New Empirical Evidence", *Working Papers on Risk Management and Insurance No 56*

Staff of Insurer A, 2010, "personal communication"

T. Coelli, P. Rao & G. Battese, 2005, "An introduction to efficiency and productivity analysis", *Springer Publishers*, 2nd Edition

Rima: Building Math Models for Reuse

Geoff Leyland
Incremental Limited
geoff.leyland@incremental.co.nz

Abstract

Rima is a yet-another tool for formulating mathematical models. Rima's goal is to make it easy to write reusable models. To this end, it supports composing a model from parts, and makes it easy create generic parts. Models are defined symbolically, problem data is late bound, and models can be highly structured. Rima is implemented in Lua, binds to CLP, CBC and lpsolve, and is awaiting review to become part of COIN.

Keywords: model reuse, submodels, modelling languages, model abstraction

1 Introduction

1.1 Model Reuse

In this paper *reuse* means reusing a model means taking an existing model and reusing it with minimal modifications in a context for which it was not originally intended. Most math modelling languages offer some separation of model equations and data, meaning the same model can be reused with different data, but this is a weak case of reuse, and, frankly, even facilities for separation of equations and data in many languages are weak[1].

In this paper we will focus on a single case of reuse: a model for a single knapsack model is available, and we wish to reuse it in a model that consists of multiple knapsacks. We've chosen this example because a knapsack model is simple enough that the model should not complicate the message of this paper, and because is not possible to reuse a knapsack in this manner in any other modelling language without modifying the original model.

1.2 Reusing a Knapsack

The "traditional" view of a knapsack is that a burglar is running around a house with a bag to carry his or her loot, and wishes to steal items of as much total value as possible, without overfilling the sack.

The knapsack model is presented in pseudocode below:

[1]Most languages' example code puts the model *after* the data

```
maximize(sum(i in ITEMS) take(i) * value(i))
sum(i in ITEMS) take(i) * size(i) <= CAPACITY
forall(i in ITEMS) take(i) is_binary
```

Now we extend our "burglar" model so that there are several burglars acting together, each with a sack. They'd like to maximise the value of all the sacks, but they can't overfill any sacks. In pseudocode this model is:

```
maximize(
  sum(s in SACKS, i in ITEMS) take(s, i) * value(i))
forall(s in SACKS)
  sum(i in ITEMS) take(s, i) * size(i) <= CAPACITY(s)
forall(s in SACKS, i in ITEMS) take(s, i) is_binary
forall(i in ITEMS) sum(s in SACKS) take(s, i) <= 1
```

The first three lines of the multiple-knapsack model have the same purpose as the three lines of the knapsack model, only we've had to modify every line to add a sacks index (highlighted in italics). The fourth line is new information - an extra constraint that says that just because you have several sacks now, you can't steal the TV twice - making this "multiple knapsack" a generalised assignment problem.

We can't avoid adding the fourth line to the model, as it adds extra information. This paper concerns avoiding the changes to the first three lines.

1.3 What's the big deal?

The changes to the knapsack model only amount to a few characters, so what's the big deal? It's not that hard to edit the model quickly, right?

Firstly, we might be working on a more complex model than a knapsack, and the changes might amount to a few more characters. But the main problem is that editing the model violates two fundamental tools we have to build an understanding of the world.

1.3.1 Abstraction

Abstraction can be viewed as the separation of *implementation* and *interface*, and is a fundamental tool for understanding our world. In order to build a car, it is necessary to have detailed understanding of areas such as internal combustion engines, steering systems, suspension, and brakes[2]. In order to drive a car, though, you only need to be able to turn a wheel, push two or three pedals, and have a rudimentary understanding of the road rules[3]. This separation of implementation and interface is what allows us to drive cars without a Ph.D. in thermodynamics[4].

In order to reuse the knapsack model in a multiple knapsack, we had to edit it, which required us to understand its implementation, violating abstraction, and making reuse unnecessarily complicated.

[2] Understanding brakes recently appears to have become optional for some vehicle manufacturers

[3] Very rudimentary if you live in Auckland

[4] I don't deny that it might be a better world if only people who could accurately describe an Otto cycle were allowed to drive

1.3.2 Standing on the Shoulders of Giants

Human knowledge progresses through the sharing of knowledge, and by allowing the whole population to come up with improvements and act as a market for ideas and improvements[5].

If one person writes a knapsack model, and another takes that model and modifies it as shown to work as a multiple knapsack, then sharing improvements becomes much harder. If one or other author finds and fixes a bug, or makes an improvement to the model, that improvement is hard to share, because it's hard to tell which differences are due to the changes that were made because of the differences in the model context, and which are the actual improvement.

Changing the model makes sharing ideas hard, and hinders human progress[6].

In this paper, we will build a single knapsack model, and then reuse it in a multiple knapsack *without making any changes to the original knapsack*. In doing so we will develop a system for building math models by composing reusable parts.

1.4 Introducing Rima

> Why program by hand in five days what you can spend five years of your life automating?
>
> - Terence Parr, author of numerous compiler tools

Faced with the problems described above, I developed Rima, a new modelling tool that focuses on making it easy to construct and re-use models. Rima:

- is MIT licensed and available at `http://rima.googlecode.com/`
- is implemented in Lua: `http://www.lua.org/`
- currently binds to CLP, CBC and lpsolve
- has been submitted to COIN for review

1.4.1 Lua

There has been some confusion about the use of Lua in Rima, so it is worthwhile to give the language a brief introduction.

According to lua.org, "Lua is a powerful, fast, lightweight, embeddable scripting language".

For powerful, Lua is a full-featured scripting language with garbage collection, proper closures, proper coroutines, tail-call optimisation and no global interpreter lock.

As far as fast goes, the default implementation of Lua is about 4 times faster than the default implementation of Python on the same problem. The fast, just-in-time compiled version is about 10 times faster than the fastest implementation in Python, and has performance impressively close to C.

[5]Thanks Nicholas Taleb

[6]In the case of the knapsack, we might not mind hindering burglars

For lightweight, the windows version of Lua is about 140kb, and Lua has been ported to systems with as little as 64kb of RAM.

Lua is very easy to bind to, and is embedded a large number games, such as World of Warcraft and Civilisation 5, is in commercial software such as Adobe Lightroom and in software tools like Nmap, and Apache.

However, this paper is *not* about Lua, it is about methods building reusable math models that could work in any language, and for these purposes, Lua is no more than an implementation detail[7].

2 Symbolic Expressions

2.1 Expressions

A mathematical model is built from a set of mathematical expressions - the expression for the objective, and the expressions making up the constraints. In Rima, the objective and constraints are stored as symbolic expressions. This means the expressions are independent of any data, and particularly the dimensions of any sets, and provides strong separation of the model and the model data.

Dedicated modelling languages such as AMPL and GAMS do the same, so we are not introducing anything new. However, in most "language bindings" to modelling systems, such as PuLP and FlopC++, the equations you build are directly manipulating matrix rows.

2.1.1 Constructing Expressions

Rima expressions involve *references*, which are placeholders for variables whose values we'll look up later. References are constructed with `rima.R`, as illustrated below:

```
e = rima.R("a") * rima.R("x") + rima.R("b") * rima.R("y")
```

Expressions can be printed, and you can see below that the expression has been stored in symbolic form (`--` introduces a comment in lua, and `-->` is a convention we'll use in this paper for showing output):

```
print(e)                          --> a*x + b*y
```

All the `rima.R`'s become cumbersome, so `rima.define` provides a shortcut by letting us define references in advance:

```
rima.define("a, x, b, y")
e = a * x + b * y
print(e)                          --> a*x + b*y
```

Expressions don't just have to be constructed from references, they can be about other expressions too:

```
print(3 * e)                      --> 3*(a*x + b*y)
print(e^2)                        --> (a*x + b*y)^2
```

[7] A very carefully chosen implementation detail!

In all these cases, `e` encapsulates a symbolic representation of the expression, providing one component of a clear separation of expressions and data.

2.1.2 Evaluating Expressions

Combining an expression with data makes it concrete (unless some references are undefined, see below) so it can be evaluated. `rima.E` evaluates expressions by matching references to a *table* of values:

```
rima.define("a, x, b, y")
e = a * x + b * y
print(rima.E(e, {a=2,x=3,b=4,y=5}))--> 26
```

If some references are undefined, `rima.E` returns a new expression involving the undefined references:

```
print(rima.E(e, {a=2,b=4}))        --> 2*x + 4*y
```

The values you provide as data to `rima.E` are not restricted to being immediate data, they can be other expressions:

```
rima.define("xpos, xneg")
print(rima.E(e, {x=xpos - xneg})   --> a*(xpos - xneg) + b*y
```

2.2 A Simple LP

Expression construction is just enough to allow us to build a very simple optimisation model.

First, we declare some references:

```
rima.define("a, b, x, y")
```

Then we create a new modelling environment with `rima.mp.new`. In this case, `rima.mp.new` takes a single argument which is a table of key-value pairs defining parts of the model:

```
M = rima.mp.new({
```

First, we define the objective and its sense (Rima also understands "maximize" and "MaXImiSE"):

```
  sense = "maximise",
  objective = a*x + b*y,
```

Then we define a couple of constraints with `rima.mp.C`. Note that the constraints are named. The constraint construction syntax is a little awkward (the constraint comparison operators are strings) because we have to keep the Lua parser happy:

```
  C1 = rima.mp.C(x + 2*y, "<=", 3),
  C2 = rima.mp.C(2*x + y, "<=", 3),
```

Finally, we define some bounds on the references that will become our LP variables.

```
  x = rima.positive(),
  y = rima.positive()
})
```

Rima makes no distinction between "parameters" and "variables" in the way other modelling languages do, but when the time comes to solve, the solver needs to know the bounds of the variables it will solve for.

M, the value returned from rima.mp.C encapsulates a complete symbolic representation of our little model.

As with expressions, M can be printed:

```
print(M)
--> Maximise:
-->   a*x + b*y
--> Subject to:
-->   C1: x + 2*y <= 3
-->   C2: 2*x + y <= 3
-->   0 <= x <= inf, x real
-->   0 <= y <= inf, x real
```

The output is very useful for documentation and debugging.

Of course, we'd like to solve M even more than print it. rima.mp.solve takes the model and a table of data and solves, returning tables of primal and dual variables:

```
primal, dual = rima.mp.solve("clp", M, {a=2, b=2})
```

The primal and dual tables are structured in the same way as the input data and model. Because the constraints are named it's easy to access their values and duals:

```
print(primal.objective)          --> 4
print(primal.x)                  --> 1
print(primal.y)                  --> 1
print(primal.C1)                 --> 3

print(dual.x)                    --> 0
print(dual.C1)                   --> 0.333
```

3 Structured Data

Rima data can be richly structured. Like all other languages we're aware of (modelling and general-purpose), Rima supports arrays. Rima also supports structures. Structures are missing from a number of popular modelling languages, and are poorly supported in others. Likewise, though the languages bindings are written in support structures, the modelling systems themselves are not fully integrated with structures.

3.1 Arrays, Sums and Array Assignment

If you define a reference, and then treat it like an array, Rima will hope that the data you match to the reference is, in fact, an array[8]:

[8] If you give Rima the wrong type of data, it'll try to help you work out what's wrong

```
rima.define("X")
e = X[1] + X[2] + X[3]
print(e)                          --> X[1] + X[2] + X[3]
print(rima.E(e, {X={1,2,3}}))     --> 6
```

`rima.sum` sums an expression over a set. Here, Rima runs through each element of `X`, assigning the current element of `X` to `x` in each iteration of the sum:

```
rima.define("x, X")
e = rima.sum{x=X}(x^2)
print(rima.E(e, {X={1,2,3}}))     --> 14
```

You can assign to a whole array at once, much like the `foreach` syntax you might find in other languages. Here, the `i`th element of `X` is set to 2^i. As with constraints, the syntax is a little awkward: we need to keep the Lua parser happy:

```
rima.define("i, X")
t = { [X[i]] = 2^i }
print(rima.E(X[5], t))            --> 32
```

3.2 Structures

Like arrays, if you treat a reference like a structure by acting as if it has fields to access, Rima will comply:

```
rima.define("item")
mass = item.volume * item.density
print(mass)
  --> item.volume * item.density
print(rima.E(mass, {item={volume=10, density=1.032}}))
  --> 10.32
```

Although this seems almost trivial, it's a very uncommon feature in math modelling languages, and it's one of the key features that allows us to address submodels.

3.3 A Structured Knapsack

We now have enough tools to revisit our knapsack model, but this time we'll build the model with structured data.

First, we define some references:

```
rima.declare("i, items")          -- items in knapsack
rima.declare("capacity")
```

Then we construct the new model. Note that here we construct the model with no fields, and then add the fields to it, in contrast to the earlier example where the entire model was specified in the argument to `rima.mp.new` :

```
knapsack = rima.mp.new()
```

Next, we define the objective. Note than we refer to the items as if `take` and `value` are fields of an item, rather than as `take[i]` and `values[i]`:

```
knapsack.sense = "maximise"
knapsack.objective = rima.sum{i=items}(i.take * i.value)
```

Then we declare the capacity constraint, again, using `i` as if it's a structure:

```
knapsack.capacity_limit = rima.mp.C(
  rima.sum{i=items}(i.take * i.size), "<=", capacity)
```

Finally, we set all the `take` variables to binaries, using the array assignment syntax:

```
knapsack.items[{i=items}].take = rima.binary()
```

Remember this model, because now we've written our single knapsack, we won't change it at all.

As before, Rima can print the model:

```
print(rima.repr(knapsack, {format="latex"}))
```

In LaTeX, if we ask nicely:

$$\begin{aligned} \textbf{maximise} \quad & \sum_{i \in \text{items}} i_{\text{take}} i_{\text{value}} \\ \textbf{subject to} \quad & \\ \text{capacity\_limit} : \quad & \sum_{i \in \text{items}} i_{\text{size}} i_{\text{take}} \leq \text{capacity} \\ & \text{items}_{i,\text{take}} \in \{0, 1\} \forall i \in \text{items} \end{aligned}$$

To solve the model we first define the items in the sack. Note that each item is a little table[9]:

```
ITEMS = {
  camera   = { value =  15, size =  2 },
  necklace = { value = 100, size = 20 },
  vase     = { value =  15, size = 20 },
  picture  = { value =  15, size = 30 },
  tv       = { value =  15, size = 40 },
  video    = { value =  15, size = 30 },
  chest    = { value =  15, size = 60 },
  brick    = { value =   1, size = 10 }}
```

Then we solve the model with CBC, passing it the set of items and the capacity of the sack as data:

```
primal = rima.mp.solve("cbc", knapsack,
  {items=ITEMS, capacity=102})
```

We manage to steal 160 worth of stuff, including the camera and the vase, but not the brick:

```
print(primal.objective)           --> 160
print(primal.items.camera.take)   --> 1
print(primal.items.vase.take)     --> 1
print(primal.items.brick.take)    --> 0
```

[9]or dictionary, record, struct or object depending on your upbringing

4 Reusable Model Components

Now that we've built our knapsack model, we'll reuse it in a number of contexts without modifying it at all. We will

- Configure the knapsack in a stronger manner than is currently possible
- Extend (or make a subclass of) the knapsack
- Include the knapsack multiple times in another model

4.1 Customising the Knapsack

Suppose, for example, the burglars have trouble taking both the camera and the vase. In Rima, constraints, like expressions, are just data, and so, as we saw earlier, we can just include the extra constraint in the model data we pass to `rima.mp.solve`:

```
primal = rima.mp.solve("cbc", knapsack,
  {items=ITEMS, capacity=102,
  camera_xor_vase =
    rima.mp.C(items.camera.take + items.vase.take, "<=", 1)})
```

Now the burglars don't take the vase (at a cost of 14):

```
print(primal.objective)             --> 146
print(primal.items.camera.take)     --> 1
print(primal.items.vase.take)       --> 0
```

4.2 Extending the Knapsack

What if the burglars keep having the same problem with vases and cameras? They don't want to keep specifying the extra constraint on the command-line. Instead, they'd rather have a new model that included the extra constraint.

For this purpose, `rima.mp.new` can take two arguments, the model you want to extend and any extensions to the model:

```
side_constrained_knapsack = rima.mp.new(knapsack, {
  camera_xor_vase =
    rima.mp.C(items.camera.take + items.vase.take, "<=", 1)})
```

And you can solve the new model exactly as we solved the original knapsack:

```
primal = rima.mp.solve("cbc", side_constrained_knapsack,
  {items=ITEMS, capacity=102})

print(primal.objective)             --> 146
```

`side_constrained_knapsack` is a model object exactly like the original knapsack - it encapsulates a symbolic model of side-constrained knapsack. We've reused a model in a slightly different context without modifying the original model at all, so we're nearly done.

4.3 Multiple Sacks

Finally, we are ready to try a multiple sack model. Initially, we don't consider that they're knapsacks, and just write a model for burglars with more than one sack. The burglars wish to maximise the value of all sacks, but still can't steal the TV more than once just because they have more than one sack:

```
rima.define("s, sacks")

multiple_sack = rima.mp.new({
  sense = "maximise",
  objective = rima.sum{s=sacks}(s.objective),
  only_take_once[{i=items}] =
    rima.mp.C(rima.sum{s=sacks}(s.items[i].take), "<=", 1)
})
```

Note that we haven't said anything about what the sacks actually are - that'll come later. Also note that we're treating `sacks` and `s` as if they're structures - we're referencing `sacks[s].objective` and `sacks[s].items[i].take`. The ability to name and address submodels is derived from Rima's support for structures, and is a key part of Rima's ability to handle submodels so comprehensively.

We don't need to specify what the sack submodel is until we actually solve the problem (though we could specify it earlier). Here, we solve a multiple knapsack model - the highlighted line showing where we specify the submodel:

```
primal = rima.mp.solve("cbc", multiple_sack, {
  items = ITEMS,
  [sacks[s].items] = items,
  sacks = {{capacity=51}, {capacity=51}},
  [sacks[s]] = knapsack})

print(primal.objective)              --> 146
```

Sack 1 contains the `camera`, `vase` and `brick` while sack 2 contains the `necklace` and `video`.

Now we've achieved our goal - we've used our single knapsack model in a multiple sack model without changing the original model at all.

4.4 Multiple Side-Constrained Knapsacks

As a last trick, remember that the burglars can't carry the camera and vase in the same sack. It's easy to model this: we just change the definition of `sacks` when we solve:

```
primal = rima.mp.solve("cbc", multiple_sack, {
  items = ITEMS,
  [sacks[s].items] = items,
  sacks = {{capacity=51}, {capacity=51}},
  [sacks[s]] = side_constrained_knapsack})

print(primal.objective)              --> 146
```

Sack 1 contains the `camera`, `picture` and `brick` while sack 2 contains the `necklace` and `vase`.

Here we've taken a knapsack model, and without modifying it, we've extended it to be a side-constrained knapsack, and then used it in a multiple sack model. We've also taken a multiple sack model that was intended to be used with a knapsack and, without modifying it, used it with a side-constrained knapsack.

So not only have we achieved model reuse, we've gained a very flexible system for "mix and match" modelling, where we can build models from parts - almost like snapping lego blocks together.

5 Conclusion

To summarise, we wrote a single-knapsack model and reused it without any modification in a side-constrained knapsack, in a multiple knapsack, and a multiple side-constrained knapsack. This was possible because Rima models are symbolic and structured.

Models that are reusable without modification don't violate abstraction, and assist the sharing of modelling knowledge.

The techniques shown clearly work well for simple models such as a knapsack, and we hope to show in the future that the same techniques can work for more complex models.

6 Thanks

- Stu Mitchell, maintainer of PuLP, for clearing the way
- Phil Bishop at the New Zealand Electricity Authority for providing motivation
- Everyone who took the time to review this presentation
- The global financial crisis for giving me plenty of free time last year!

OpenSolver: Open Source Optimisation for Excel

www.opensolver.org

Andrew J Mason
Department of Engineering Science
University of Auckland
New Zealand
a.mason@auckland.ac.nz

Iain Dunning
Department of Engineering Science
University of Auckland
New Zealand

Abstract

We have developed an open source Excel add-in, known as OpenSolver, that allows linear and integer programming models developed in Excel to be solved using the COIN-OR CBC solver. This paper describes OpenSolver's development and features.

Key words: OpenSolver, Open Source, Excel, Solver, Add-in, COIN-OR, CBC.

1 Introduction

Microsoft Excel (Wikipedia, 2010) contains a built-in optimisation tool known as Solver (Frontline, 2010). Solver is developed by Frontline Systems, who provide the software to Microsoft. Using Solver, a user can develop a spreadsheet optimisation model and then solve it to find an optimal solution. Many introductory optimisation courses use Solver and Excel to introduce students to modelling and optimisation. Unfortunately, when students apply their optimisation skills to real-world problems, they often discover that the size of problem Solver can optimise is artificially limited to no more than 200 decision variables (Solver Limits, 2010), and that they need to upgrade Solver to one of Frontline's more expensive products, such as Premium Solver. (Beta versions of the latest Excel version, Excel 2010, have a smaller variable limit (Dunning 2010); the current variable limit in shipping versions of Excel 2010 has not been confirmed.)

We have recently been involved in the development of a large staff scheduling model. This model was developed as part of a consultancy exercise where it was important that the model could be used and scrutinised by a range of different users. Satisfying this requirement using commercial products would have been logistically difficult and expensive.

For a number of years, a group known as COIN-OR (Computational Infrastructure for OR) (COIN-OR, 2010) have been developing and promoting open source optimisation software, including the linear/integer programming optimiser known as

CBC (CBC, 2010). While typically not as fast as its commercial equivalents, the COIN-OR software has gained a reputation for quality and reliability and is now widely used in industrial applications.

The combination of Excel and COIN-OR is a compelling one, and appeared to be a perfect solution for our large staff scheduling model. Unfortunately, we were unable to find any software that allowed Excel spreadsheet models to be solved using any of the COIN-OR optimisers. A decision was taken to develop such software to progress the scheduling project. This decision eventually led to the development of OpenSolver, a freely available Excel add-in that allows existing spreadsheet linear and integer programming models to be solved using the COIN-OR CBC optimiser

2 Solver Operation

We start by briefly describing how optimisation models are built using Solver. Readers familiar with Solver may wish to skip this section.

A typical linear programming model is shown in Figure 1. The decision variables are given in cells C2:E2. Cell G4 defines the objective function using a 'sumproduct' formula defined in terms of the decision variables and the objective coefficients in cells C4:E4. Each of the constraints are defined in terms of a constraint left hand side (LHS) in G6:G9 and right hand side (RHS) in I6:I9. Note that the 'min' in B4 and the constraint relations in H6:H9 are for display purposes only, and are not used by Solver.

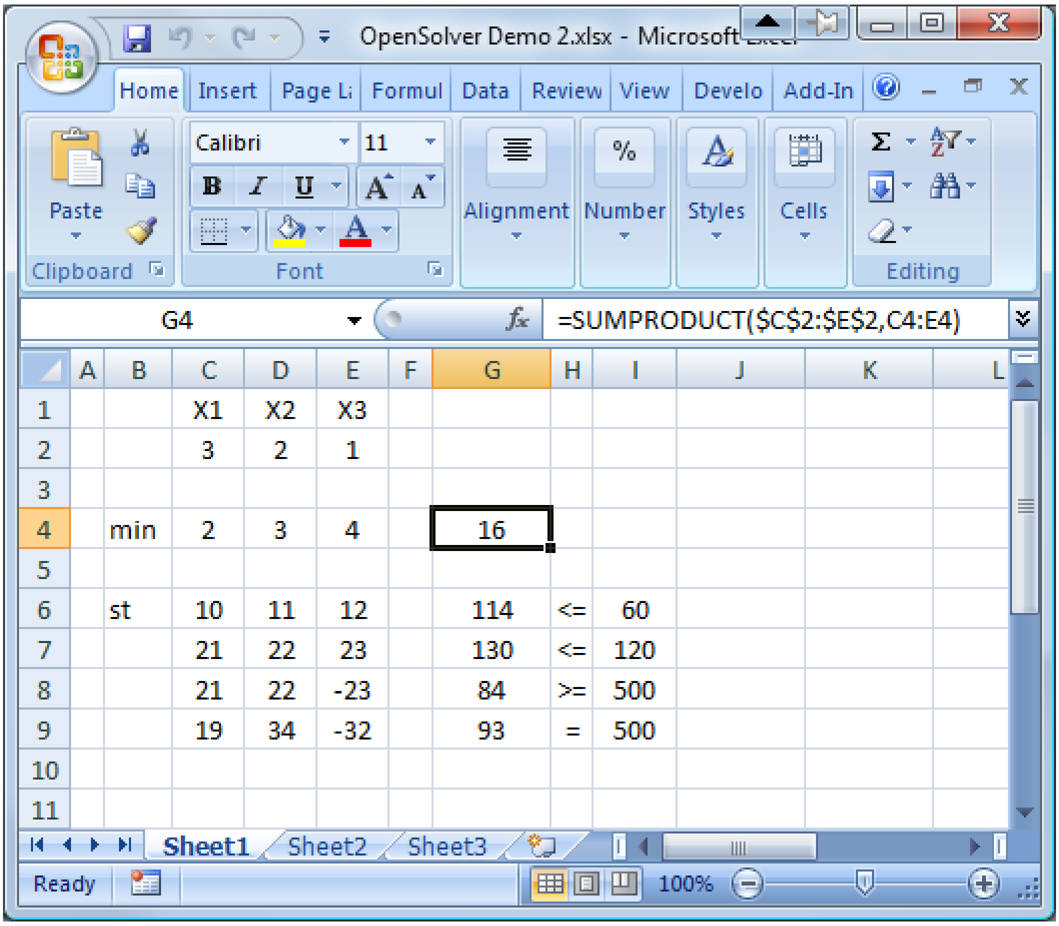


Figure 1: A typical spreadsheet optimization model. Many models do not follow this layout, but instead 'hide' the model inside the spreadsheet formulae.

Figure 2 shows how this model is set up using Solver. The decision variables, objective function cell and constraint left and right hand side cells must all be specified by the user within Solver. Note that a constraint can either consist of two multi-cell ranges of the same dimension, such as A4:A8 >= B4:B8, or a multi-cell range and a constant or single-cell range such as A4:A8 <= 8. Integer and binary restrictions on the variables can also be specified using constraints.

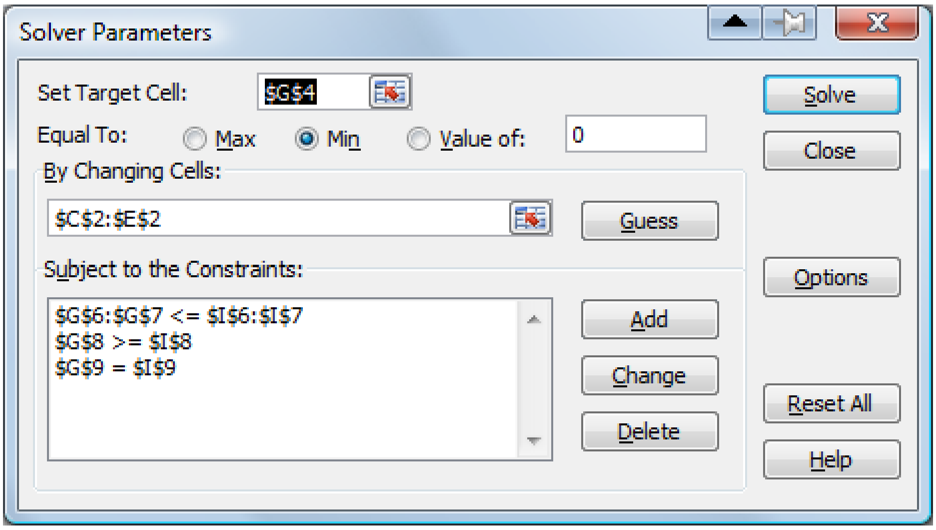


Figure 2: Solver setup for the model above.

The means by which Solver stores a model does not appear to be documented. However, a Google search eventually uncovered a comment that Solver uses hidden named ranges (see, e.g., Walkenbach 2010). The Excel Name Manager add-in (Pieterse, 2010), developed by Jan Karel Pieterse of Decision Models UK, was then used to reveal this hidden data. The names used by Solver are documented in Appendix 1. OpenSolver uses these named ranges to determine the decision variables, the objective function and LHS and RHS ranges of each constraint.

3 Constructing a Model from Excel Ranges

We wish to analyse the spreadsheet data to build an optimisation model which has equations of the form:

$$
\begin{array}{lll}
\text{Min/max} & c_1x_1+c_2x_2+\ldots & c_nx_n \\
\text{Subject to} & a_{11}x_1+a_{12}x_2+\ldots & a_{1n}x_n \le/=/\ge b_1 \\
 & a_{21}x_1+a_{22}x_2+\ldots & a_{2n}x_n \le/=/\ge b_2 \\
 & \ldots & \\
 & a_{m1}x_1+a_{m2}x_2+\ldots & a_{mn}x_n \le/=/\ge b_m \\
 & x_1, x_2, \ldots, x_n >= 0 &
\end{array}
$$

Assuming the model is linear, then the Excel data can be thought of as defining an objective function given by

$$\text{Obj}(\mathbf{x}) = c_0+c_1x_1+c_2x_2+\ldots c_nx_n$$

where $\mathbf{x}=(x_1, x_2, \ldots, x_n)$ are the decision variable values, $\text{Obj}(\mathbf{x})$ is the objective function cell value, and c_0 is a constant. Similarly, each constraint equation *i* is defined in Excel by

$$LHS_i(\mathbf{x}) \le/=/\ge RHS_i(\mathbf{x}) \Leftrightarrow LHS_i(\mathbf{x}) - RHS_i(\mathbf{x}) \le/=/\ge 0, \ \ i=1, 2, ..., m$$

where $LHS_i(\mathbf{x})$ and $RHS_i(\mathbf{x})$ are the cell values for the LHS and RHS of constraint *i* respectively given decision variable values $\mathbf{x}$. Because Solver allows both $LHS_i(\mathbf{x})$ and $RHS_i(\mathbf{x})$ to be expressions, we assume that both of these are linear functions of the decision variables. Thus, we have

$$a_{i1}x_1+a_{i2}x_2+...a_{in}x_n- b_i = LHS_i(\mathbf{x}) - RHS_i(\mathbf{x}), \ \ i=1, 2, ..., m.$$

OpenSolver determines the coefficients for the objective function and constraints through numerical differentiation. First, all the decision variables are set to zero, $\mathbf{x}=\mathbf{x}^0=(0,0,...,0)$ giving:

$$c_0=Obj(\mathbf{x}^0)$$
$$b_i = RHS_i(\mathbf{x}^0) - LHS_i(\mathbf{x}^0), \ \ i=1, 2, ..., m$$

Then, each variable x_j is set to 1 in turn, giving decision variables values $\mathbf{x}=\mathbf{x}^j$, where $\mathbf{x}^j=(x_1^j, x_2^j, ..., x_n^j)$ is a unit vector with the single non-zero element $x_j^j=1$. Then, the spreadsheet is recalculated, and the change in the objective and LHS and RHS of each constraint recorded. This allows us to calculate the following coefficients:

$$c_j = Obj(\mathbf{x}^j)-c_0$$
$$a_{ij} = LHS(\mathbf{x}^j)-RHS(\mathbf{x}^j)+b_i, \ \ i=1, 2, ..., m$$

This process assumes that these equations are indeed all linear. OpenSolver does not currently check this, but will do so in a future release.

The speed of this process is set by the speed at which Excel can re-calculate the spreadsheet. As an example, the large staff scheduling model discussed earlier has 532 variables and 1909 constraints, and took 22s to build (and just 1 second to solve) on an Intel Core-2 Duo 2.66GHz laptop. There are a number of different Excel calculation options available, such as just recalculating individual cells or recalculating the full worksheet or workbook. OpenSolver currently does a full workbook recalculation as the other options did not appear to give any faster results.

In our model form above, we have specified that the decision variables are all non-negative. This assumption is also made by the COIN-OR CBC solver we use. To ensure this is the case, OpenSolver requires the user to set Solver's "Assume Non-Negative" option. OpenSolver cannot currently solve models for which this is not the case. OpenSolver also requires that Solver's "Assume Linear Model" is set.

4 Integration with COIN-OR CBC Solver

OpenSolver uses the COIN-OR CBC linear/integer programming solver to generate its solutions. While this solver can be called directly as a DLL, OpenSolver adopts the simpler process of writing a '.lp' file that contains the model definition and then calling the command line version of CBC with this file as an input argument. CBC then produces an output file which is read back by OpenSolver. If a solution has been found, OpenSolver uses this to set the values of the decision variable cells. Any infeasibility or unboundedness, or early termination caused by time limits, is reported using a dialog.

Solver provides the user with a number of solution options. Of these, the "Max Time" option is passed to CBC (using the CBC parameter "-seconds"), and the "Tolerance" option is passed (using "-ratioGap") to set the percentage tolerance required for CBC integer solutions. Other CBC options can be set by creating a two column range named “OpenSolver_CBCParameters”on the spreadsheet that contains CBC parameter names and values.

If the “Show Iteration Results” option is set, OpenSolver displays the CBC command line window while CBC is running. (OpenSolver does not provide the step-by-step functionality that this option enables in Solver.)

5 Excel Integration and User Interface

OpenSolver is coded in Visual Basic for Applications and runs as an Excel add-in. The add-in presents the user with new OpenSolver controls using the standard ribbon interface. These OpenSolver buttons and menus are shown in Figure 3 and Figure 4.

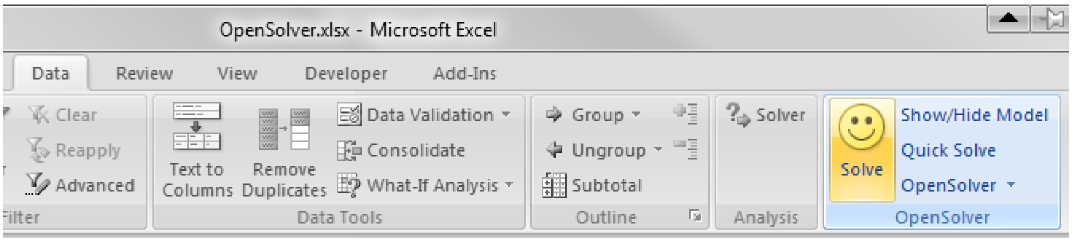


Figure 3: OpenSolver’s buttons and menu appear in Excel's Data ribbon.

Figure 4: OpenSolver’s menu gives access to its more advanced options.

OpenSolver is downloaded as a single .zip file which when expanded gives a folder containing the CBC files and the OpenSolver.xlam add-in. Double clicking OpenSolver.xlam loads OpenSolver and adds the new OpenSolver buttons and menu to Excel. OpenSolver then remains available until Excel is quit. If required, the OpenSolver and CBC files can be copied to the appropriate Office folder to ensure OpenSolver is available every time Excel is launched. Note that no installation program needs to be run to install either OpenSolver or CBC.

6 Performance

We have found OpenSolver’s performance to be similar or better than Solver’s. CBC appears to be a more modern optimizer than Solver’s, and so gives much improved performance on some difficult problems. For example, large knapsack problems which take hours with Solver are solved instantly using OpenSolver, thanks to the newer techniques such as problem strengthening and preprocessing used by CBC.

7 Model Visualisation

To view an optimisation model developed using the built-in Solver, the user needs to check both the equations on the spreadsheet and the model formulation as entered into Solver. This separation between the equations and the model form makes checking and debugging difficult. OpenSolver provides a novel solution to this in the form of direct model visualisation on the spreadsheet. As Figure 5 shows, OpenSolver can annotate a spreadsheet to display a model as follows:

- The objective cell is highlighted and labelled min or max
- The adjustable cells are shaded. Binary and integer decision variable cells are labelled ‘b’ and ‘i’ respectively.
- Each constraint is highlighted, and its sense (≥,≤, or =) shown (using a ‘>’, ‘<’ or ‘=’ respectively).

We have found this model visualisation to be very useful for checking large models, and believe it will be very useful debugging tool for students learning to use Solver.

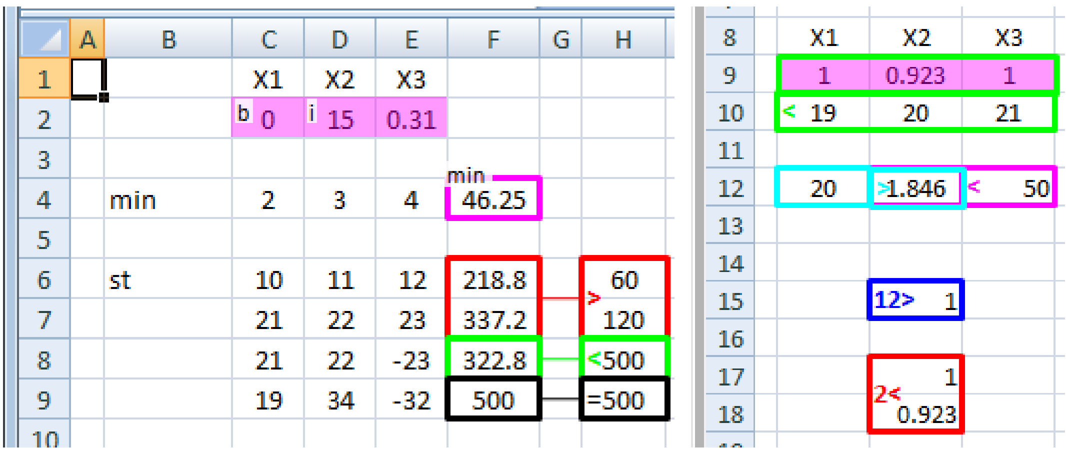


Figure 5: OpenSolver can display an optimisation model directly on the spreadsheet. The screenshot on the left shows OpenSolver’s highlighting for the model given earlier, while the screenshot on the right illustrates OpenSolver’s highlighting for several other common model representations.

8 Automatic Model Construction

The original design of OpenSolver envisaged the continued use of Solver to build (but not solve) the optimisation models. However, we have developed additional functionality that allows OpenSolver to build Solver-compatible models itself without requiring any user-interaction with Solver. Our approach builds on the philosophy that the model should be fully documented on the spreadsheet. Thus, we require that the spreadsheet identifies the objective sense (using the keyword ‘min’ or ‘max’ or variants of these), and gives the sense (>=,<=, or =) of each constraint. Our example problem shown in Figure 1 satisfies these requirements. In our teaching, we have always recommended this model layout as good practice, and note that many textbooks follow a similar approach. Our keyword-based approach creates only a minimal additional burden for the user but, by allowing the model construction process to be automated, delivers what we consider to be a significantly improved modelling experience.

To identify the model, OpenSolver starts by searching for a cell containing the text `min' or 'max' (or variants on these). It then searches the cells in the vicinity of this min/max cell to find a cell containing a formula (giving preference to any cell containing a 'sumproduct' formula); if one is found, this is assumed to define the objective function. The user is then asked to confirm that this is the objective function cell, or manually set or change the objective function cell.

After identifying the objective function cell, OpenSolver then tries to locate the decision variables. Because the objective function depends on the decision variable cells, these decision cells must be contained in the set of the objective function's 'precedent' cells (i.e. the cells referred to by the objective function formula, either directly or indirectly through other intermediate cells). However, if a sumproduct is used, for example, these precedents will also contain the objective function coefficients. OpenSolver attempts to distinguish between decision variables and objective function coefficients by looking at the number of dependent cells. An objective function coefficient will typically only have one dependent cell, being the objective function cell. However, decision variables will have many dependents as they feature in both the objective function cell and in the constraint cells. Thus, OpenSolver examines each objective function precedent cell, and keeps as decision variables those cells with more than 1 successor. The resultant set of decision variables is presented to the user to confirm or change.

The next step is to identify any binary or integer restrictions on the decision variables. These are assumed to be indicated in the spreadsheet by the text 'binary' or 'integer' (and variants of these) entered in the cell beneath any restricted decision variable.

The final requirement is to determine the constraints. OpenSolver considers all cells (except for the objective function cell) that are successors of the decision variables, and searches in the vicinity of each of these for one of the symbols '<=', '<', '=', '>=', or '>'. A constraint is created for each occurrence of these symbols.

As shown in , OpenSolver allows the user to correct mistakes made when determining the objective function cell, the decision variable cells, and/or any binary or integer restrictions on these cells. Errors in the constraints currently need to be corrected using Solver. Once the model has been confirmed (and/or corrected) by the user, it is constructed and stored using the standard Solver named ranges, and thus can be solved using Solver or OpenSolver. It can also be edited using Solver. Testing on a range of spreadsheet models has shown that the auto-model algorithms work correctly for commonly taught model layouts including tabular models such as transportation.

The AutoModel feature has been developed by Iain Dunning, a student in the Engineering Science department at the University of Auckland

9 Advanced Features

OpenSolver offers a number of features for advanced users, including:

- The ability to easily solve an LP relaxation,
- Interaction with the CBC solver via the command line,
- Faster running using 'Quick Solve' when repeatedly solving the same problem with a succession of different right hand sides, and
- Viewing of the .lp problem file and CBC's solution file

Note that our large scheduling problem had to be solved repeatedly for 52 different right hand side values. The Quick Solve feature was developed to remove the need to repeatedly analyse the spreadsheet to determine the model coefficients. Using Quick Solve reduced the scheduling problem's run times for from one hour to one minute.

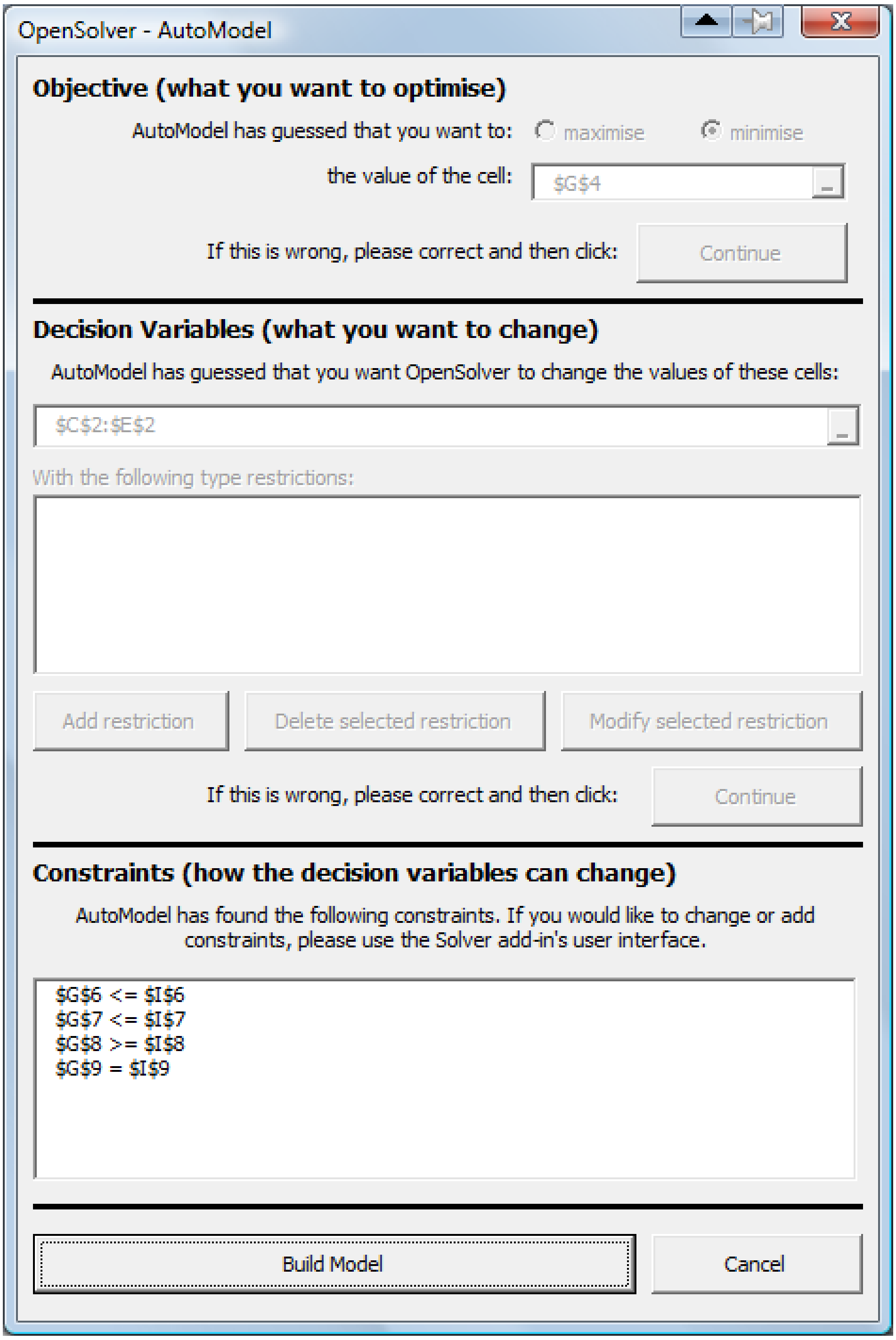


Figure 6: The auto-model dialog, shown here for the model in Figure 1, guides the user through the model building steps, allowing them to correct any mistakes made by OpenSolver's auto-model algorithms.

10 User Feedback

As at 10 November 2010, OpenSolver has been downloaded over 750 times since 16 July 2010. Since 20 June 2010, the OpenSolver web site, www.opensolver.org, has served over 6000 page views to over 1800 visitors. Much of this interest can be attributed to OpenSolver being featured on Mike Trick's OR blog (Trick, 2010) and mentioned on the "IEOR Tools" website (Larry, 2010). It is difficult to determine how OpenSolver is being used, but a comment by Joel Sokol from Georgia Tech on the OpenSolver web site describes one use of OpenSolver as follows:

Thanks, Andrew! I'm using OpenSolver on a project I'm doing for a Major League Baseball team, and it's exactly what I needed. It lets the team use their preferred platform, creates and solves their LPs very quickly, and doesn't constrain them with any variable limits. Thanks again! - Joel Sokol, August 11, 2010

11 Conclusions

We have shown that it is possible to leverage the COIN-OR software to provide users with a new open source option for delivering spreadsheet-based operations research solutions. While our software is compatible with Solver, it also provides novel tools for visualising and building spreadsheet models that we hope will benefit both students and practitioners.

References

CBC, 8 November 2010, https://projects.coin-or.org/Cbc
COIN-OR, 8 November 2010, www.coin-or.org
Dunning, I., personal communication, August 2010
FrontLine, 8 November 2010, www.solver.org
Larry, 7 July 2010, http://industrialengineertools.blogspot.com/2010/07/open-source-solver-for-excel.html
Pieterse, J.K., 6 November 2010, http://www.jkp-ads.com/officemarketplacenm-en.asp
Solver Limits, 8 November 2010, http://www.solver.com/suppstdsizelim.htm
Trick, M., 7 July 2010, http://mat.tepper.cmu.edu/blog/?p=1167
Walkenbach, J., "Excel 2010 Formulas", p75, Wiley Publishing, Indiana, 2010, online on 7 November 2010 at Google Books.
Wikipedia, "Microsoft Excel", 7 November 2010, http://en.wikipedia.org/wiki/Microsoft_Excel

Appendix 1: Solver's internal model representation under Excel 2010 and earlier versions. (This is a partial listing for Excel 2010.)

Name[1]	**Interpretation**	**Example Values**[2]
solver_lhs1, solver_lhs2, ...[5]	A range defining the left hand side of constraints 1, 2, ...,	=Sheet1!$G$6:$G$7 =Sheet1!$G$8
solver_rhs1, solver_rhs2, ...[5]	The right hand side of constraints 1, 2, ... defined by either a range or a constant value or one of the keywords: integer, binary	=Sheet1!$I$6:$I$7 =7
solver_rel1, solver_rel2, ...[5]	The nature of the constraint, being one of the following integer values: 1: <=, 2: =, 3: >=, 4: integer, 5: binary	=1 =binary
solver_num[5]	The number of constraints.	=2
solver_adj	A range defining the decision variables (termed adjustable cells)	=Sheet1!$C$2:$E$2

solver_lin[3,7]	1 if the user has ticked "Assume linear model", 2 if this is unchecked.	=1 =2
solver_neg[3]	1 if the user has ticked "Assume non-negative", 2 if this is unchecked.	=1 =2
solver_nwt	Solver's "Search" option value, 1="Newton", 2="Conjugate"	=1
solver_pre	Solver's "Precision" option value	=0.000001
solver_cvg	Solver's "Convergence" option value	=0.0001
solver_drv	Solver's "Derivative" option value, 1="Forward", 2="Central"	=1
solver_est	Solver's "Estimates" option value, 1="Tangent", 2="Quadratic"	=1
solver_itr	Solver's "Iterations" option value, the maximum number of iterations (branch and bound nodes?) Solver can run for before the user is alerted	=100
solver_scl	Solver's "Use Automatic Scaling" option value, 1=true, 2=false	=2
solver_sho[8]	Solver's "Show Iteration Results" option value, 1=true, 2=false	=2
solver_tim	Solver's "Max time" option value, the maximum number of seconds Solver can run for before the user is alerted	=100
solver_tol	Solver's "Tolerance" option value, stored as a fraction and displayed to the user as a percentage value.	=0.05
solver_typ	Objective sense, as in "Set target cell to": 1="Max", 2="Min", 3="Value of"	=2
solver_val	If solver_typ=3, this gives the target value for the objective function.	=0
solver_ver[8]	Solver version.(=3 in Excel 2010)	=3
solver_eng[8]	Engine. 1=Nonlinear, 2=Simplex, 3=Evolutionary	=2

Notes:

1: The name is local to the sheet, and so is prefixed by the sheet name, e.g. Sheet1!solver_lhs1.

2: The value always begins with an "=", e.g. "=2" or "=Sheet1!$C$2:$E$2"

3: Models built using older versions of Solver may not have this defined.

4: Some values may be missing. For example, a model can be defined that has no decision variables.

5: solver_num may be less than the apparent number of constraints. This occurs when constraints are deleted because Solver does not delete the unused solver_lhs, solver_rhs and solver_rel names.

6: References such as =Sheet1!#REF! can occur in the current model if cells have been deleted. These are detected by OpenSolver.

7: Excel 2010 removes the "Assume linear model" option, and instead asks the user to choose the specific solver (linear solver, non-linear solver or genetic algorithm).

8: Not defined in versions before Excel 2010

Dippy – A Simplified Interface to Advanced Integer Programming Techniques

Michael O'Sullivan, Qi-Shan Lim and Cameron Walker
Department of Engineering Science, University of Auckland
michael.osullivan@auckland.ac.nz

Stu Mitchell
Light Metals Research Centre, University of Auckland

Abstract

The development of mathematical modelling languages such as AMPL, GAMS, Xpress-MP and OPL Studio have made it possible to formulate mathethematical models such as linear programmes, mixed integer linear programmes and non-linear programmes for solution in solvers such as CPLEX, MINOS and Gurobi in a reasonable time frame. However, some models cannot be solved using "out-of-the-box" solvers, so advanced techniques need to be added to the solver frame-work.

Many solvers, including CPLEX and open source solvers such as Symphony and DIP, provide callback functions to enable users to customise the overall solution framework. However, this approach involves either expressing the mathematical formulation in a low level language such as C++ or Java or implementing a complicated indexing scheme to be able to track formulation components such as variables and constraints between the mathematical modelling language and the solver's callback framework.

In this paper we present Dippy, a combination of the Python mathematical modelling language PuLP and the open source solver DIP. Using Dippy, users can express their model in a straightforward modelling language and use callbacks that access the PuLP model directly. We discuss the link between PuLP and DIP and give examples of advanced solving techniques expressed simply in Dippy.

Key words: Integer programming; cutting planes; branch, price and cut.

1 Introduction

Using a high-level modelling language such as AMPL, GAMS, Xpress-MP or OPL Studio enables Operations Research practitioners to express complicated mixed-integer linear programming (MILP) problems quickly and (reasonably) easily. Once defined in one of these high-level languages, the MILP problem can be solved using one of a number of solvers. However, for many MILP problems, using solvers "out of the box" will only work for small problem instances (due to the fact that MILP problems are NP-hard). Advanced MILP techniques are needed for large problem instances and, in many cases, problem-specific techniques need to be included in the solution process.

Both commercial solvers such as CPLEX and open source solvers such as Cbc, Symphony and DIP (from the COIN-OR repository (LH03)) provide callback functions that allow user-defined routines to be included in the solution framework. However, to make use of the callback functions and develop the user-defined routines requires the user to first create their MILP problem in a low-level language (C, C++ or Java for CPLEX, C or C++ for Cbc, Symphony or DIP). While defining their problem, they need to create structures to keep track of appropriate constraints and/or variables for later use in their user-defined routines. For a MILP problem of any reasonable size and/or complexity, the problem definition in C/C++/Java is a major undertaking and, thus, a major barrier to the development of customised MILP frameworks by both practitioners and researchers.

Given the difficulty of defining a MILP problem in a low-level language, another alternative is to return to the high-level mathematical modelling languages (AMPL, GAMS, Xpress-MP, OPL Studio) or their open source contemporaries (such as FLOPC++, GAMSlinks and PuLP) to define the MILP problem. Then, by carefully constructing an indexing scheme, constraints and/or variables in the high-level language can be identified in the low-level callback functions and advanced MILP techniques used. However, implementing the indexing scheme is possibly as difficult as simply using the low-level language to define the MILP problem in the first place and so combining high-level and low-level languages does not remove the barrier to experimentation.

The purpose of the research presented here is the removal of the barrier that prevents easy experimentation with/customisation of advanced MILP solution frameworks. To achieve this aim we need to:

1. provide a straightforward modelling system so that users can quickly and easily describe their MILP problems;

2. enable the callback functions to easily identify constraints and variables in the solution framework.

PuLP already provides a straightforward modelling system for describing MILP problems in Python. Using PuLP a user can quickly and (reasonably) easily define a MILP problem and solve it using a variety of solvers including CPLEX, Gurobi and Cbc. Decomposition for Integer Programming (DIP) (RG05) provides a framework for solving MILP problems using 3 different methods: 1) branch and cut; 2) branch, price and cut; and 3) branch, decompose and cut.[1] In our research we extended PuLP so that a MILP could be defined and solved using DIP within a Python file. We also extended DIP to enable routines defined within the PuLP Python file to be called by the DIP callback functions. In addition to the existing DIP callback functions, we modified DIP to include advanced branching as a callback function. Table 1 gives the mapping of DIP callback functions to Dippy routines. Variable scope within Python means that that these user-defined

[1] The skeleton for a fourth method branch, relax and cut exists in DIP, but this method is not yet implemented.

(Python) routines have complete knowledge of the original problem defined in PuLP. Any extra information required to generate cuts, determine branches or generate columns is provided by the DIP callback functions. This "glue" between PuLP and DIP overcomes the barrier to easy customisation of DIP and provides both practitioners and students a straightforward interface for describing problems and customising their solution framework.

DIP callback function	**Description**	**Dippy mapping**
`chooseBranchSet`	Selects sets of variables and corresponding bounds for the down and up nodes of a branch	`branch_method`
`generateCuts`	Generates user-defined cuts for the current node solution	`generate_cuts`
`APPisUserFeasible`	Checks if the current node solution is feasible, i.e., there are no more user-defined cuts to be generated	`is_solution_feasible`
`APPheuristics`	Runs node heuristics using the current node solution as an input	`heuristics`
`solveRelaxed`	Solves a specified subproblem (relaxation) to generate master problem variables (columns)	`relaxed_solver`
`generateInitVars`	Finds initial master problem variables (columns)	`init_vars`

Table 1: Summary of Dippy callback functions

The rest of this article is structured as follows. Section 2 contains the description and model definition for a case study we will use to demonstrate the effectiveness of Dippy for experimenting with advanced MILP techniques. Next, in sections 3 and 4 we describe how to customise the DIP framework with Dippy. We conclude in section 5 where we discuss how Dippy enhances the ability of researchers to experiment with approaches for solving difficult MILP problems.

2 The Multiple Knapsack Problem (multiknap.py)

Extensions of this problem arise often in MILP problems including network design, rostering and trim loss. The problem is to determine how to place n items into m knapsacks in a way that minimises the wasted space in the knapsacks used. Each item $j = 1, \ldots, n$ has a weight w_j and each knapsack has (weight) capacity W.

The MILP formulation of the multi-knapsack problem is straightforward. The decision variables are

$$x_{ij} = \begin{cases} 1 & \text{if item } j \text{ is placed in knapsack } i \\ 0 & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if knapsack } i \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

$$s_i = \text{space left in knapsack } i$$

and the formulation is

$$\begin{array}{lll} \min & \displaystyle\sum_{i=1}^{m} s_i & \text{(minimise waste)} \\ \text{s.t.} & \displaystyle\sum_{i=1}^{m} x_{ij} = 1, j = 1, \ldots, n & \text{(assignment constraints)} \\ & \displaystyle\sum_{j=1}^{n} w_j x_{ij} + s_i = W y_i, i = 1, \ldots, m & \text{(capacity constraints)} \\ & x_{ij} \in \{0, 1\}, y_i \in \{0, 1\}, i = 1, \ldots, m, j = 1, \ldots, n & \end{array}$$

Using PuLP we can easily define this MILP problem in Dippy. The entire input file is given below with summaries for each section.

1. Load PuLP and Dippy;

```
import coinor.pulp as pulp
from pulp import *
import coinor.dippy as dippy
```

2. Get the problem data from another file. This determines $j = 1, \ldots, n$, $i = 1, \ldots, m$, $w_j, j = 1, \ldots, n$ and W;

```
from multiknap_ex1 import Weight, ITEMS, KNAPSACKS, Capacity
```

For `multiknap_ex1.py` $n = 5, m = 5, w = (7, 5, 3, 2, 2)^\top$ and $W = 8$.

3. Define the MILP problem and the problem variables;

```
prob = dippy.DipProblem("Multi-Knapsack")

assign_vars = LpVariable.dicts("InKnapsack",
              [(i, j) for i in KNAPSACKS
                      for j in ITEMS],
              0, 1, LpBinary)
use_vars    = LpVariable.dicts("UseKnapsack",
              KNAPSACKS, 0, 1, LpBinary)
waste_vars  = LpVariable.dicts("Waste",
              KNAPSACKS, 0, Capacity)
```

4. Define the objective function;

```
# objective: minimise waste
prob += lpSum(waste_vars[i] for i in KNAPSACKS), "min"
```

5. Define the constraints;

```
# assignment constraints
for j in ITEMS:
    prob += lpSum(assign_vars[(i, j)] for i in KNAPSACKS) == 1

# capacity constraints
for i in KNAPSACKS:
    prob += lpSum(assign_vars[(i, j)] *
                  Weight[j] for j in ITEMS) + waste_vars[i] == \
              Capacity * use_vars[i]
```

6. Solve the MILP problem using DIP with default options except for a user-defined tolerance and then display the solution;

```
dippy.Solve(prob, {
    'TolZero': '%s' % tol,
})

# print solution
for i in KNAPSACKS:
    if use_vars[i].varValue > tol:
        print "Knapsack ", i, " contains ", \
              [j for j in ITEMS
               if assign_vars[(i, j)].varValue > tol]
```

Running the preceding Python code takes 0.94s and 449 nodes and gives the following output:

```
Knapsack  1  contains  [3, 4]
Knapsack  2  contains  [1]
Knapsack  3  contains  [2, 5]
```

Note that DIP uses cuts from the COIN-OR Cut Generator Library (CGL) (LH03) by default. We can implement user-defined cuts in Dippy, but we haven't considered that customisation here.

3 Adding Customised Branching

In DIP we modified `chooseBranchVar` to become `chooseBranchSet`. This makes it possible to define:

1. a *down* set of variables with (lower and upper) bounds that will be enforced in the down node of the branch; and,

2. an *up* set of variables with bounds that will be enforced in the up node of the branch.

A typical variable branch on an integer variable x with integer bounds l and u and fractional value α can be implemented by:

1. choosing the down set to be $\{x\}$ with bounds l and $\lfloor \alpha \rfloor$;

2. choosing the up set to be $\{x\}$ with bounds $\lceil \alpha \rceil$ and u.

However, other branching methods may use advanced branching techniques such as the one demonstrated in the remainder of this section. From DIP `chooseBranchSet` calls `branch_method` in Dippy.

When solving the multiple knapsack problem one difficulty is symmetry in the solution space. Since the knapsacks are identical, solvers consider multiple solutions that differ only in the labelling of the knapsacks. To overcome this constraints for ordering the knapsacks can be ordered:

$$y_i \geq y_{i+1}, i = 1, \ldots, m - 1$$

```python
for n, i in enumerate(KNAPSACKS):
    if n > 0:
        prob += use_vars[KNAPSACKS[n-1]] >= use_vars[i]
```

This ordering branch also introduces the opportunity to implement an effective branch on the number of knapsacks. If $\sum_{i=1}^{m} y_i = \alpha \notin \mathbb{Z}$, then:

the branch down restricts $\sum_{i=1}^{m} y_i \leq \lfloor \alpha \rfloor$ and the ordering means that $y_i = 0, i = \lceil \alpha \rceil, \ldots, m$	the branch up restricts $\sum_{i=1}^{m} y_i \geq \lceil \alpha \rceil$ and the ordering means that $y_i = 0, i = 1, \ldots, \lceil \alpha \rceil$

To implement this branch in Dippy simply requires the definition of the `branch_method`. Note that Dippy can access the variables from the original formulation due to the scope of Python, no complicated indexing or searching is required (also note Python starts its array indexing at 0 – cf. C/C++).

```python
def choose_antisymmetry_branch(prob, sol):
    num_knapsacks = sum(sol[use_vars[i]] for i in KNAPSACKS)
    up = ceil(num_knapsacks)
    down = floor(num_knapsacks)
    if (up - num_knapsacks > tol) and \
       (num_knapsacks - down > tol):    # num_knapsacks is fractional
        # Define the down branch ubs, lbs = defaults
        down_branch_ub = [(use_vars[KNAPSACKS[n]], 0) for
                          n in range(int(down), len(KNAPSACKS))]
        # Define the up branch lbs, ubs = defaults
        up_branch_lb = [(use_vars[KNAPSACKS[n]], 1) for
                        n in range(0, int(up))]

        return ([], down_branch_ub, up_branch_lb, [])

prob.branch_method = choose_antisymmetry_branch
```

The effect of the ordering constraints and the advanced branching are given in section 5.

4 Adding Customised Column Generation

To solve the multiple knapsack problem using branch, price and cut using Dippy, no extra formulation is required. We simply need to identify the constraints for each subproblem and DIP will automatically generate the Dantzig-Wolfe restricted master problem. To add constraints to a subproblem in Dippy we use the same PuLP syntax, but add them to the appropriate `.relaxation` subproblem. In the multiple knapsack problem we use subproblems to “build” our knapsacks, so we form subproblems from the knapsack capacity constraints:

```
# capacity constraints
for i in KNAPSACKS:
    prob.relaxation[i] += lpSum(assign_vars[(i, j)] * Weight[j]
                                for j in ITEMS) + waste_vars[i] == \
                          Capacity * use_vars[i]
```

We can let DIP solve our subproblems, including the generation of initial variables, using its own MILP solver (in our case this is Cbc (LH03)). However, we may be able to speed up the overall solution process by providing our own approaches to solving the pricing subproblems and generating initial variables.

In the pricing subproblem we are looking for knapsacks that provide negative reduced cost. Including an item in the knapsack will add the reduced cost of x_{ij} (`assign_vars[(i, j)]`) but will decrease the waste and hence reduce the contribution of the reduced cost of s_i (`waste_vars[i]`). Here, we calculate the total contribution to reduced cost of adding an item (= reduced cost of $x_{ij}-$ reduced cost of $s_i \times w_j$) and then calculate an items efficiency by dividing this contribution by its weight. Then, we build a minimum reduced cost knapsack by using a greedy algorithm to choose items in order of best efficiency. Since we can access the problem data, variables and their reduced cost, this is straightforward to implement in Dippy:

```
def solve_subproblem(prob, index, redCosts, convexDual):
   knap = index

   # Calculate efficiency of items with negative reduced cost
   effs = {}
   for j in ITEMS:
       effs[j] = (redCosts[assign_vars[(knap, j)]] -
          redCosts[waste_vars[knap]] * Weight[j]) / Weight[j]

   # Sort the dictionary by value
   seffs = sorted(effs.items(), key=itemgetter(1))

   # Add efficient items to the knapsack if they fit
   kp = []
   waste = Capacity
   for p in seffs:
       j = p[0]
       if Weight[j] <= waste:
           kp.append(j)
           waste -= Weight[j]

   # Calculate the reduced cost of the knapsack
   rc = sum(redCosts[assign_vars[(knap, j)]] for j in kp)
   rc += redCosts[use_vars[knap]] + redCosts[waste_vars[knap]] * waste
```

```
    # Return the solution if the reduced cost is low enough
    if rc < convexDual:
        var_values = [(assign_vars[(knap, j)], 1) for j in kp]
        var_values.append((use_vars[knap], 1))
        var_values.append((waste_vars[knap], waste))

        dv = dippy.DecompVar(var_values, rc - convexDual, waste)
        return [dv]

    return []

prob.relaxed_solver = solve_subproblem
```

To generate initial knapsacks we implemented two approaches. The first approach used a first-fit approach and considered the items in order of decreasing weight. The second approach simply placed one item in each knapsack. Using Dippy we can define both approaches at once and then define which one to use by setting the `solve_relaxed` method:

```
def first_fit(prob):

    kps = []
    sw = sorted(Weight.items(), key=itemgetter(1), reverse=True)
    unallocated = [s[0] for s in sw]
    while len(unallocated) > 0:
        kp = []
        waste = Capacity
        j = 0
        while j < len(unallocated):
            if Weight[unallocated[j]] <= waste:
                item = unallocated[j]
                unallocated.remove(item)
                kp.append(item)
                waste -= Weight[item]
            else:
                 j += 1
        kps.append((kp, waste))

    bvs = []
    for k in range(len(kps)):
        knap = KNAPSACKS[k]
        kp = kps[k][0]
        waste = kps[k][1]
        var_values = [(assign_vars[(knap, j)], 1) for j in kp]
        var_values.append((use_vars[knap], 1))
        var_values.append((waste_vars[knap], waste))

        dv = dippy.DecompVar(var_values, None, waste)
        bvs.append((knap, dv))

    return bvs
```

```python
def one_each(prob):

    bvs = []
    for i, knap in enumerate(KNAPSACKS):
        kp = [ITEMS[i]]
        waste = Capacity - Weight[ITEMS[i]]
```

⋮ (generate a `dv` and add `bv` as in `first_fit`)

```python
    return bvs

prob.init_vars = first_fit
##prob.init_vars = one_each
```

The effects of column generation, included a customised subproblem solver and initial variable generator are shown in section 5.

5 Conclusions

Table 2 shows the effect on the solve time of the ordering constraints (OC), advanced branching (AB), column generation (CG), customised subproblem solver (CS) and initial column generator using first-fit IC_{FF}) and one item each (IC_{OE}). Note that in some cases the solve process was affected by the order of the items. To account for this variation we ran 5 tests for each strategy with the order of the items randomised. The results in Table 2 show 95% confidence intervals from these tests.

Strategies	**Nodes**	**Time (s)**
Default[2]	[502.23, 650.97]	[1.24, 1.51]
OC	[88.80, 96.40]	[0.29, 0.32]
OC + AB	[7, 7]	[0.11, 0.11]
CG	[35.72, 41.48]	[10.38, 13.07]
CG + CS	[49.23, 65.57]	[10.39, 14.44]
CG + IC_{FF}	[26.68, 30.52]	[8.69, 9.36]
CG + IC_{OE}	[26.30, 42.10]	[7.66, 11.58]
CG + CS + IC_{FF}[3]	[9.04, 73.76]	[2.07, 16.76]
CG + CS + IC_{OE}	[58.09, 86.31]	[12.59, 17.66]

Table 2: Results from solving the multiple knapsack problem `multiknap_ex1`

The results in Table 2 show that the ordering constraints and advanced branching significantly reduce the size of the search tree for branch and cut. They also show that the customised subproblem solver with first-fit generation of initial variables may result in a small search tree for branch, price and cut. Note that we did not consider all the possible customisations provided by Dippy, including the ability to generate cuts and add both a root heuristic and node heuristics.

Dippy presents the best of both worlds. It provides a mathematical modelling framework that makes it easy to quickly formulate MILP problems. It also makes it easy to quickly customise the MILP solution framework to experiment with

[2]The DIP default is branch and cut with the CGL library.

[3]The minimum solution scenario had 1 node and took 0.34s.

the effectiveness of advanced MILP techniques for the problem being solved. We have used Dippy successfully to enable final year undergraduate students to experiment with advanced branching, cut generation, column generation and root/node heuristics. Dippy breaks down the barrier to experimentation with advanced MILP approaches for both practitioners and researchers, thus allowing them to concentrate on furthering Operations Research knowledge instead of programming MILP formulations in a low-level language.

6 Acknowledgments

The authors would like to thank the authors of DIP, Ted Ralphs and Matt Galati, for their help throughout this project and the Department of Engineering Science at the University of Auckland for their support of Qi-Shan during this research project.

References

Robin Lougee-Heimer, *The common optimization interface for operations research*, IBM Journal of Research and Development **47** (2003), no. 1, 57–66.

T K Ralphs and M V Galati, *Decomposition in integer programming*, Integer Programming: Theory and Practice (J Karlof, ed.), CRC Press, 2005.

Constraint Branching Techniques

Alastair McNaughton
Department of Mathematics
University of Auckland
Private Bag 92019
New Zealand
a.mcnaughton@auckland.ac.nz

David Ryan
Department of Engineering Science
University of Auckland
Private Bag 92019
New Zealand
d.ryan@auckland.ac.nz

Abstract

This talk will outine the applicability of constraint branching particularly in relation to very large problems. It will be discussed in relation to constrained forest harvesting.

Key words: forest harvesting, adjacency branches, area restriction model.

1 Introduction

Many significan applications in OR can be modelled as mixed integer programs. It is common for the shere size of a modern industrial application to preclude most of the usual formulations. In these cases a combination of constraint branching and column generation may be worth considering. The particular application we address is the area-restricted forest harvesting problem.
The basic idea is that we begin with a highly simplified formulation which omits any complcated constraints. This relaxed model is optimized and the relaxed solution is searched for any infeasibilities with respect to the real application. Then, instead of removing these by the inclusion of additional constraints, we impose branches that remove the infeasible solution and permit only the generation of new columns which are feasible with respect to the relevant

restrictions. The success of the method depends on the availability of a very fast column generation technique.
The choice of decision variables should harmonize with the constraint branches. this works best if the decision variables are in a sense orthogonal to the constraint branches. In our forestry application we use road harvest plan variables in which a single variable represents a plan detailing the entire management of all blocks on the road in question throughout the planning horizon. As this is a further comment on an ongoing research project the details of the model formulation are omitted. Please refer to the bibliography for these.

2 The Ideal Constraint Branch

A constraint branch should move the relaxed solution smoothly and swiftly towards the optimal feasible solution. To achieve this a number of things are required. The branch should have a sensible practical meaning. The branch should involve the removal (inclusion) of considerable number of candidate columns. The branch should incorporate time aspects. The branch should have an easily identified most likely side. The selection of branch nodes should allow prioritization such that the relaxed LP solution moves naturally towards those parts of the solution space which have attractive objective values. The branch should help to integerize the problem. Such a branch will likely be particular to the application.

3 A Nuclear Set

We illustrate the concept of a constraint branch with a practical example taken from the area constrained forest harvesting problem. A nuclear set is a local arrangements of harvesting units consisting of a central nucleus surrounded by some perimeter blocks. The *nucleus* is a contiguous block of units with total area less than or equal to the maximum clearfell area. The *perimeter* consists of the surrounding units which are adjacent to the nucleus such that the total area of nucleus plus each individual perimeter unit exceeds the maximum clearfell area. An example of a nuclear set is given in Figure 1. This detail taken from the 400 unit forest is circled in Figure 2. The maximum clearfell area is 30 hectares. Units 75 and 95 form the nucleus, and units 55, 56, 74, 94 and 114 are the perimeter. Unit 96 is not part of the perimeter since its area is too small. A nuclear set is not restricted to the units along 1 road. The example given in Figure 1 spans 4 roads. The concept of the nuclear set lies at the heart of the constraint branch. Figure 2 illustrates the orthogonality of the decision variables and the nuclear sets.

4 The Column Generation

There are many standard approaches to column generation. For a large industrial application involving constraint branching it is imperative that the column generation be extremely rapid. We use a technique that should be suitable to

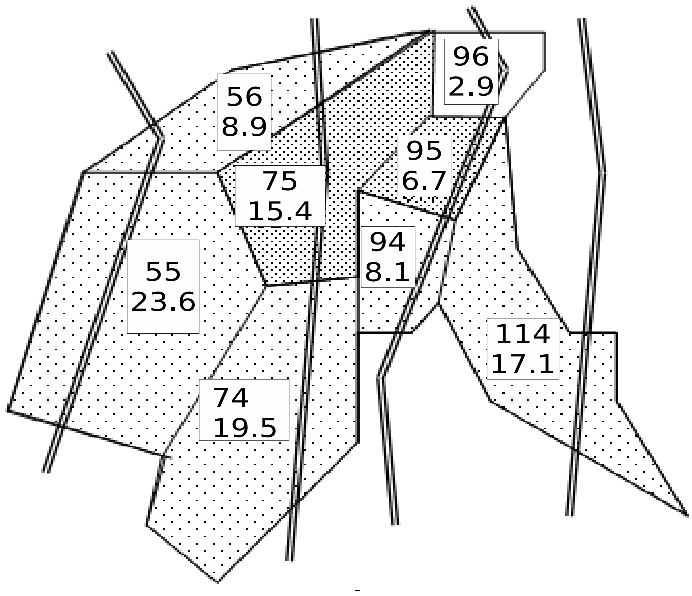


Figure 1: A nuclear set with nucleus (dark shaded) and perimeter (light shaded).

many applications in which the size of the problem stems from combinatorial complexity. We begin with a set of elementary columns each representing the harvest of just 1 unit in 1 particular year.The reduced costs of these columns are then decomposed and recombined so as to construct a new combined column representing the harvesting of several blocks throughout the time horizon so as to achieve optimality with respect to the current branches.

5 The Algorithm

We discuss the solution algorithm for the area restricted forest harvesting problem in the hope that this in some way gives an outline that could be adapted to other applications. The solution algorithm deals with the planning horizon as a single optimization. The strategy is as follows. A list is made of all possible nuclear sets. We then wait to see which of these will be required. For example, the nuclear set in Figure 1 would be included in our list at this stage. However, it would only be used in the solution algorithm if at some stage a relaxed LP solution happened to contain it as part of an adjacency violation.
Figure 3 presents a flow diagram. The forest harvesting problem is initially formulated and solved as a relaxed LP, with no adjacency or integer requirements. A phase of column generation follows with a number of composite road harvest plan variables being added to the model. Optimization (2) follows. The solution obtained is searched so as to find any cases of adjacency violation. This is done by

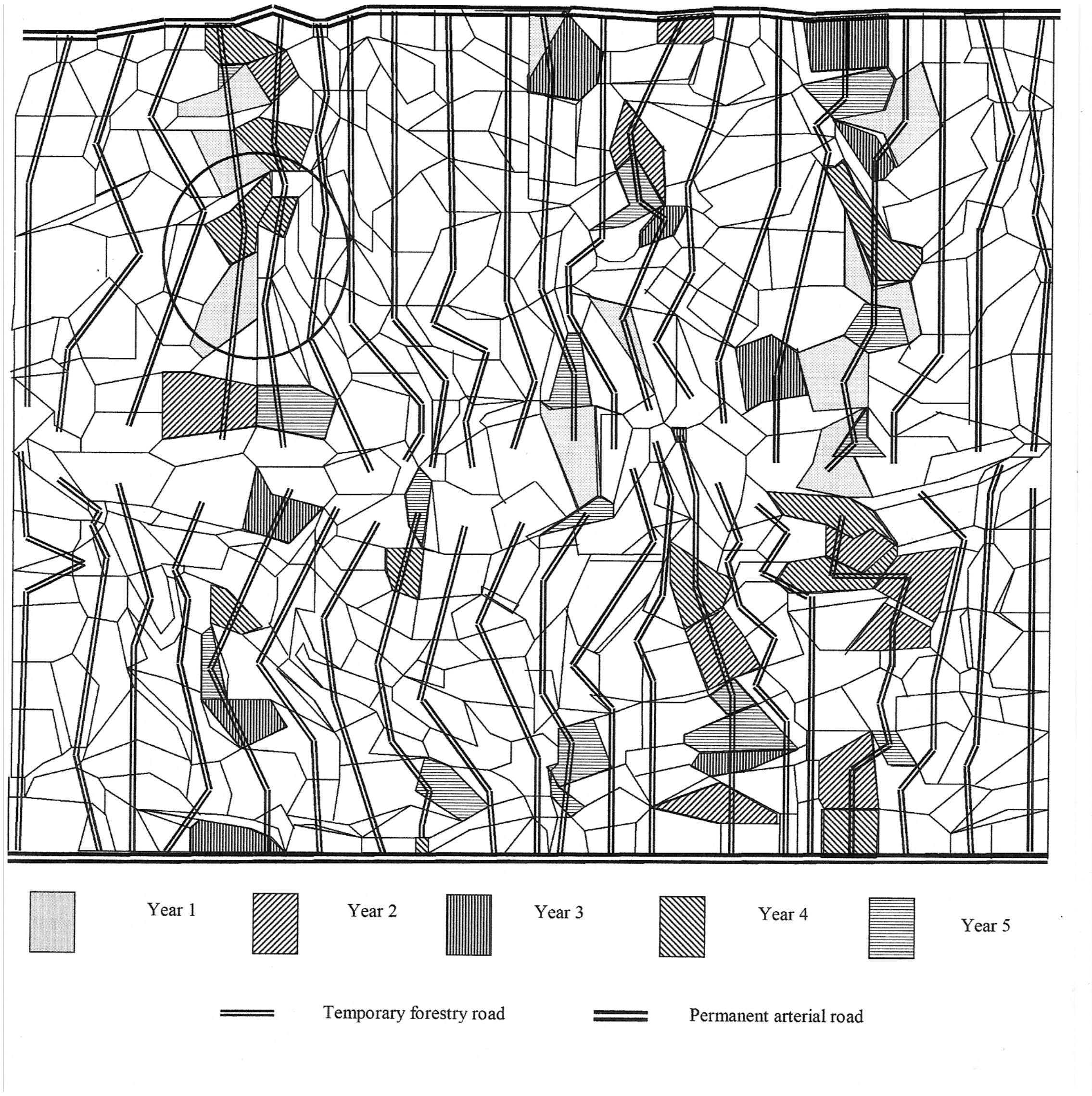


Figure 2: A 400 unit poorly regulated forest. The solution for the trial using a planning horizon of 5 time periods with a green-up of 2 time periods is shown.

a simple scanning process in which each nuclear set on the list is checked sequentially time-wise. An infringement is detected in the form of an identification of a nuclear set with a time interval $[a, b]$, where all the units in the nucleus are harvested within the interval $[a, b]$, with $b - a < T$, and at least 1 unit in the perimeter harvested during the time interval $[b - T + 1, a + T - 1]$. Each iteration only 1 infringement is dealt with.

Each adjacency branch is associated with a nuclear set. In the 1-branch all units in the nucleus are felled within the time interval $[a, b]$. Concurrently with this, all the units in the perimeter are left unharvested during the appropriate time interval, $[b - T + 1, a + T - 1]$. After each adjacency branch has been implemented, the modified linear programme is re-solved, with more column generation as required. During optimization (1) the new branch is enforced by an artificial penalty on the decision variables which are to be removed by the branch. Column generation then produces several new columns. After this the decision variables which are being removed have their upper bounds set at 0 and optimization (2) follows. If the solution obtained is unacceptable perhaps due to fathoming or to infeasibility, then we back-track. This involves replacing the 1-branch with the 0-branch. The process is repeated until no adjacency violations can be detected. At this stage the solution may still contain fractional values of the decision variables. Integer branches are then used until an integer solution with an acceptable objective value has been obtained. Then the problem is re-optimized with more column generation after every branch. During this integer branching, regular checks are made to detect any further adjacency violations with further adjacency branches are implemented as required. A feasible integer solution is obtained once no adjacency violations and no fractional values of decision variables are left.

6 Adjacency Branches

This section is the main distinctive of this paper. Again we treat it by presenting the area restricted forest harvesting problem as a typical example. After each episode of column generation, each nuclear set on the previously prepared list is scanned. Let us suppose an adjacency violation is found with the nucleus felled during a time interval $[a, b]$. If there are several we select the one with the nucleus of greatest yield. We impose the 1- branch which forces the nucleus units to be harvested within the time interval $[a, b]$.
To implement an adjacency branch we impose an upper bound of 0 on every road harvest plan variable that represents the harvesting of any of the nucleus units *outside* the interval $[a, b]$, including the possibility of a null harvest. Also we require there to be no harvesting of any of the perimeter units during the period $[b - T + 1, a + T - 1]$, where T is the number of time periods in the green-up. To do this we impose an upper bound of 0 on all road harvest plan variables that represent the harvesting of any of the perimeter units *inside* the interval $[b - T + 1, a + T - 1]$.
The adjacency branches also tend to remove fractions from the relaxed LP and so work harmoniously with the integer branches. In the trials the algorithm gave precedence to adjacency branches, followed by integer branches. Nuclear sets corresponding to parts of the forest desirable for harvesting take integer values at an early stage during the algorithm. As a consequence they make adjacency branches which occur early in the branch and bound tree. In this way the branching is prioritized in a good way.
Integer branches remove any remaining fractional values from the variables, x_{jn}, in the relaxed LP, so as to obtain an integer solution. First, the required harvesting time for each unit is obtained from the current relaxed LP solution. For some units the result may be a series of fractions, relative to the time periods, which sum to 1. These are spread over a time interval, say $[a, b]$. We find the smallest fraction associated with either a or b across all the possible units. If the

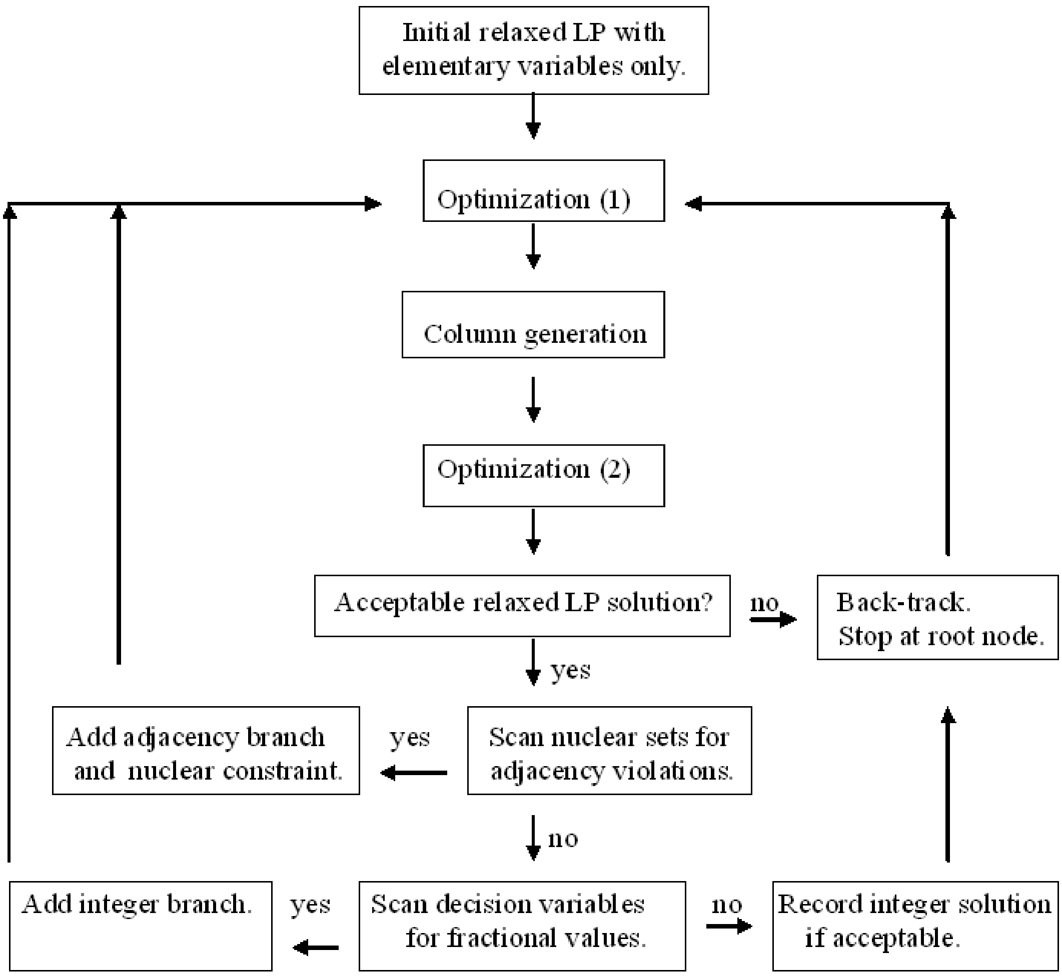


Figure 3: A flow chart of the solution algorithm.

smallest fraction is associated with unit k on road j in time period a, then the 1-branch requires unit k to be harvested after time a. Any variable x_{jn} in which unit k is harvested up to time period a is removed from the problem by having its upper bound set at 0. The associated 0-branch requires unit k to be harvested no later than time period a.

relaxed LP (objective	green-up (time periods)	objective (million $US)	upper bound	optimality gap	time (seconds)	adjacency branches
forest type: poorly regulated						
17.83						
	0	17.77				
	1	17.73	17.81	0.08	34	88
	2	17.62	17.75	0.13	45	142
	3	17.36	17.65	0.29	47	176
	4	17.35	17.51	0.16	47	193
	5	17.02	17.36	0.34	122	216
forest type: well regulated						
15.65						
	0	15.60				
	1	15.50	15.63	0.13	30	96
	2	15.35	15.55	0.20	27	151
	3	15.23	15.45	0.22	57	191
	4	15.18	15.25	0.07	180	240
	5	15.12	15.23	0.11	20	205
forest type: over mature						
20.31						
	0	20.24				
	1	20.20	20.24	0.04	40	60
	2	20.15	20.23	0.08	138	131
	3	20.10	20.12	0.02	228	141
	4	19.96	19.97	0.01	85	204
	5	19.60	19.72	0.12	80	208

Table 1 : Results from trials with a forest of 400 units.

7 Conclusions

Table 1 shows typical performance of the algorithm with respect to the area restricted forest harvesting problem with a variety of data set simulations comprising forests of 400 units. Various forest types have been used so as to test the robustness of the model. The treatment of green-up spanning up to 5 time periods is very significant. In each case the planning horizon is 25 time periods. The level of resolution in the trials indicates that this approach compares favourably with other area restricted forest harvesting models in terms of fine detail, with regard to both time and area. It is hoped the methods presented here may achieve similar improvements in some other applications involving massive combinatorial complexity.

References

[1] Gunn, E.A. and E.W. Richards. 2005. *Solving the adjacency problem with stand-centered constraints*, Can. J. For. Res. 35, pp 832-842.

[2] McDill, M.E., S.A. Rebain and J. Braze. 2002. *Harvest Scheduling with Area-Based Adjacency Constraints*, Forest Science 48, pp 631 - 642.

[3] McNaughton, A.J., M. Rönnqvist and D.M. Ryan. 2000. *A Model which Integrates Strategic and Tactical Aspects of Forest Harvesting*. In System Modelling and Optimization, Methods, Theory and Applications, Edited by M.J.D. Powell and S. Scholtes, Kluwer Academic Publishers Boston, pp 189-208.

[4] McNaughton, A.J., G.D. Page and D.M. Ryan. 2001. *Adjacency Constraints in Forest Harvesting*, proceedings of the ORSNZ, 2001, pp 9-15.

[5] McNaughton, A.J. 2002. *Optimisation of Forest Harvesting Subject to Area Restrictions on Clearfell*, proceedings of the ORSNZ, 2002, pp 307-313.

[6] McNaughton, A.J. 2003. *Adjacency constraints and adjacency branches*, proceedings of the ORSNZ, 2003, pp .

[7] McNaughton, A.J. 2004. *Recent Progress on the Area Restriction Problem of Forest Harvesting*, proceedings of the ORSNZ, 2004, pp .

[8] McNaughton, A.J. and D.M. Ryan. 2007. *Area-restricted forest harvesting with adjacency branches* , proceedings of the ORSNZ, 2007, pp 114-118

[9] McNaughton, A.J. and D.M. Ryan. 2008. *Adjacency branches used to optimize forest harvesting subject to area restrictions on clearfell* , Forest Science 54(4), 2008, pp 442 - 454.

[10] McNaughton, A.J. and D.M. Ryan. 2009. *Area restricted forest harvesting with adjacency branches* , proceedings of the ORSNZ, 2009, pp 110-119

[11] Murray, A. 1999. *Spatial Restrictions in Forest Scheduling*, Forest Science 45(1), pp 45-52.

[12] Murray, A.T. and A. Weintraub. 2002. *Scale and Unit Specification Influences in Harvest Scheduling with Maximum Area Restrictions*, Forest Science 48, pp 779-789.

[13] Vielma, J.P., A.T. Murray, D. Ryan and A. Weintraub. 2003. *Improved Solution Techniques for Multiperiod Area-based Harvest Scheduling Problems*, Systems Analysis in Forest Resources: Proceedings of the 2003 Symposium, pp 285-290.

A Users' Critical Characteristics (UCC) in ERP System Training Transfer: An HRD Perspective

Christopher Arasanmi
The University of Auckland
c.arasanmi@auckland.ac.nz

Abstract

Several reasons explained organisational motivations for embarking on enterprise resource planning systems; this includes elimination of inefficiency, and provision of better control on operations and cost. In spite of ERP contributions to business success, it was speculated that about 50% of ERP projects failed to justify the millions of dollars invested into the projects. The reasons for the failure apart from other reasons are rooted in human related factors, including user characteristics. Evidence shows that, ERP success is dependent on the skills and the human capital of an organisation; hence a fundamental issue in effective ERP implementation.

Management information systems (MIS) studies have acknowledged training as a critical factor in the success of ERP system implementation. This is apparent in the financial expenditures on ERP training and post implementation training of the users. Since training influences users' attitudes, behaviour and performance, the ERP system implementation requires that training must be imparted to a substantially large user group. However, there is paucity of knowledge on the determinants of users' ERP training application in an ERP system environment. Therefore this proposed research examines Users' Critical Characteristics (UCC) in ERP training effectiveness.

Key words: ERP, users, training, transfer, critical characteristics, systems.

Analyzing the Brazilian Higher Education System using System Dynamics

Luisa Mariele Strauss
University of Vale do Rio dos Sinos
Brazil
luisa@unisinos.br

Denis Borenstein
Management School
Federal University of Rio Grande do Sul
Brazil
denisb@ea.ufrgs.br

Abstract

Higher education in Brazil has experienced a rapid expansion since the publication of the Directives and Bases Law (LDB) in 1996 and the pliability of the Government for the launching of new programs and educational institutions, mainly driven by the private sector. Despite this expansion, Brazil has not yet reached the aim expected in the Education National Plan: 30% of young people from 18 to 24 years old in higher education in 2010. Moreover, the demand for undergraduate programs presents signs of retraction, characterizing a system with fast initial growth followed by stagnation. This work proposes to understand the dynamics of this system by developing a model in System Dynamics, analyzing the undergraduate higher education. The model considers regulation, aims, demand, supply and especially the balance between public and private sectors. The methodology consists in the stages of problem definition, formulation of the dynamic hypothesis, development of the simulation model and in the validation and scenario testing. This step resulted in an analysis of alternative scenarios for the Brazilian undergraduate higher education. As a result, this model allows one to analyze the possible behavior of key variables in each scenario and to make observations on the variables that are not defined in the real system.

Key words: System Dynamics, Modelling, Higher Education, Brazil.

1. Introduction

Higher education can be seen as a public asset, insofar as it benefits the society as a whole (Mizrahi & Mehrez, 2002). Dias Sobrinho & Brito (2008), Dalvi *et al.* (2005), Porto & Régnier (2003) highlights that access to education is necessary for economic growth, indicative of social justice, and, especially in Brazil, an aspiration of young people seeking social ascension. According to Carneiro (1998), access to quality education is one of the aspects needed to achieve equality among people. Given its importance, higher education access was considered by the legislator as a social right guaranteed by the Brazilian Constitution.

Taking into consideration several aspects such as: the strategic role of higher education, the regulatory policies and legislation, and the coexistence of public and private institutions acting in this field, this paper aims to answer the following question: "How to help managers involved in higher education in better understanding the dynamics of the undergraduate education system in Brazil, in view of regulatory and political issues, budget constraints, different curves of supply and demand, fluctuations in enrollment, different levels of quality, goals, and socioeconomic status?".

The main objective of this paper is to describe the system dynamics model developed to address this research question. The model was implemented in iThink© and is able to evaluate the dynamic behaviour of the undergraduate education system considering several different scenarios, regulatory policies, and strategies from the several actors involved.

This paper is organized as follows. The next section briefly describes major aspects related to the higher education system in Brazil. Section 3 introduces the developed systems dynamics model and how the model was partially validated. The final section presents the conclusions, limitations and further developments.

2 The Undergraduate Brazilian System

In Brazil, education, at all levels, can be provided both by the public and the private sector, according to the 1988 Federal Constitution and Law 9.394/1996, the Law of Directives and Bases of Education. Aiming to expand the system and access to higher education, the government relaxed in 1995 the rules for opening new courses and higher education institutions (Dias Sobrinho & Brito, 2008, Dalvi *et al*. 2005). The response was the rapid growth of higher education as a whole, driven primarily by the private sector. According to McCowan (2007), private institutions have grown in number and size worldwide, and Latin America has changed from a small and elitist public system to a diversified system in which the private sector has an important role, since it represents a viable solution for expansion. This view is shared by Schwartzman and Schwartzman (2002), for which private enterprise is no longer seen as a "necessary evil". Beyond Latin America, other countries such as Israel (Mizrahi & Mehrez, 2002, Shoham & Perry, 2009), Taiwan (Ka-Ho Mok, 2002) and Romania (Niculescu, 2006), the participation of private enterprise in education is significant and had an important role in the expansion of the sector.

The rapid expansion of private initiative has created a situation of competition between institutions. Higher education is now seen also as an economic sector, a business area (Dalvi *et al*. 2005, Porto & Régnier, 2003, Schwartzman & Schwartzman, 2002). The coexistence of the public education, aiming at quality, and a private sector, motivated by profit, is one of the main topics of this research work.

In 2009, there were 2,250 public and 244 private higher education institutions in Brazil. The private sector accounts for around 75% of all undergraduate enrollments. The rapid growth of this sector was due to the supply due to a better economic situation of the population, a more affordable private sector, and finally by the existence of a pent-up demand, e.g., people already in the labor market and wanted to return to studies (Dalvi *et al*., 2005) to enhance their competitiveness in the market. Fig. 1 shows the evolution of the number of students enrolled in undergraduate courses in Brazil, from 1980 to 2008. There is an accelerated growth from the approval of the LDB in 1996, being the private sector the major participant. It took 20 years, from 1980 to 2000 for doubling the number of enrolled students, but only eight years, 2000 to 2008, for that number to double again.

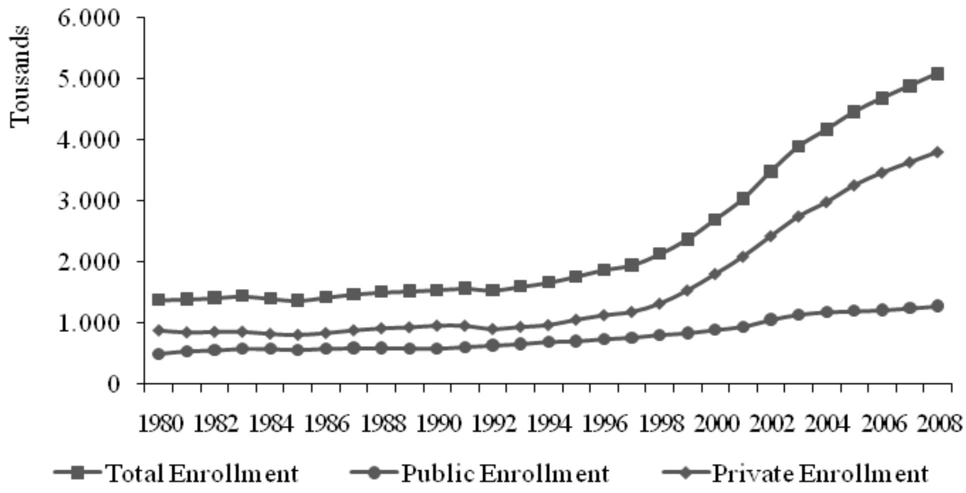


Figure 1 – Higher Education Enrollment

3. System Dynamics Model

System Dynamics is a method to enhance learning in complex systems. The main application of SD is in the dynamic behavior of complex systems, for which mathematical models are of little use due to the strong and complex interaction among the several variables involved.

The higher education system in Brazil has experienced rapid growth since 1995. Such growth, though desired, was very fast and therefore did not come without its problems. Senge (2004) warns that “when growth becomes excessive [...] the system itself will seek to compensate him”. In this case, the expansion was a result of legislation and government policy. However, when the government started to note that the private sector growth was chaotic, it took some steps to halt this situation. As a consequence, several investments in the private sector did not fulfill their potential, establishing a half-crisis situation.

The major objective to the development of the SD model is to evaluate the behavior of the system, given that the involved agents, government and private sector, can play several different strategies, and the environment in general, such as the macroeconomic and demographic aspects of the country, can have a sensible effect over the system.

The model was developed following the methodology presented by Sterman (2000). The causal loop diagram emphasizes the balance between the public and the private sectors and sought to establish the dynamics between them under government and market constraints and regulations. The elements presented in the causal loop diagram were converted into stock and flows, whose relationships were modeled using the data collected in the Ministry of Education site and expert opinions. The defined structures were then used to build a simulation model that allowed us to analyze the dynamic behavior of the involved variables. The outputs were compared with actual data, indicating that the model, albeit with some caveats, is a satisfactory representation of the system. The last step of the methodology was the analysis of scenarios, not presented in this article for the sake of economy. This analysis aimed to assess the potential of the model as a learning tool. Next, the two first steps are briefly described.

3.1 Causal Loop Diagram

Fig. 2 shows the causal loop diagram developed for the addressed problem. The diagram emphasizes the coexistence of public and private education and their main elements as follows: (i) demand for higher education, (ii) the regulatory policies; (iii) the public sector, (iv) the private sector, and (v) the attractiveness between the two sectors.

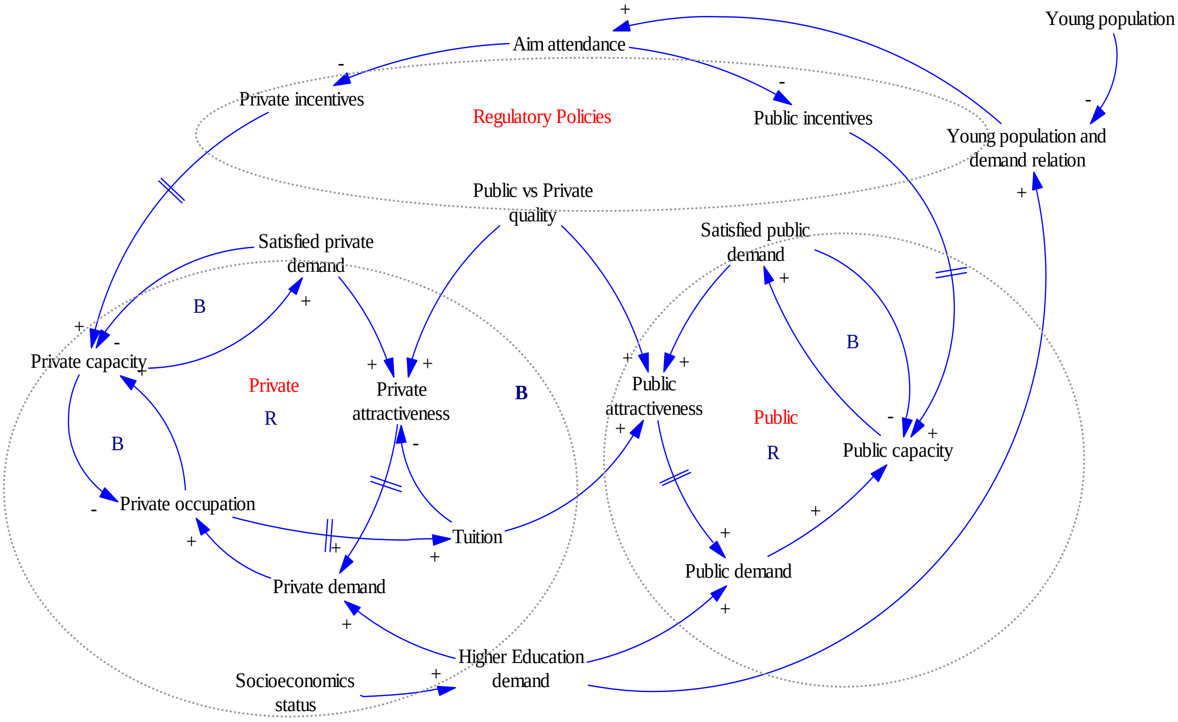


Figure 2 – Causal Loop Diagram

The demand for higher education is influenced by socioeconomic status. Policies for higher education are set by the Ministry of Education. The combination of policies in the model is translated into incentives to create new openings for graduate students in both the public and the private sectors.

The loops that represent the public and private sectors include the service capacity of each sector. They present reinforcement loops, where the growth in capacity is driven by the demand and the perceived incentives by the government; and balance loops to limit the exponential growth of the system, as demand is limited. In the private sector, there is the additional issue of pricing: tuition increases with the demand. If the capacity starts to idle, the tuition tends to fall. The balance between the two sectors is given by the attractiveness of the private sector in relation to public sector, which takes into account the quality (computed by the Ministry of Education), competition (relation candidates/places), and tuitions (inspired by the model developed by Carvalho (2001)).

3.2 Stock and Flow Diagrams

The stock and flow diagrams were developed with iThink 9.1.2. We divided the model into smaller units, all intertwined, in an attempt to facilitate its understanding. These units were nominated as follows: (i) demand for higher education, (ii) policies for higher education, (iii) the public sector, (iv) distance learning in the public sector, (v) the private sector, (vi) distance learning in the private sector, (vi) pricing; and (vii) attractiveness of the private sector in relation to the public sector.

The demand for higher education is based on the number of students graduating from the high school system (prerequisite for entrance in the higher education), pent-up demand, and the socioeconomic conditions of the population. The high school system is represented by a structure of stock (Enrolled) where entering students stays a predetermined period of time and then exits (Finished). The abandonment rate (high in

Brazilian higher education system) is represented as an exhaust flow (Escape). The outflow of the high school system and the pent-up demand generate the demand for higher education, highly influenced by socioeconomic aspects (represented in the model by GDP per capita) and the attractiveness of the private sector in relation to the public one. Finally, it is considered that distance learning allows an increase in the demand for higher education.

Policies for higher education can be regarded as intangible and abstract, and therefore very difficult to be directly represented in the model. The solution adopted was to model incentives as auxiliary variable that varies from 0 to 1. The coexistence of public and private sectors is represented by means of auxiliary variables *Private* and *Public*, combined with other auxiliary variables that represent changes in the regulation policies. The main goal for higher education defined by the Ministry of Education is to enroll 30% of the young population aged 20 to 24 years in an undergraduate course in the period of 2001 to 2010. The comparison between the number of enrolled students by the age-adjusted series with the target is verified by the discrepancy variable, as shown in the equation below.

$$Discrepancy = Min\ (1, ((Enrollment\_Pv + Enrollment\_Pb + Enrollment\_Pv\_DL + Enrolled\_Pb\_DL) * Age\_distortion) / Target))$$

where *Enrollment_** is the number of students enrolled in the private sector (Pv) or public sector (Pb) or distance learning (DL), *Age_distortion* represents the proportion of students enrolled, but out of the range 20 to 24 years old , and *Target* is the objective defined by the regulator. The discrepancy from the target increases or decreases the amplitude of government's incentives to open new places in higher education.

Fig. 3 shows the complete modeling of policies for higher education. The budget allocated to education is also considered.

The public sector, the private sector, and the distance learning within the public and private sectors units follow a similar structure. Each sector contains two stock structures: *Places* and *Enrollment*. The *Places* inventories are obtained from the adjustments in the flows of opening and closing places, resulting from government control and incentives, as well as entrance, fusions and bankruptcy of private institutions. The main difference between the public and private sectors is the insertion of the variable price (representing tuitions) in the modeling of the latter sector.

Stocks related to student enrollment include one in-flow of fresher students and two out-flows related with graduates and drop outs, respectively. They are modeled differently in the two sectors. In the public sector, a function computes the occupation of places as an average of the collected data in the Census of Higher Education, annually defined by the Ministry of Education. In the private sector, this relationship has suffered a decline since 2003, and in recent years, is around 50% (INEP, 2008). Therefore, the entrance flow is also linked to the economic situation of the population. Fig. 4 shows the structure of the public sector unit.

The Price Definition unit was modeled based on Sterman (2000), introducing some adaptations. Finally, the attractiveness unit considers three criteria: entrance requisites for both sectors, tuitions, and education quality. Strauss (2010) describes with details these two units.

3.3 Model Validation

Model validation occurred through analysis of extreme situations (tests included in borderline situations of the real system), sensitivity analysis (to check for discrepancies between the expected behavior and the simulation result), and especially comparisons with actual data observed.

Comparisons between real and simulated output data were performed in a qualitative way, focusing on trends rather on a rigorous statistical analysis. Several comparisons were carried out, mainly for variables for which historical data were available such as students' enrollment and graduating from high school, freshmen enrollment in higher education and graduating, places offered, and applicants in higher education. To illustrate the carried out analysis, Fig. 5 shows the comparison made for the number of places for each sector since 1990. This indicator influences all variables in the system. The model has captured the growth observed for both sectors. The real data for the private sector have faster growth than the simulated ones, but the simulation has captured the fairly captured the verified behavior, and therefore was considered adequate.

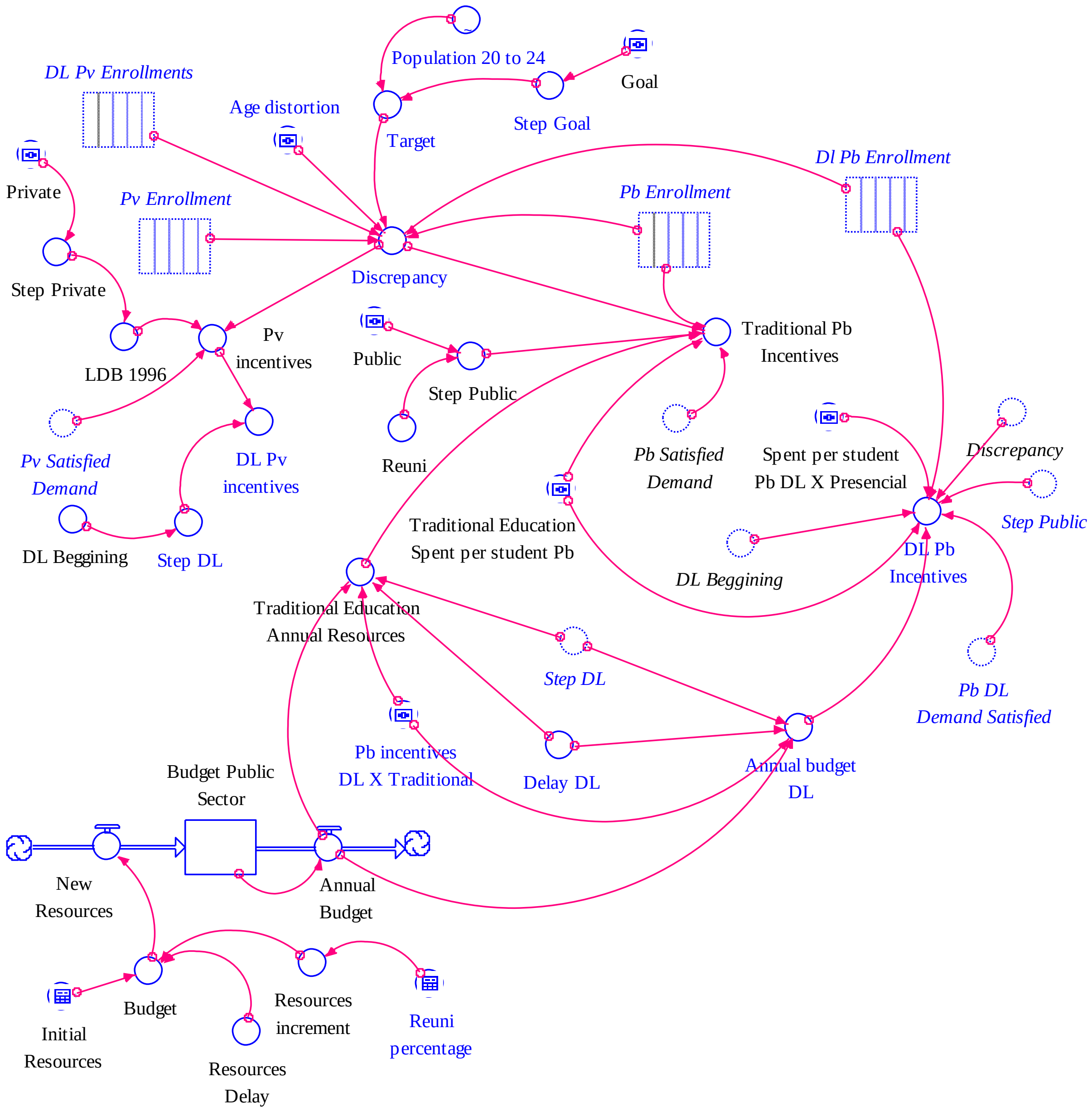


Figure 3 – Policies for Higher Education

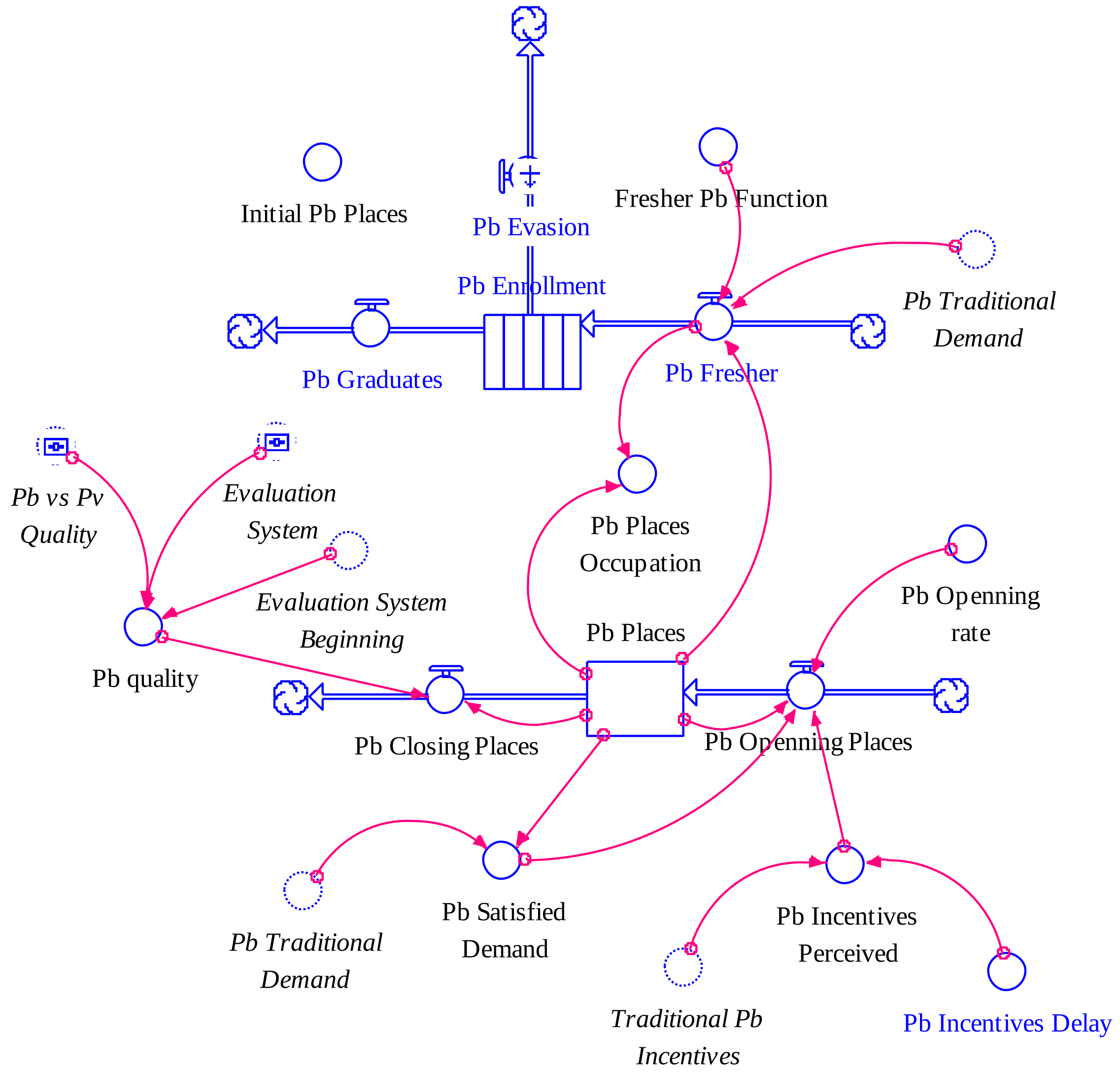
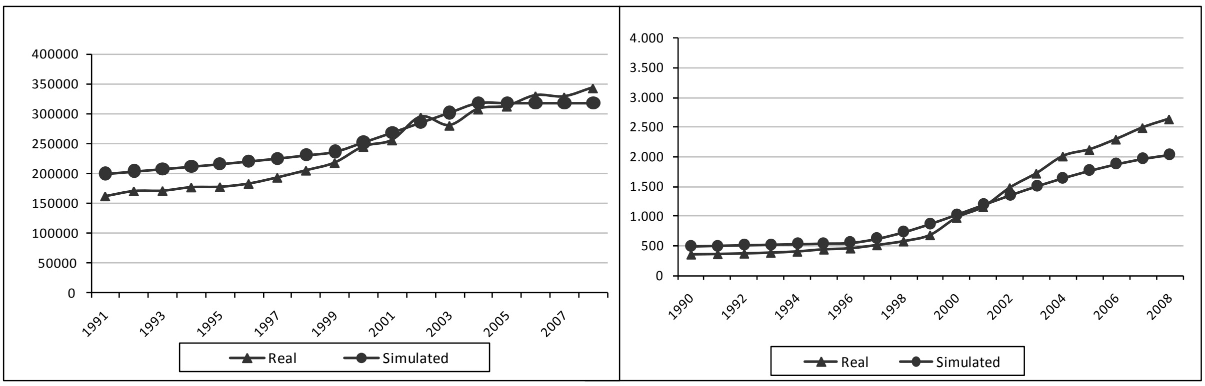


Figure 4 – Public Sector



a) Public Sector b) Private Sector

Figure 5 – Comparison and real data and simulation results - enrolments

4. CONCLUSIONS

The major motivation for this study was related to the importance that education has for the economic and social development of a country and the difficulties faced by Brazil in achieving its goals in the higher education system. The objective was to develop a learning model using system dynamics, which could help managers of higher education to analyze and understand the dynamics of undergraduate higher education.

The simulation developed in iThink© is mainly a prototype. Several elements, for the sake of simplification were not incorporated in the model. Some are due to the inability to establish all the relationships and links between variables in the real system. Others deal with the lack of formal measurement of several indicators, such as pent-up demand, prices charged by the private sector and the attraction between the public and private sector. At the moment, the model is in validation process after being totally verified. However, it is possible to confirm that the use of system dynamics has offered a useful and flexible learning tool to understand the very complex dynamic behavior of the Brazilian higher education system.

Future research will be directed towards the expansion of the model through the refinement of some assumptions and limitations, particularly to investigate and implement: pent-up demand, perceived attractiveness of the public sector compared to the private, tuition prices in the private sector, perceived attractiveness of distance education the classroom, and the encouragement of distance learning in the public sector. It is also suggested to expand the model to incorporate internal efficiency performance indicators, such as spending per student, tax evasion and other quality indicators established by the Ministry of Education. Finally, it would be interesting to differentiate between different types of high education institutions such as universities, colleges, and polytechnics.

References

Carneiro, M. A. 1988. *LDB Fácil*. 3rd edition. Vozes, Petrópolis. In Portuguese.

Carvalho, M. A. *A System Dynamics Analysis of the Higher Level Educational System in Brazil*. 2001. 47 f. Thesis, Rockefeller College of Public Administration and Policy, State University of New York at Albany – SUNY/Albany, Albany.

Dalvi, C. *et al*. 2005. *Análise setorial do ensino superior privado no Brasil*: tendências e perspectivas 2005-2010. Hoper, Brazil. In Portuguese.

Dias Sobrinho, J., Brito, M. R. F. 2008. "La educación Superior en Brasil: principales tendencias y desafios". *Avaliação*, v.13, n.2 (Jul 2008):487-507. In Spanish.

INEP National Institute of Educational Research. 12 Apr 2009. www.inep.gov.br/superior/censosuperior/sinopse. In Portuguese.

Ka-Ho Mok, J. 2002. "From nationalization to marketization". *Governance: An International Journal of Policy, Administration, and Institutions*, v.15, n.2:137-159.

McCowan, T. 2007. "Expansion without equity: An analysis of current policy on access to higher education in Brazil". *Higher Education*, 53:579-598.

Mizrahi, S., Mehrez, A. 2002. "Managing quality in higher education systems via minimal quality requirements: signaling and control". *Economics of Education Review*, 21: 53–62.

Niculescu, M. 2006. "Strategic positioning in Romanian higher education". *Journal of Organizational Change Management*, v.19, n.6: 725-737.

Porto, C., Régnier, K. 2003. *O Ensino Superior no Mundo e no Brasil : Condicionantes, Tendências e Cenários para o Horizonte 2003-2025*. 28 Jan 2009. http://www.macroplan.com.br

Schwartzman, J., Schwartzman, S. 2002. *O ensino superior privado como setor econômico*. 28 Jan 2009.

http://www.schwartzman.org.br/sitesimon/?page_id=546&lang=pt-br. In Portuguese.

Senge, P. M. 2006. *The Fifth Discipline: The Art and Practice of the Organization that Learn*, Revised Edition, Crown Business Edition, Random House, New York.

Shoham, S., Perry, M. 2009. "Knowledge management as a mechanism for technological and organizational change management in Israeli universities". *Higher Education*, 57:227–246.

Sterman, J. D. 2000. *Business dynamics: systems thinking and modeling for a complex world*, McGraw-Hill, Boston.

Strauss, L. M. 2010. *Um modelo em Dinâmica de Sistemas para o Ensino Superior*. M.Sc. Dissertation, Management School, Federal University of Rio Grande do Sul, Porto Alegre, Brazil. In Portuguese.

Scenario Modelling for Managers: A System Dynamics Approach

Robert Y. Cavana
Victoria Management School, Victoria University of Wellington,
PO Box 600, Wellington, New Zealand
bob.cavana@vuw.ac.nz

Abstract

This paper presents an outline of scenario planning and some guidelines on developing and using scenarios. This is followed by a brief overview of scenario modelling. An example of scenario modelling is provided based on system dynamics. The system dynamics scenarios involve dynamic simulation modelling of business policy issues for a strategic business unit in a large organisation in the telecommunications industry in New Zealand. The data and situation are confidential but the case illustrates the scenario modelling process based on the approach developed by Paul Schoemaker. The scenarios in the paper are illustrated with the use of a Management Flight Simulator, developed with system dynamics software. Finally some reflections of the role of system dynamics in scenario planning, modelling and analysis are provided.

Key words: Scenario planning, scenario analysis, system dynamics, simulation, management flight simulator, scenario modelling.

1 Introduction

The term 'scenario' has entered the everyday language of managers in all sectors of the economy and it has been heavily popularised by the media. In earlier times, a scenario was a term more related to the theatrical scene than the business world. The Oxford Dictionary (1995) defines a scenario as "**1** an outline of the plot of a play, film, opera, etc., with details of the scenes, situations, etc. **2** a postulated sequence of imagined events." The author Jules Verne wrote futuristic stories (i.e. scenarios) of human beings travelling to the moon and into the depths of the ocean, well before such travel became possible. Similarly, social commentators such as Aldous Huxley and George Orwell wrote stories (scenarios) about the future state of society, with the intention of warning people how things could turn out if society developed along 'undesirable' paths.

During World War II, scenario planning became popular with military strategists analysing potential deployment of military resources, personnel and weapons. Scenario planning was further popularised in the 1950s by Herman Kahn, a well known futurist from the Rand Corporation and Hudson Institute. "Kahn was best known for his scenarios about nuclear war, in which he advocated that people should "think about the unthinkable" so that, if nuclear war did become imminent, society would be less vulnerable and less likely to slide into a holocaust" (De Geus, 1998, p57).

More recently, scenario planning was introduced to the business world by business planners at the Royal Dutch Shell Group. In the late 1960s and early 1970s, planners at Shell developed a process of preparing a range of stories about future potential states of the business environment, and communicating these stories and their implications to management within the Shell Group. This enabled Shell's management to be better prepared for the 1973 oil crisis than the other international oil companies. Also, scenario planning provided Shell management with substantial competitive insights again in 1981, "when other oil companies stockpiled reserves in the aftermath of the outbreak of the Iran-Iraq war, Shell sold off its excess before the glut became a reality and prices collapsed." (Wack,1995).

Scenario planning should not be confused with forecasting, or single path projections. Although forecasting does have its place in managerial decision analysis, history is littered with 'forecasts' from famous people and experts that have proved to be totally wrong. Cerf and Navasky (1984) summarise some of these:

'Heavier-than-air flying machines are impossible.'
Lord Kelvin, British mathematician, physicist, and president of the British Royal Society, c. 1895

'There is no likelihood man can ever tap the power of the atom.'
Robert Millikan, Physics Nobel Prize, 1923

'Who the hell wants to hear actors talk?'
Harry M. Warner, Warner Bros., 1927

'A severe depression like that of 1920–1921 is outside the range of probability.'
The Harvard Economic Society, 16 November 1929

'I think there is a world market for about five computers.'
Thomas J. Watson, chairman of IBM, 1943

'There is no reason for any individual to have a computer in their home.'
Ken Olson, president, Digital Equipment Corporation, 1977

'We don't like their sound. Groups of guitars are on the way out.'
Decca Recording Co. Executive, turning down the Beatles in 1962

These 'forecasts' almost provide sufficient reason to consider multiple futures! Scenario planning provides a framework to help managers understand the forces driving their businesses, rather than relying on forecasts presented to them with a hidden set of assumptions and judgements incorporated into a set of figures that become a substitute for thinking about the future. In short, scenario planning attempts to capture the richness and range of future possibilities, stimulating decision makers and managers to consider changes they would otherwise ignore. At the same time, it organises those possibilities into stories that are easier to grasp and use than huge volumes of data. Above all, however, scenarios are aimed at challenging managers' mental models and their prevailing mindsets.

In particular, organisations that face the following conditions will benefit from scenario planning (Schoemaker, 1995, p27):

- "uncertainty is high relative to managers' ability to predict or adjust;
- too many costly surprises have occurred in the past;

- the company does not perceive or generate new opportunities;
- the quality of strategic thinking is low (i.e. too routinised or bureaucratic);
- the industry has experienced significant changes or is about to experience such change;
- the company wants a common language and framework, without stifling diversity;
- there are strong differences of opinion, with multiple opinions having merit;
- competitors are using scenario planning."

A scenario is not a forecast or an intention to describe a certain future state, but it is intended to provide a possible set of future conditions. As Becker (1983, p96) outlines:
"A scenario can present future conditions in two different ways. It can describe a snapshot in time, that is, conditions at some particular instant in the future. Alternatively, a scenario can describe the evolution of events from now to some point of time in the future. In other words, it can present a "future history". This latter approach is generally preferred by those engaged in policy analysis and choosing strategy, because it provides cause-and-effect information. Indeed, preparing scenarios as a future history requires that a possible evolution of events and trends be described as an integral part of the scenario".

2 Methodology

2.1 Scenario construction

Pierre Wack (1985, p140) stresses that "scenarios must help decision makers develop their own feel for the future of the system, the forces at work within it, the uncertainties that underlie the alternative scenarios and the concepts useful for interpreting key data."

The main building blocks for constructing scenarios are illustrated in Figure 1 and more detailed steps in scenario construction are provided in Table 1. It should be noted that there are many alternative ways of constructing scenarios (eg see Schwartz, 1996; van den Heijden, 1997) but the method outlined here appears the most consistent with the subsequent use of simulation modelling for testing the assumptions, internal consistency and future implications of the scenarios.

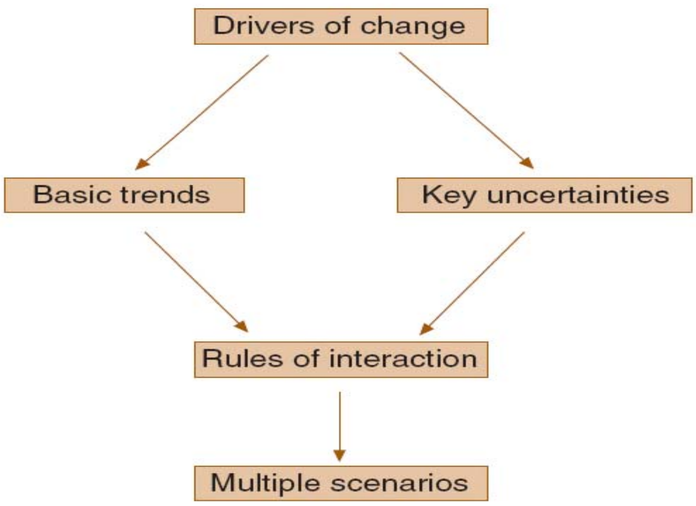


Figure 1. Building blocks for scenarios (Schoemaker, 1995)

Table 1. Steps in scenario construction

1	Define the issues you wish to understand better in terms of time frame, scope and decision variables… Review the past to get a feel for degrees of uncertainty and volatility.
2	Identify the major stakeholders or actors who would have an interest in these issues, both those who may be affected by it and those who could influence matters appreciably (*perhaps prepare a stakeholder map*). Identify their current roles, interests and power positions.
3	Make a list of current trends or predetermined elements that will affect the variable(s) of interest. Briefly explain each, including how and why it exerts an influence. Constructing a diagram may be helpful to show interlinkages and causal relationships (*e.g. a causal loop diagram*).
4	Identify key uncertainties whose resolution will significantly affect the variables of interest to you. Briefly explain why these uncertain events matter, as well as how they interrelate.
5	Construct two forced scenarios by placing all positive outcomes of key uncertainties in one scenario and all negative outcomes in the other. Add selected trends and predetermined elements to these extreme scenarios.
6	Next assess the internal consistency and plausibility of these artificial scenarios. Identify where and why these forced scenarios may be internally inconsistent (in terms of trends and outcome combinations).
7	Eliminate combinations that are not credible or are impossible, and create new scenarios (two or more) until you have achieved internal inconsistency. Make sure these new scenarios bracket a wide range of outcomes.
8	Assess the revised scenarios in terms of how the key stakeholders would behave in them. Where appropriate, identify topics for further study that would provide stronger support for your scenarios, or might lead to revisions of these learning scenarios.
9	After completing additional research, re-examine the internal consistencies of the learning scenarios and assess whether certain interactions should be formalised via a quantitative model… (*such as a system dynamics simulation model*).
10	Finally, reassess the ranges of uncertainty of the dependent (i.e. target) variables of interest, and retrace Steps 1 through 9 to arrive at decision scenarios that might be given to others to enhance their decision making under uncertainty (*or used to test strategies and generate new ideas*).

(Source: Schoemaker, 1993, p97. Words added in italics.)

2.2 System dynamics

The general modelling approach used in this paper to illustrate the Schoemaker approach to scenario construction and analysis is system dynamics. The main phases are summarised in Table 2, following the five phase integrated approach outlined by Maani & Cavana (2007). For further details of the general approach to system dynamics, see for example Forrester (1961), Coyle (1996) or Sterman (2000).

Table 2. Systems Thinking, System Dynamics Methodology

Phases	
1	Problem Structuring
2	Causal Loop Modelling
3	Dynamic Modelling
4	Scenario Planning and Modelling
5	Implementation & Organisational Learning

Source: Maani & Cavana (2007).

The main focus of this paper is on the application of phase 4 'scenario planning and modelling'. In this phase, various policies and strategies are formulated and tested. Here 'policy' refers to changes to a single internal variable such as hiring, quality, or price. Strategy is the combination of a set of polices and as such deals with *internal* or *controllable* changes. When these strategies are tested under varying *external* conditions, this is referred to as scenario modelling. This stage involves working closely with all major stakeholders. Chapter 5 of Maani & Cavana (2007) outlines the scenario planning and modelling approach more fully.

2.3 Scenario modelling

The process outlined by Schoemaker (1993, p197) for developing scenarios is summarised in Table 2. A dynamic simulation model is most useful during stages 6, 9 and 10 of the process, i.e. to test the internal consistency and consequences of a range of scenarios. In addition, a dynamic simulation model can be used to design and analyse the implications of policies and strategies against the backdrop of the scenarios developed. In the next section, we briefly outline a case study involving scenario modelling and strategy development for a business unit in the telecommunications industry in New Zealand.

3 Case: scenario analysis for a telecommunications business unit

3.1 Case overview

This case is based on a consultancy project for a business unit of a telecommunications company operating in a small city in New Zealand (Cavana & Hughes, 1995). However, the issues, data and names have been changed to preserve client confidentiality. The major issues dealt with are how to design policies, and test strategies to help managers turn around a business unit that is experiencing a declining market share and eroded profitability (see Figure 2).

The operations of the telecommunications business unit (called TBU for the purposes of this case) include replacing, maintaining and installing telecommunications lines and connections to commercial and residential sites in the small city in which it works. The processes involve the design and planning of these connections, as well as undertaking the work to deliver the services. These are known as 'jobs' in the industry and we will refer to the whole process from design to delivery as a 'job' in this case.

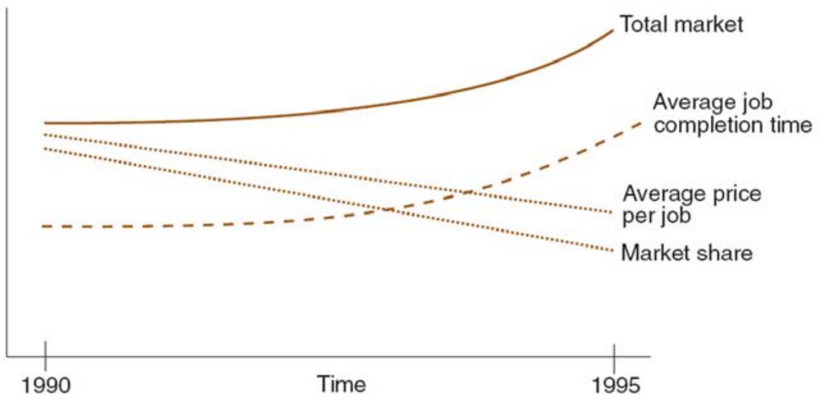


Figure 2. Behaviour over time chart for the TBU strategy case

After a number of meetings with the project team (two consultants, the finance manager and CEO of the TBU, and two members of the head office (HO) strategy support staff), a causal loop diagram (CLD) was developed. This development was iterative, and a whiteboard was used. This CLD became the basis for the development of the telecommunications business unit (TBU) strategy model, using the *ithink* computer simulation software (Richmond & Petersen, 1997). The model contains four sectors: a market sector which includes variables for price, quality, market share, and the total market; an operations sector that deals with the processing of the services provided by the business unit, and relative productivity factors; a human resources sector that incorporates natural attrition, replacement and hiring decisions; and a finance sector that calculates costs, revenues, profit margins, cash flow and net present value for the business unit over a five-year period.

The case is outlined in Cavana & Hughes (1995) and fully discussed in Maani & Cavana (2007, Case 5, pp.225-270). The documented equations are provided, and an electronic version of the *ithink* and Vensim versions of the model are provided in the CD-rom accompanying the book (Maani & Cavana, 2007).

3.2 Model development

The analysis was done on a monthly basis as records and data were available on that basis, and the planning horizon (or length of the simulation run) was five years (60 months). This was TBU's usual planning horizon for strategic analysis and planning. Data were collected from interviews with TBU and HO staff, and where data were not readily available, estimates were made – typically in collaboration with, and with the endorsement and/or approval of, the financial manager and CEO of TBU. An example of one of the model sectors is provided in Figure 3.

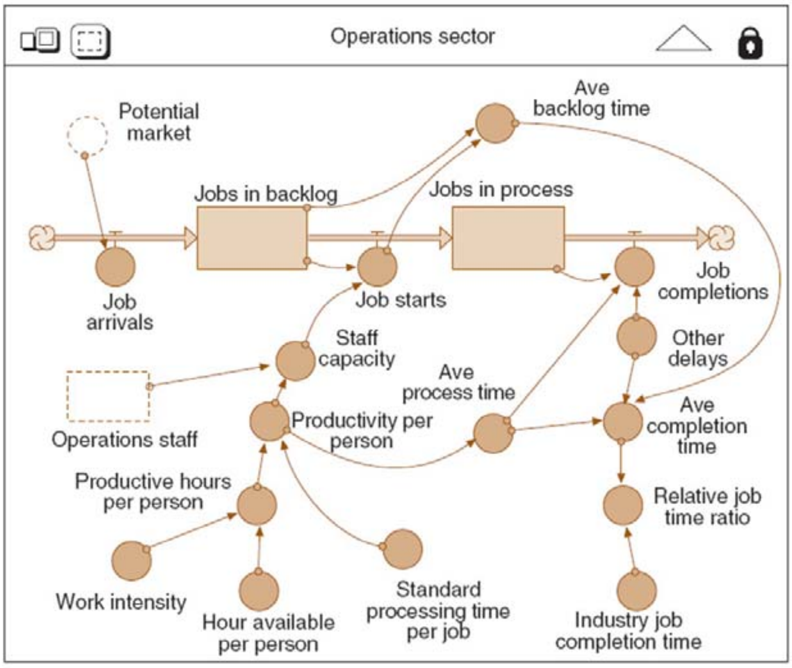


Figure 3. Stock flow diagram for the TBU operations sector

3.3 Base case behaviour

The base case model behaviour for some of the main variables is summarised in Figure 4.

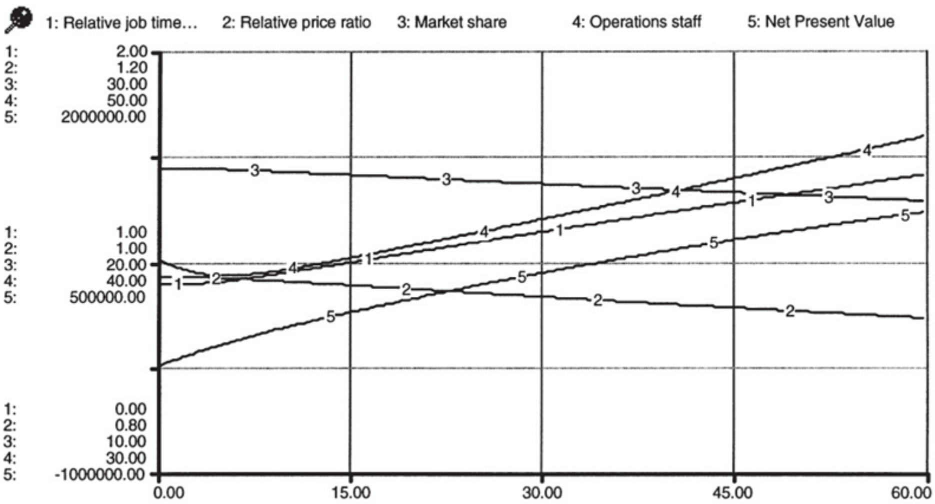


Figure 4. Base case behaviour of the TBU strategy model

The detailed model validation, sensitivity testing, policy analysis and strategy development was facilitated by the construction of a user friendly Management Flight Simulator for the model. An example of a simple control panel for the market sector input parameters, graphs and main outputs is provided in Figure 5.

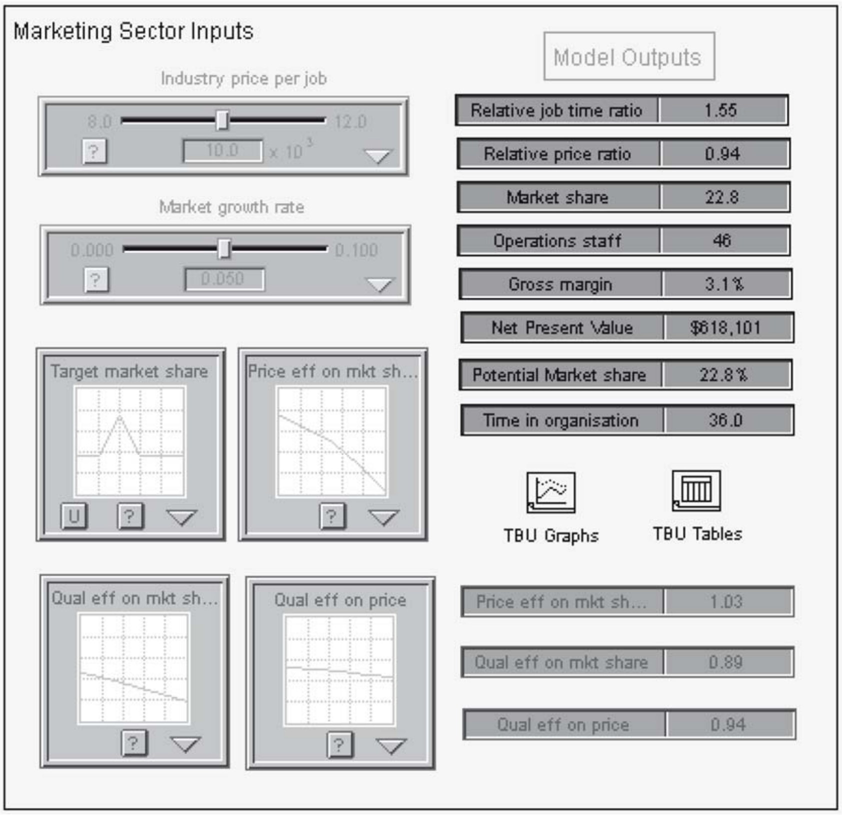


Figure 5. A simple control panel for the market sector of the TBU strategy model

4 Scenario analysis

In this section we present a range of scenarios for the TBU strategy case and then test the redesigned strategies against these alternative futures (i.e. scenarios). Initially we indicate the scope of the major uncertainties that could have an impact on the TBU, then we present two 'forced' scenarios. These form the basis for the 'learning' scenarios, which we analyse with the dynamic simulation model to test the robustness of our redesigned strategies.

4.1 Uncertainties

There are a large number of uncertainties that could have an impact on the future of the TBU. These include the factors summarised in Table 3.

Table 3. Factors potentially affecting the future of TBU

Market factors

- Market – stagnation and decline, slow growth, rapid growth
- Competitors – major new competitors, some competitors departing
- Technological changes in market – nature of telecommunications greatly different, remains the same
- Prices – remain constant in real terms, declining

Operations factors

- Operations – big changes in delays (logistic, workers, regulatory authorities, etc.) – improvements and deterioration in relationships causing increased or decreased delays.
- Technological changes in equipment used – no change, substantial changes
- Equipment – continue leasing from market, purchase own equipment

However, to undertake scenario analysis, we cannot consider every possible change that may take place in the future. Instead, we need to identify the changes that are likely to happen, or could have the biggest impact on the organisation (i.e. the TBU). Also we need to consider the variables, parameters and factors that are most uncertain, or are most important to the organisation.

4.2 Forced scenarios

Having identified a range of uncertainties and factors that could have an impact on the TBU, the project team then developed the forced scenarios which are summarised in Table 4. The base case was taken as the 'surprise-free' scenario, and all the negative external parameter changes were 'dumped' into the 'pessimistic' scenario and all the positive external changes to the parameters were 'dumped' into the 'optimistic' scenario.

The 'surprise-free' scenario is the one based on the existing model parameters that TBU management expected would continue into the future (i.e. the 'business as usual' scenario). These forced scenarios then needed to be checked for internal consistency and some extra research was necessary to determine realistic boundary values for the parameters and graphical relationships, so that 'learning' scenarios could be developed.

Table 4. Forced scenarios for the TBU strategy case

	Forced scenarios		
	Major influence	Pessimistic	Optimistic
Market sector			
Industry price per job	External	-15%	+10%
Market growth rate	External	0%	+10%
Target market share	Internal	Decline to 20%	Increase to 30%
Price effect on market share	External	Deteriorate by 20%	Improve by 10%
Quality effect on market share	External	Deteriorate by 25%	Improve by 10%
Quality effect on price	External	Deteriorate by 20%	Improve by 10%
Operations sector			
Industry job completion time	External	-15%	+10%
Other delays	Both	+25%	-20%

We discussed these forced scenarios with TBU management and realised that there were some inconsistencies and some assumptions that were most unlikely to occur, or would affect the whole industry so the TBU would be no better off. This led to the development of the 'learning scenarios' against which the new redesigned policies and strategies could be tested.

4.3 Learning scenarios

A number of the combinations included under the positive and pessimistic scenarios were not logically consistent. For example, the assumption that costs would decrease by between 10 and 20%, while the market is growing at 10% p.a., is probably not consistent with economic logic. An increase in demand for the final product or service is likely to result in an increase in the demand for the factor inputs and cause costs to rise rather than fall, unless there are some other compensating changes.

Similarly, TBU management was of the opinion that it was unlikely there would be any significant changes in the tax rate over the next five years, and most of the costs were expected to remain at about the same level. Some of the ranges for the parameter changes shown in the forced scenarios were also considered rather extreme.

Finally we generated some 'learning scenarios' for testing our strategies for the TBU. These are summarised in Table 5. We considered three scenarios: base case (i.e. 'business as usual'); 'bleak outlook' (a combination of 'consistent' negative parameters in the external environment); and a 'rosy picture' scenario (based on a feasible combination of positive factors that could occur in the external environment).

Table 5. Learning scenarios for the TBU strategy case

External parameter or behavioural relationship	Base case scenario	'Bleak outlook' scenario	'Rosy picture' scenario
Industry price per job	$10 000	$9500	$10 300
Market growth rate	5% p.a.	2% p.a.	8% p.a.
Price effect on market share	1.5/1.25/1/0.6/0.1	1.2/1.1/1/0.4/0.05	1.6/1.3/1/0.7/0.2
Quality effect on market share	1.2/1.1/1/0.9/0.8	1.1/1.05/1/0.8/0.6	1.3/1.15/1/0.95/0.9
Quality effect on price	1.1/1.05/1/0.95/0.9	1.05/1.025/1/0.8/0.6	1.2/1.1/1/0.975/0.95
Industry job completion time	2.5 mths	2.2 mths	2.8 mths
Time in organisation[10] (mths)	60/48/36/20/0	45/36/27/15/0	75/60/45/25/0

We inputted each of these scenarios into the model using the control panel features of the *ithink* computer simulation software (eg see Figure 5), and tested each strategy against the scenarios. The results of some of these scenario tests are presented in Table 6.

Table 6. Selected results of the 'learning' scenario tests

Strategy	Base case scenario	'Bleak outlook' scenario	'Rosy picture' scenario
Relative job time ratio			
1 'Base case'	1.17	1.33	1.04
2 'Tightening up ops'	1.05	1.20	0.94
Relative price ratio			
1 'Base case'	0.98	0.87	1.00
2 'Tightening up ops'	0.99	0.92	1.01
Market share (%)			
1 'Base case'	23.7	22.1	23.8
2 'Tightening up ops'	24.1	23.1	24.2
Net present value ($m)			
1 'Base case'	1.12	0	2.35
2 'Tightening up ops'	1.53	0.30	2.87

5 Conclusions

This analysis indicates strategy 2 (tightening up operations) is robust, when the alternative plausible future scenarios are taken into consideration. Strategy 2 performs better than the base case strategy under each of the performance measures, i.e. lower relative job times; higher relative price ratio; greater market share; and higher net present value. Overall, strategy 2 would help TBU's management address the strategic issues that were outlined at the beginning of this case.

The system dynamics and scenario modelling approach outlined in this paper provided considerable insights and learning for the whole TBU project team, including the managers from the TBU and the strategy support staff from the head office holding company. In addition, the paper demonstrates the value of utilising Schoemaker's (1993, 1995) scenario construction process with system dynamics modelling.

6 References

Becker, H.S. 1983. "Scenarios: a tool of growing importance to policy analysts in government and industry." *Technological Forecasting and Social Change*, **23**: 95-120.

Cavana, R.Y. and R.D. Hughes. 1995. "Strategic Modelling for Competitive Advantage." *Proceedings of the 1995 International System Dynamics Conference*. Gakushuin University, Tokyo, Japan. July 30 – Aug 4, Vol II: 408–417.

Cerf, C. and V. Navasky. 1984. *The experts speak*. Pantheon, New York.

Coyle, R.G. 1996. *System Dynamics Modelling: A Practical Approach*. Chapman and Hall, London.

De Geus, A. 1998. *The Living Company*. Nicholas Brearly, London.

Forrester, J.W. 1961. *Industrial Dynamics*. MIT Press, Cambridge, MA.

Maani, K.E. and R.Y. Cavana. 2007. *Systems Thinking, System Dynamics: Managing Change and Complexity*. 2nd ed. Pearson Education New Zealand, Auckland.

Oxford Dictionary. 1995. *The Concise Oxford Dictionary of Current English*, 9th ed. Oxford University Press, Oxford.

Richmond, B. and S. Peterson. 1997. *An Introduction to Systems Thinking.* High Performance Systems, Hanover.

Schoemaker, P.J.H. 1993. "Multiple scenario development: its conceptual and behavioural foundation." *Strategic Management Journal*, **14**: 193-213.

Schoemaker, P.J.H. 1995. "Scenario planning: a tool for strategic thinking." *Sloan Management Review*, **36**:25-40.

Schwartz, P. 1996. *The Art of the Long View*. Currency Doubleday, New York.

Sterman, J.D. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Irwin McGraw-Hill: Boston.

van den Heijden, K. 1997. *Scenarios: The Art of Strategic Conversation*, Wiley, Chichester

Wack, P. 1985. "Scenarios: shooting the rapids." *Harvard Business Review*, **63**: 131–142.

Comparing Public and Private Sector Decision Making: Problem Structuring and Information Quality Issues

Stuart Dillon
John Buchanan
Jim Corner
Department of Management Systems
University of Waikato
New Zealand
jtb@waikato.ac.nz

Abstract

Generally speaking managerial decision making can be enhanced through providing appropriate decision support. The nature of that support depends on a number of variables, including the organisational context of the decision (public vs. private sector), the quality of information available and, of course, the willingness of decision makers to utilise such support. Anecdotal evidence suggests that since public and private sector decision makers operate in different decision making contexts and employ different decision processes, their support needs differ. This mixed-methodology empirical study seeks to isolate key decision processes as well as attitudes towards information quality of senior managers within public and private sectors in New Zealand. Results clearly indicate differences in the sectors; particularly in terms of how decisions are structured and how decision makers view the timeliness and relevance of the information they receive. This has implications for the provision of support for decision makers in these two sectors.

Key words: Decision making, information quality, public sector, private sector.

1. Introduction

Decision making is the most common task of managers and executives. "Successful [organisations] 'outdecide' their competitors in at least three ways: they make better decisions; they make decisions faster; and they implement decisions more" (McLauglin 1995:443). While most would agree with this, much less is known about what 'makes' good or bad decisions, although it is generally accepted that a good decision process should give rise to good decision outcomes (e.g. Henig &Buchanan 1996).

To begin we first must understand the basic decision making process. Simon (1960) proposed a three-phase model of the decision process comprising the activities of Intelligence, Design and Choice. Intelligence involves identifying the need for a decision. Design begins when a decision need is identified and concludes when a choice is ready to be made. Weick (1979) conceptualises the design phase as moving from an unworkable version of reality to a workable version of reality. The final choice phase describes the activity of selecting the most appropriate course of action among available alternatives.

The early literature on decision making focused on the choice phase and was essentially normative in its orientation, prescribing a 'best' way to decide. Decision Analysis (Keeney 1982) and the Analytic Hierarchy Process (Saaty 1980) are two well-known examples of this approach. Such approaches tend to adopt a 'one size fits all' approach, to make unreasonable assumptions about decision maker rationality and are not always cognizant of the different contexts (or environments) which constrain the decision making process. The impact of these constraints is such that many (e.g. Nutt 1993) have rejected the more normative approaches for decision support, and have turned their attention to descriptive approaches which take account of these real-world constraints. Rather than requiring decision makers to behave in certain ways (prescription), descriptive approaches start with how decisions are actually made and from there seek to improve these processes (Dillon 2002).

This study compares decision making for senior managers in the different contexts of the private and public sectors.

2. Literature

2.1 Information Quality

There are four commonly accepted dimensions of information quality: intrinsic, contextual, representational and accessibility (Wang &Strong 1996). Within these four dimensions many different measures of information quality have been introduced, including accuracy and objectivity (intrinsic dimension); relevance and timeliness (contextual dimension); and interpretability and ease of understanding (representational dimension). Access is the principal measure for the accessibility dimension.

In this paper we compare attitudes of upper level managers in the public and private sectors to the three measures of information quality; accuracy, relevance and timeliness, which are from Wang and Strong's (1996) intrinsic and contextual dimensions.

2.2 Decision Structuring

As noted earlier, the decision structuring process is more or less equivalent to the design phase of Simon's three-phase model and is possibly the most important part of the entire decision process; (see for instance Abualsamh, *et al.* 1990;Mintzberg, *et al.* 1976;Perry &Moffat 1997;von Winterfeldt 1980). This importance is not surprising; if, for instance, a decision is structured where some genuinely 'good' alternatives are not considered or if some important goals are missing, decision quality is likely to be compromised.

There are many definitions of decision structuring in the literature. Dillon (2002) has proposed a simple, non-directive, process-based definition: *Decision structuring is the process by which a decision situation is transformed into a form enabling choice*.

While decision structuring has received far less attention than other phases of the decision process (especially choice), a number of significant contributions have been made over the past three decades. Keeney (1982) was one of the first to incorporate specific problem structuring activities within the (prescriptive) decision analysis process. Contained within Keeney's structuring phase are the generation of proposed alternatives and the specification of objectives or goals. Nutt (1993) sought to characterise the various types of problem formulation (decision structuring) that occur in practice. With data from 163 decisions he identified four structuring processes; idea, issue, objective-directed and reframing. Nutt found some processes to be more effective than others, with a strong inverse correlation between efficacy and frequency of use.

2.3 Public and Private Sector

Private sector motivations generally differ from those of the public sector; which is not surprising given that they have different environments or contexts. The private sector is typically associated with market forces while the public sector is more noticeably shaped by political considerations; one is about 'business' and the other 'government'; one tends to be decentralised and the other centralised (Perry &Rainey 1988).

These different environments imply different decision content. Bozeman and Pandey (2004) distinguish between the two poles of technical and political content where technical aligns with the concepts of efficiency and effectiveness, where there is general agreement about ends (or goals). This is in contrast with political content where disagreement about ends is the norm. Nutt (2006) has compared public and private sector decision making using the metrics of analysis and bargaining and found that private sector managers are more supportive of analysis-based decisions and public sector managers are more supportive of bargaining-based decisions. It would seem that the public sector is more problem-based, while the private sector is more opportunity-based. In general the conclusions of Bozeman and Pandey (2004) are supported; that decision content is a significant determinant of decision process and by implication one should expect to observe different decision processes in the public and private sectors.

3. Method

Our investigation into the impact of information quality and decision structuring is a mixed-methodology study (e.g. Mingers 2001), using an in-depth case study for decision structuring and a separate large scale survey for information quality.

3.1 Decision Structuring Case Studies

Sixteen senior managers representing a diverse range of public and private sector organisations were interviewed. Each was interviewed in depth on the nature of the decisions they make, the manner in which they make these decisions, and what they considered to influence their behaviour. Participants also agreed to complete a questionnaire so that cross-participant comparison and interview response validation could be performed. Following data collection (and confirmation with participants that this had been interpreted correctly), raw data from all sources was analysed using an adaptation of the data analysis aspects of the grounded theory approach (e.g. Glaser &Strauss 1967). This adaptation occurred such that the procedures best match the subject of the study and the theoretical underpinnings of the research and the researcher. Successful implementation of the grounded theory approach permits such adoption (Strauss &Corbin 1990).The methodology is fully explained in Dillon (2002).

3.2 Information Quality Survey

The data on information quality comes from a larger international study of the impact of culture on business decision making and information use. The original study for New Zealand was undertaken in 2002 using a survey method. From the Kompass Products and Services of New Zealand Companies information database, three distinct industries were selected in order to achieve a balanced cross-section of the population: manufacturing, service and government. The sample population was further stratified by organisation size, with organisations having a greater number of employees selected to ensure a sufficient number of respondents at three levels of management authority (described as highest, middle or lower). A total of 450 organisations were contacted, 150 from each sector. Each of these 450 organisations received six questionnaires, two questionnaires for each of the three levels of management, resulting in a total of 2,800 distributed questionnaires. A total of 818 usable responses were returned, providing a response rate of 29%.

The analysis was limited to managers and executives which reduced the sample size to 466, evenly split between public and private sector managers. Further detail is provided in Parslow (2003).

4. Results and Discussion

4.1 Decision Structuring

The following model (Figure 1) of decision structuring emerged from the study. Although the model is essentially the same for both public and private sector senior managers, the nature of and impact on the various process activities are quite different.

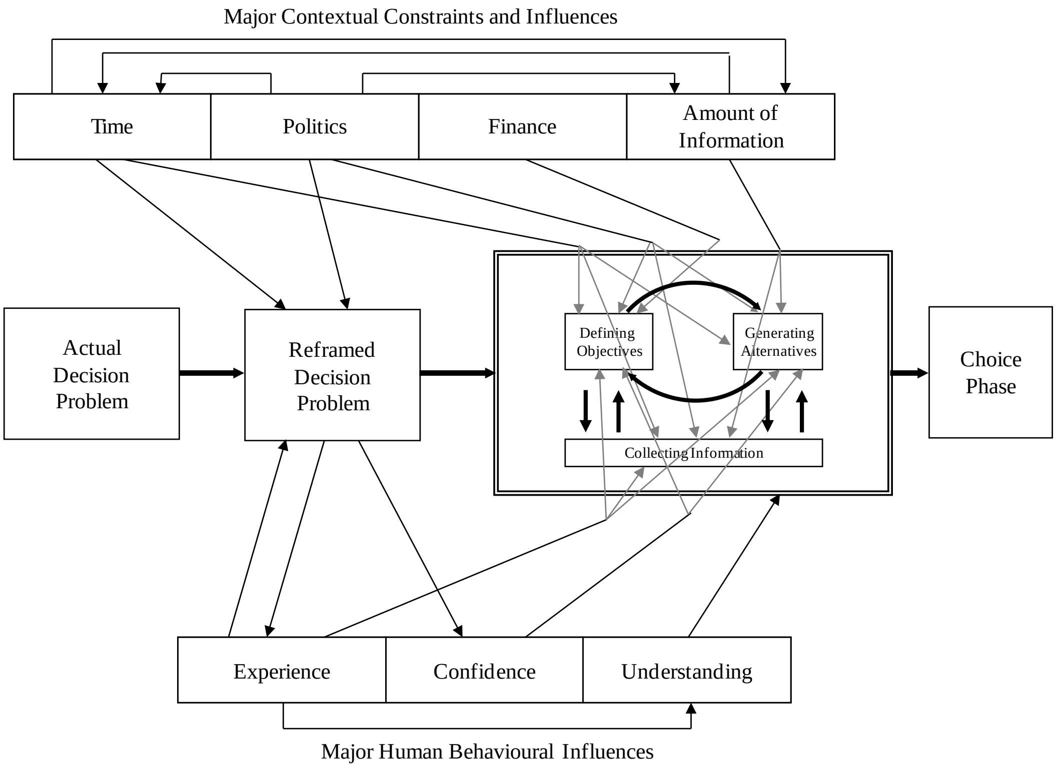


Figure 1: An Influence Model of Decision Making

There are a few key decision structuring activities common to all decision makers examined in the case studies. However the sequencing of these activities varied. During decision problem structuring, a combination of information gathering, goal definition and alternative generation takes place, in no particular order, and often in a repetitive and cyclical manner. Similar to Nutt's (1993) findings, the more structured decision processes began with defining objectives or goals, whereas others would begin with generating potential alternatives, often without any explicit consideration given to the underlying objectives of the decision problem. Information gathering, while often a stage in its own right, was usually employed to support the activities of objective definition and alternative generation. However, the significance of information in the decision structuring task varied considerably. It appeared that with more information available and/or greater use of information, a senior manager’s reliance on intuition decreased.

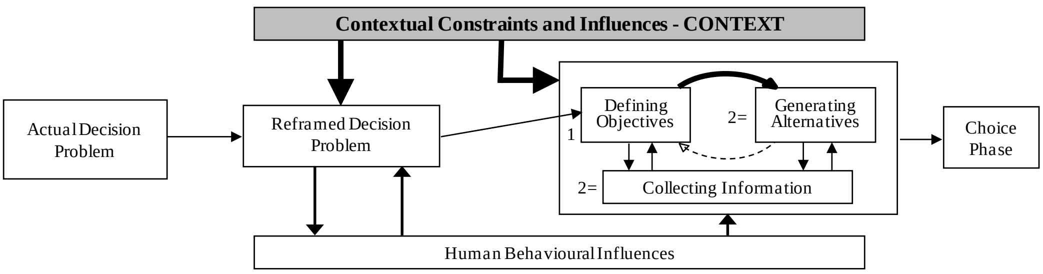
4.1.1 Decision Influences

Decision-maker behaviour was influenced by both external influences and human behavioural factors. External influences (context) included time, politics (both internal to the organisation and external), limited finance and the quantity of information (too much and too little).

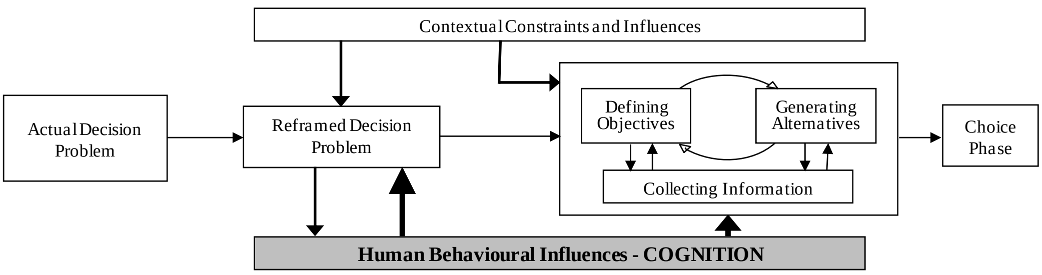
Internal influences (cognition) comprised experience, in terms of the level of experience a senior manager has in making non-trivial decisions, confidence to act in a way that he or she feels is most appropriate, and understanding of the context/ domain of the decision problem.

4.1.2 Contrasting Behaviours

Incorporating the elements of the Figure 1 model, two variations are presented that exhibit the difference between decision structuring in the public and private sectors. These contrasting models of public and private sector decision structuring are shown in Figure 2. Differences are noted by different sized arrows and shading (strength of influence), and numbers reflect the order of problem structuring activities in the public sector model. Such ordering of process was not observed in the private sector.



Public sector



Private sector

Figure 2: Public and Private Sector Models of Decision Structuring

Analysing the two sectors separately suggests that public sector decision making is typically more open, transparent and structured. Unlike the private sector, decisions made in the public arena are open to broad stakeholder scrutiny and therefore decision processes are often prescribed or at least devised prior to the actual decision making. Private sector decision making is generally more ad-hoc and occurs more proactively. As shown in Figure 2, contextual constraints and influences play significant roles in structuring public sector decisions. Because of the constrained nature of decisions made in this environment, human behavioural aspects have much less influence.

Another obvious point of difference with the public sector model is sequentiality of process; only public sector senior managers were able to describe activities that had any form of sequential process. The typical public sector decision making process begins with the definition of objectives or goals (shown as number “1” in Figure 2). Given that disagreement about ends or goals is a feature of the public sector (Bozeman &Pandey 2004), beginning with objective definition is not surprising. Moreover, the public sector requirement for transparency increases the importance of clarity of objectives. Alternatives are generated and information is collected only after objectives have been defined. Alternative generation and information collection are simultaneous activities (shown as “2=” in Figure 2). The nature and frequency of information collection is strongly influenced by the requirements of the sector and the particular decision situation. The typical public sector decision generally emerges from within the organisation, i.e. it is bottom-up, unforeseen, and therefore reactive.

In contrast to the public sector, the contextual aspects of private sector decisions were less influential; several senior managers stated clearly that constraints of limited finances or time would not be allowed to influence their decision making. Conversely, given the relatively unregulated nature of private sector decision making, human behavioural influences played a much greater part in the decision structuring process. Confidence played a major role and overall, private sector decision makers appeared more confident in their decision making ability. Unlike the public sector model, no obvious (sequential) process could be observed; the activities of defining objectives and goals, generating alternatives and collecting information occurred in no particular order. In another significant contrast, the typical private sector decision can be described as top-down, foreseen and proactive, consistent with a more opportunity-based mode of decision making.

An interesting observation relates to the backgrounds of participants. While many had only worked in either the public sector or in the private sector, some had worked in both sectors. Senior managers who had worked in both often stated that they experienced significant difficulties when working in one of these sectors. Those who had developed skills within the public sector found the private sector environment extremely pressurised and unsupportive. Conversely, those with prior private sector experience felt the public sector was overly restrictive and bureaucratic. It appears that the sector in which a manager has developed his or her skills and experience is most likely to be the sector in which they are most suited and most successful.

4.2 Information Quality

As shown in Figures 1 and 2, information collection and by implication, information quality, is a significant activity in the decision structuring process for both sectors. We now turn attention to the quality of information both desired (ideal) and perceived (actual) for senior managers in the public and private sectors.

Responses from 11 of the 76 questions from the questionnaire were used, comprising five demographic questions and six questions on information quality. These six questions addressed ideal and actual scores using a five-point Likert scale for the measures of information quality. The format for 'ideal' questions was: How important is it to you to: e.g. *...have accurate information?* Responses ranged from 1 – 'of utmost importance to me' to 5 – 'of very little importance to me'. The 'actual' question format was: Now, as compared to what you want, how satisfied are you at present with: e.g. *...the accuracy of the information you receive at work?* Responses ranged from 1 – 'very satisfied' to 5 – 'very dissatisfied'.

Table 1 presents the mean scores for the three measures of information quality: accuracy, timeliness and relevance. These results are analysed in three ways: (1) according to the gap between the ideal and actual information quality, (2) according to the public and private sector responses over the ideal, actual and gap columns, and (3) according to the relative importance of the three information quality measures.

4.2.1 Ideal – Actual Information Quality Gap

The gaps between ideal and actual are all significantly different from zero—for private, public and in total—across all three measures of information quality. Expectations exceed perceptions in every instance, as one would realistically expect.

4.2.2 Sector Responses to Information Quality

In terms of the ideal measures of information quality, both public and private sector senior managers scored them roughly the same—there was no significant difference between any of the three quality measures. Despite the different characteristics of the two sectors, both desired very similar levels of information accuracy, timeliness and relevance.

Info Quality Measure	Sector	Ideal	Actual	Gap
Accuracy	Private	1.44	2.44	1.00
	Public	1.42	2.33	0.90
	Total	**1.43**	**2.38**	**0.95**
Timeliness	Private	1.77	2.60	0.83
	Public	1.87	2.41 *	0.54 **
	Total	**1.82**	**2.51**	**0.69**
Relevance	Private	1.85	2.30	0.44
	Public	1.85	2.09 **	0.24 **
	Total	**1.85**	**2.19**	**0.34**

*p<0.050; **p<0.01, using one-way ANOVA.

Table 1: Mean Information Quality Scores

In terms of actual satisfaction and the gap between ideal and actual, there were significant differences between private and public sector senior managers for both timeliness and relevance, as shown by the astericks in Table 1. For these two measures, public sector senior managers were more satisfied with information quality and their gaps were smaller than their private sector counterparts.

However, for the accuracy measure there is no significant difference between the public and private sectors; they are similarly (dis)satisfied and have similar gaps. Wang and Strong (1996) offer an explanation: it is the contextual measures of timeliness and relevance that differ between sectors, whereas the intrinsic measure of accuracy applies equally to decision makers in each sector. Thus information accuracy appears to be independent of the context (public or private), as expected for an intrinsic quality dimension.

4.2.3 Relative Importance of Information Quality Measures

Additionally, and not shown in the table, a Tukey HSD test shows that the total gaps for each measure of information quality are all significantly different ($p<0.01$), which allows a clear ranking of the three measures from best to worse as: relevance, timeliness, accuracy (ie, 0.95>0.69>0.34). Information accuracy is clearly the most important issue for both sectors.

5. Results and Discussion

This study is largely consistent with previous research in demonstrating that decision makers in the public and private sectors differ in regard to attitudes to information quality and approaches to decision structuring. While it is not a new result that there are significant differences between the public and private sector, this study highlights new dimensions of difference. Common to both sectors is the primary importance of information accuracy, an area where both sectors are dissatisfied with actual measures.

Public sector decisions were characterised as being bottom-up, unforeseen and reactive; decisions are made as and when they need to be rather than being actively sought. The public sector appears to have more structured decision processes as necessitated by the political context. Although this has resulted in a more satisfactory contextual information quality in respect of timeliness and relevance and an apparently high level of analysis, in the end political bargaining appears to be the main determinant of the decision outcome. The challenge, therefore, in the public sector is perhaps for better decision support in politics of bargaining without compromising the value of the information already collected.

Conversely, decision making within the private sector is considered to be an essential enabler of a company’s growth and competitiveness. For this reason, private sector decision making is characterised as top-down, foreseen and proactive. If, then, there is a foreseen aspect to decision making, then it is possible that if private sector managers initiate the information collection process earlier, the timeliness of information should improve.

Decision support, and in particular the decision structuring process, must be more responsive to the dynamic and complex nature of real decision situations. It needs to be accepted that many observed constraints cannot be eliminated; instead consideration needs to given to their management. In addition, less emphasis should be placed on trying to persuade decision-makers to follow a pre-specified process; rather the focus should be on ensuring that good decision making principles are adhered to. These principles include developing a process (at the outset) that will guide the overall decision-making, identifying objectives/goals by which decision alternatives can be measured, being aware how the decision context can affect the decision process, and getting accurate, timely, and relevant information.

There are also recruitment implications. When recruiting, employers should consider the decision making experience of candidates, while candidates should be fully aware of the decision making context within which the organisation operates.

Finally, a necessary condition for good knowledge management is good information management, particularly as the activity of information collection influences the decision structuring process and the all important decision outcome.

6. References

Abualsamh, R. A., Carlin, B. & McDaniel Jr, R. R. 1990, "Problem Structuring Heuristics in Strategic Decision-Making." *Organizational Behavior and Human Decision Processes* **45**: 159-174.

Bozeman, B. & Pandey, S. K. 2004, "Public Management Decision Making: Effects of Decision Content." *Public Administration Review* **64**: 553-565.

Dillon, S. M. 2002, Understanding the Decision Problem Structuring of Executives, Thesis, University of Waikato.

Glaser, B. & Strauss, A. 1967, *The Discovery of Grounded Theory*, Aldine, Chicago.

Henig, M. I. & Buchanan, J. T. 1996, "Solving MCDM problems: Process concepts." *Journal of Multi-Criteria Decision Analysis* **5**: 3-21.

Keeney, R. L. 1982, "Decision Analysis: An Overview." *Operations Research* **30**: 803-838.

McLaughlin, D. J. 1995, "Strengthening Executive Decision-Making." *Human Resource Management* **34**: 443-461.

Mingers, J. 2001, "Combining IS Research Methods: Towards a Pluralist Methodology." *Information Systems Research* **12**: 240-259.

Mintzberg, H., Raisinghani, D. & Theoret, A. 1976, "The Structure of ‘Unstructured' Decision Processes." *Administrative Science Quarterly* **21**: 246-275.

Nutt, P. C. 1993, "The Formulation Processes and Tactics Used in Organizational Decision-Making." *Organization Science* **4**: 226-251.

Nutt, P. C. 2006, "Comparing Public and Private Sector Decision-Making Practices." *J Public Adm Res Theory* **16**: 289-318.

Parslow, K. J. 2003, An Investigation into the Importance of Information for Organisations: A New Zealand Context, 599 Report of an Investigation Thesis, University of Waikato.

Perry, J. L. & Rainey, H. G. 1988, "The Public-Private Distinction in Organization Theory: A Critique and Research Strategy." *The Academy of Management Review* **13**: 182-201.

Perry, W. & Moffat, J. 1997, "Developing Models of Decision-Making." *Journal of the Operational Research Society* **48**: 457-470.

Saaty, T. L. 1980, *The Analytic Hierarchy Process*, McGraw Hill, New York.

Simon, H. A. 1960, *The New Science of Management Decision*, Prentice Hall, N.J.

Strauss, A. & Corbin, J. 1990, *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*, Sage, CA.

von Winterfeldt, D. 1980, "Structuring Decision Problems for Decision Analysis." *Acta Psychologia* **45**: 71-93.

Wang, R. Y. & Strong, D. M. 1996, "Beyond accuracy: what data quality means to data consumers." *J. Manage. Inf. Syst.* **12**: 5-33.

Weick, K. E. 1979, *The Social Psychology of Organizing*, Addison-Wesley, Reading, MA.

The Container Positioning Problem Revisited Yet Again!

David Ryan
Department of Engineering Science
The University of Auckland
New Zealand
d.ryan@auckland.ac.nz

Abstract

In her DTU PhD Thesis in 2008, Louise Sibbesen proposed an optimization based model for solving a version of the container positioning problem as it occurs commonly in container ports. Sibbeson discarded the optimization model as being impractical from a computational point of view and focussed on the development of heuristic methods for the problem. During 2009 in an Honours project in Engineering Science at the University of Auckland, Antony Phillips demonstrated that it was possible to develop solution methods for the optimization model which were far more efficient than predicted by Sibbeson. However Phillips also identified some serious limitations to this approach. During 2010 two Danish Masters students, Jonas Skott Sigtenbjerggaard and Jonas Ahmt have worked with me in Auckland to address these limitations and they have made some important steps towards the development of optimization based methods which we hope one day might be able to solve realistic practical container positioning problems. In this talk we will discuss aspects of the problem and their progress and identify remaining challenges.

Empty Container-truck Movement Problem: At Ports of Auckland

Samsul Islam[1], Tiru Arthanari[2], Tava Olsen[3]
Department of Information Systems and Operations Management
University of Auckland, New Zealand

[1]m.islam@auckland.ac.nz
[2]t.arthanari@auckland.ac.nz
[3]t.olsen@auckland.ac.nz

Abstract

The sea port of any country is a very important link for the national supply chain. The importance increases significantly if the country is an island nation like Australia or New Zealand. Since the containerization of shipping ports, everything has changed, from cargo handling to cargo dropping at the customer's doorstep. The demand for containerized movement is increasing day by day because of the easy handling and security. On the other hand, the growing demand has introduced the empty container movement problem. There is a lot of literature that focuses on this problem. Interestingly, there is not sufficient literature on the empty truck movement problem which has a connection to the empty container repositioning problem to a certain extent, however the two problems are significantly different. During any import or export, trucks coming or going from the sea port are empty of containers most of the time, which creates traffic congestion, carbon emission, gate delay and most importantly, increased Vehicle Miles Travelled (VMT). There is no major review, as far as the authors know, on the empty truck movement problem. This paper will investigate some possible strategies to solve the problem for Ports of Auckland for achieving efficiency.

Key words: Empty Mileage, Ports of Auckland, Sustainability, Gate Delay, Game Theory

1 Introduction

Since 1990, the globalization of world trade has contributed extensively to the increased use of containers (Cheumo & Chen, 1998). Today, at least 60% of the world's seaborne cargo is transported in containers, and in more developed countries in particular routes this is 100% (Steenken, Voß, & Stahlbock, 2004). For some years, there has been a continuous growth of container volume which is equivalent to 6.6% worldwide and it is expected to increase at a pace of 5% up to 2015 (UNESCAP, 2006). The growth rate is significant and rational because containerization of goods offers tremendous benefits like increased security, ease of handling, reaching the doorsteps of more customers and making shipping cheaper. Currently, in a sea port, there are container terminals along with multi cargo wharves. Moreover, there is specialized equipment available to accelerate the cargo handling process in the container terminals. As a result, hinterland container transportation has increased rapidly for the last several years along with seaborne movements (Carpenter, 2006).

1.1 Research Motivation and Objectives

Containerization brings some problems like container fleet management, deciding between owning and leasing containers, empty container repositioning and container preloading preparation (Bandeira, Becker, & Borenstein, 2009). A core problem for the shipping companies is to reposition empty containers in the demanding ports (Shen & Khoong, 1995). There is substantive literature addressing this problem can be found, interested readers can be referred to (Cheumo & Chen, 1998; Choong, Cole, & Kutanoglu, 2002; Crainic, Gendreau, & Dejax, 1993; Lam, Lee, & Tang, 2007).

On the other hand, the empty movement of container-trucks increases costs in the national supply chain, creates congestion in the port's territory and consumes more diesels, which is not sustainable for the environment. In order to be competitive with other leading ports of the world, it is very important to focus on improving the efficiency level of the ports operations as there are many decentralized truck operators. But the problem is not addressed in the literature sufficiently. This paper will analyze the problem in the subsequent sections and propose appropriate strategies to deal with it.

1.2 Research Questions

This paper will analyze the empty container-truck movement problem. The goal is to investigate the collaboration opportunities among the stake holders. Thus, research questions include the problem investigation and co-operation prospects among players. Stakeholders are treated as players who play different roles in a container sea port.

1.3 Paper Outlines

Section 1 summarizes the adoption of containerization with research objectives and questions. Section 2 gives an overview of the Ports of Auckland along with describing possible threats from competition and POAL's recent adopted objectives to become more competitive nationally and internationally. Section 3 introduces a short summary of the port's regular operations. Moreover, the empty container-truck movement problem is described. Section 4 outlines possible mitigation strategies from the literature. Section 5 and 6 draws conclusions and possible future research streams consecutively.

2 Ports of Auckland

New Zealand is a trading nation where import-export represents 70% of GDP and geographically isolated where sea ports account 99% of international trades by weight (Auckland Regional Holdings). According to the Global Competitiveness Report of 2008, New Zealand ranked 24th (Nagar & Enderwick, 2010). This was possible because over the last two decades, government has transformed the country from an agrarian economy to a more industrialized economy so that New Zealand can compete globally, and per capita income has risen consecutively thereafter ("Foreign Investment Climate," 2010). Moreover, New Zealand has a total of 13 commercial ports which were corporatized in 1988, and basically owned by the local governments (*Economic infrastructure*, 1996).

2.1 Importance of Ports of Auckland

One of New Zealand's leading ports is the Ports of Auckland, which is a critical part of the international trade. The Ports of Auckland is NZ's largest container port by volume and claims 35.7% share of the total of NZ's containerized and non-containerized cargo (Madsen, 2010b). Ports of Auckland (POAL) contributes the most in the distribution of import and export values of the country. For example, in the year of 2007, POAL handled 50% of imports and 24% of exports, and overall, POAL handled 37% of the total annual trade by value (Colegrave, Simpson, & Denne, 2008). An economic impact assessment of POAL concluded that economic activity by the port will increase to $16 billion annually which is equivalent to 36% of New Zealand's GDP ("Port Facts," 2000).

2.2 Competitive Pressure

There are some significant differences between the container ports of Australia and New Zealand. For example Australia which has five times more population than New Zealand, has only 6 container ports and the ports are far away from each other (Smith, 2010). It is interesting to note that the Australian government has invested sufficient money to improve the infrastructure of the Australian ports (Auckland Regional Holdings). On the other hand, New Zealand has 11 container ports and the ports are very close to each other (Smith, 2010). Most of the ports can't handle 11,000 plus container ships and several billion dollars are indeed needed to improve the infrastructure of the ports, but the current profit levels are not sufficient to justify the huge investment necessary to improve the infrastructure of the ports. The largest vessel that New Zealand can handle now is 4,100 TEU capacities (Auckland Regional Holdings). Shipping lines want to use bigger ships because it saves costs for the shipping lines, importers and exporters.

According to big shipping lines, if the money is not spent to improve the infrastructure of the ports, New Zealand risks becoming a spoke of the Australian hub (Smith, 2010). Moreover, at the same time, developing an efficient supporting transport infrastructure is very important to accommodate these bigger ship volumes and increased traffic flows through the whole supply chain approach (Auckland Regional Holdings). For an example, according to the same report, California upgraded its port infrastructure to support more TEUs (6,000-7,000 TEU) without updating the supporting transport infrastructure, which resulted in bottlenecks in 1990.

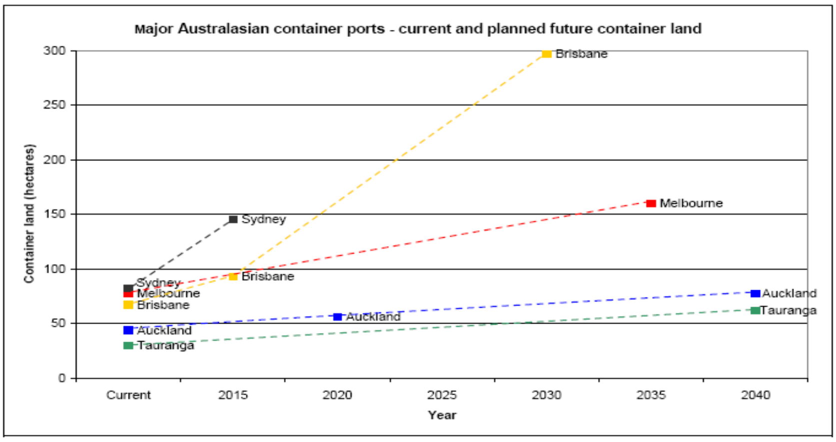


Figure 1: Major Australasian Current and Planned Future Container Land
Source: (Auckland Regional Holdings)

2.3 POAL's Objectives:

POAL's objectives can be classified in three categories, such as Improving Productivity, Reducing Unit Cost and Increasing Return on Capital Investment. Ports of Auckland is planning to improve the productivity of the port by handling more containers and eliminating container-truck queues (Jayne, 2009). This productivity improvement is very important for many reasons. One of those reasons is that North Island freight demand is expected to raise 70-75% in the next 20 years because of the increasing population growth rate which could cause systematic infrastructure failure (Madsen, 2010a).

3 Empty Container-truck Movement Problem

In a typical port, the operation of the container terminals can be seen as material flow (cargoes within containers) between two interfaces (Quayside VS Hinterland). Quayside operations are responsible for loading and unloading from ships, and hinterland operations account for loading and unloading from trucks and trains (Steenken, et al., 2004). POAL is not an exception. When a container ship arrives at the port, a berth is allocated for the ship and containers are loaded and unloaded consecutively by gantry cranes. The unloaded containers are put together in a particular place in the container terminal for transhipment in trucks or trains. Further, those are moved by straddle carriers or reach stackers to put on trucks or trains, or in empty container depots, sheds or packing centers. Rail is currently responsible for only 10% of the total landside movements of the Ports of Auckland (Ports of Auckland).

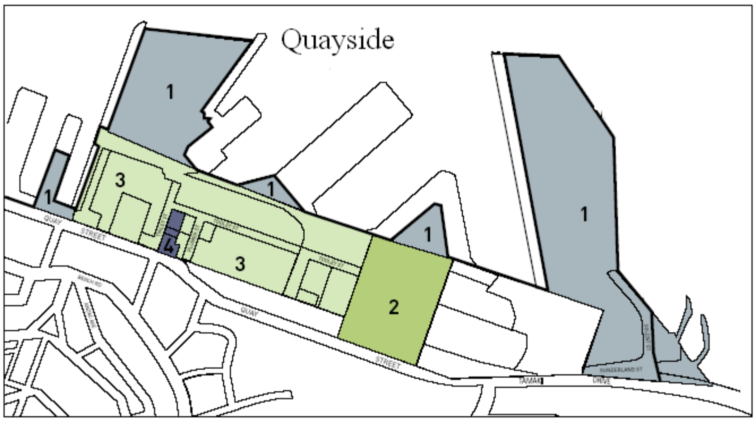


Figure 2: POAL's Current Layout
Source: (Ports of Auckland, 2008)

In an ideal import scenario, a completely empty truck comes to POAL. Afterwards it loads a laden container (cargo within) on it and goes to importer's premises or warehouse. Subsequently the importer unloads the cargo from the container and the container becomes empty. The empty container goes back to the shipping terminal on the truck. Sometimes, it brings back one empty container if it is needed by an exporter or if the same importer wants to export something. The steps are shown in detail in follwing.

On the other hand, in a typical export scenario, a completely empty truck comes to the Ports of Auckland. Afterwards, it loads an empty container which is demanded by the exporter for export. Then the truck goes back to the exporter's premises or warehouse. There the exporter loads the container with cargo. The container becomes a laden container. Thereafter, the truck attaches the laden container to its chassis and comes back to the Ports of Auckland. The steps are shown in the following figure 3. In both import and export scenarios, the question is, how to reduce the empty truck movements.

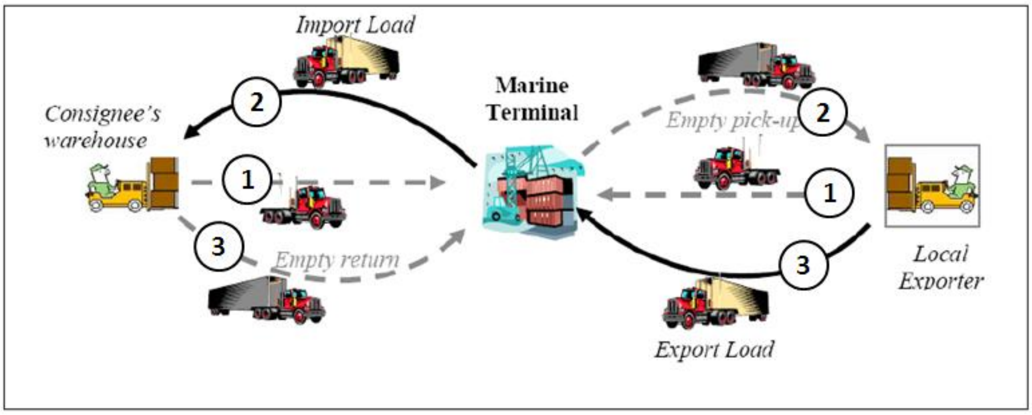


Figure 3: Illustration of Empty Truck Trips (Import and Export Scenario)
Source: (Theofanis & Boile, 2007)

At present, there are almost 250 trucking companies delivering and collecting containers from POAL (Dawson). The trucks operate for 24 hours per day and 7 days per week within the port territory. Per week in Ports of Auckland, almost 12% trucks operate on Monday, 14% on Tuesday, 2% on Sunday and 18% on each rest of the days of the week (Ports of Auckland, 2008). Moreover, research also found that most of the trucking companies attempt to deliver and collect containers in the morning (Ports of Auckland, 2008), refers to the following figure 4. This tendency of the trucking companies creates gate delay and traffic congestion in the area of the port during peak times. To alleviate the problem, POAL introduced Vehicle Booking System (VBS) to evenly distribute the road volumes (Ports of Auckland, 2008).

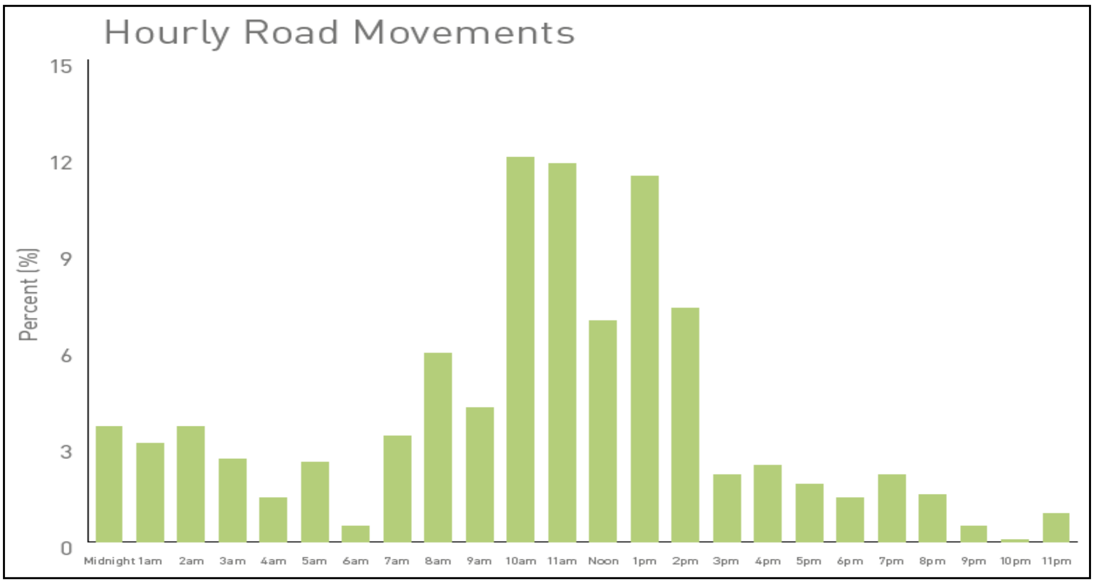


Figure 4: Hourly Road Movements of Trucks in Ports of Auckland
Source: (Ports of Auckland, 2008)

4 Collaboration is the Key

In any sea port, there are many stakeholders involved and they influence each other. Typically, the stakeholders are container owners (ocean carriers and leasing companies), ocean carriers (deep sea carriers and short sea carriers), stevedoring companies, container depot operators, consignees (may be an importer or a third party like a freight forwarder and consolidator), exporters, the marshalling company or the port authority. The success of the national supply chain depends on the collaboration of all the parties to remove inefficiency from the supply chain ("Port energy savings repay significant capital outlay," 2009). Collaboration will reduce the duplication of processes, cut inventories, avoid half-full vehicles and empty containers on back turns ("Collaboration key for success," 2008). Moreover, according to the same source, New Zealand has an added advantage that it has many smaller companies where it is possible for cross-functional co-operation, but at the same time it is important to avoid potential conflicts for mutual goals or domination by one. In the following, some of the strategies are mentioned that will help to mitigate the empty container-truck movement problem to a certain extent, but it requires a sufficient amount of collaboration to bring success. Game theory can be adapted to outline the collaboration scenarios which can be performed as the following.

Use of game theory to understand supply chain conflicts and cooperation is recently studied by researchers (*Handbook of quantitative supply chain analysis : modeling in the e-business era*, 2004; *Research methodologies in supply chain management*, 2005; Thun, 2005). It may be planned to build on this literature to model cooperation, collaboration and competition regarding the container truck movement. Assuming the players' behaviours depend only on the current state, a common assumption in operational settings, consider a repeated game by the players. In such a game, each state implies resultant players' actions in a that period, which then take the system to a new state and the players choose their actions based on the new state, and so forth. However, the resultant state is not certain and thus uncertainty enters the model. So an appropriate stochastic game formulation might be attempted. The mathematical analysis of that game may be made using existing literature or by developing new results specific to the situation as needed. Therefore the research consists of identifying and abstracting the real conflict situation into a suitable stochastic game and studying the game for equilibrium strategies for the different players. So, there will be a framework to compare the benefits of cooperation to the value creation. In addition it may have insight from game theory regarding how to share the additional value created among the players in a coalition.

4.1 Virtual Container Yard:

Virtual Container Yard (VCY) is an internet based system to collect information about an empty container interchange possibility between an importer and an exporter. This is also called street-turns (Deidda, Di Francesco, Olivo, & Zuddas, 2008) (Jula, Chassiakos, & Ioannou, 2006). This is one of the most effective methods of reducing empty container repositioning problem. Moreover, the same method can be used to reduce the unproductive empty container-truck movement problem. In this process, an importer will not return an empty container to the Auckland Port. Without returning, the empty container will be given to another nearby exporter to export cargo (Theofanis & Boile, 2007) as shown in the figure 5. This way, it will save couple of empty container-truck movements to and from the Ports of Auckland.

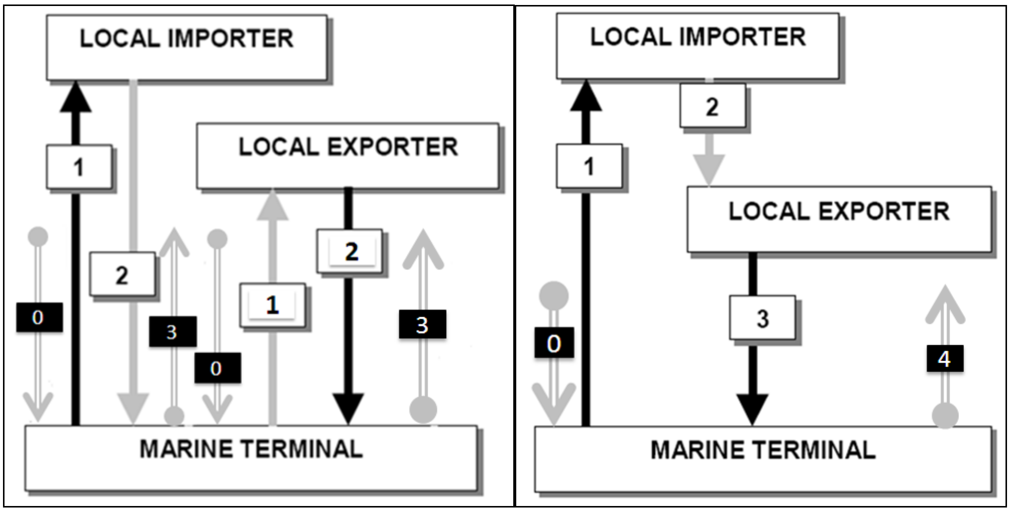


Figure 5: Left (Without Streen-turn), Right (With Street-turn)
Source: (The Tioga Group, 2002)

From the above figure it is evident that, in an import scenario, an empty truck (without any container) comes to the Ports of Auckland (step 0), grabs a loaded container and goes back to the importer's premises (step 1), the importer unloads cargo from the container and the truck with empty container returns to the Ports of Auckland (step 2), and the empty truck (without any container) returns from the port for another assignment. In this whole process, step 0 and step 3, are the unproductive empty truck trips. On the other hand, in an export scenario, step 0 and step 3 are also the empty truck trips. It is found in the research that only 25-30% of trucks servicing the Port of Auckland carry containers in and out in the same trip (Ports of Auckland, 2008). But with street-turn strategy, at least half of the unproductive truck trips can be reduced. It is quite possible for Ports of Auckland to adopt this strategy as imports and exports are almost equally balanced and most trucks are capable of bearing two TEU in each direction which will maximize the truck utilization rate significantly. At this moment, Ports of Auckland is working with a web-based platform, called VBS (Vehicle Booking System) that may help in the future to make this happen (Ports of Auckland, 2008).

4.2 Off-dock Empty Container Depot:

Usually, empty container depots are located within the port's premises because shipping lines like to see the empties there and find it more convenient to send those boxes in the places where shippers want it for exporting. But replacing the depots in other places will give significant benefits to the port authority in terms of traffic congestion, gate delay and reducing empty truck movements. According to the industry experts, off-dock empty container depot can be used as neutral point for interchange purpose in the street-turn strategy (The Tioga Group, 2002). According to this report, it can be used for other purposes as well. But clearly it will reduce empty container-truck movements in the container terminals because empties can be put in this neutral point instead of the container terminal, from where exporters will take the boxes during exporting needs. At present, there is an empty container depot located within the Ports of Auckland (United Containers Limited).

5 Conclusion

The successful implementation of concepts those are presented here, depends on the close participation of stakeholders (Theofanis & Boile, 2007). This is why it is very important to understand the needs of each player and respond positively to those expectations in order to come up with a widely accepted system. There are lots of barriers of successful collaboration with many players. But careful investigation and avoidance of conflicting objectives will bring success at the end. This current paper has performed two objectives. Firstly, the paper reports the work that is in progress relating to the empty container-truck movement problem using game theoretical approach, and secondly, presents the research problem for possible solutions.

6 Future Research Streams

In the future research, the Ports of Auckland will be used as a case to collect input for modelling and the trucking companies working with them could provide an understanding of the existing collaborations. So qualitative data from interviews with executives from relevant organizations will be collected and analysed as part of the research. In addition it is planned to do a simulation study to see whether the insight from the game model still holds, as the abstracted model may have omitted some details from the real situation.

Apart from that, it might be interesting to model the impediments of successful collaboration in the context of the Ports of Auckland. Moreover, location planning of off-dock empty container depot can be practical. The issue is not covered sufficiently in the literature. System thinking approach to understand the problem may be appealing.

7 References

Auckland Regional Holdings. Infrastructure: Facts and Issues. Retrieved 3rd November, 2010, from http://www.infrastructure.govt.nz/plan/submissions/pdfs/s-ifi-arh-oct09.pdf

Bandeira, D. L., Becker, J. L., & Borenstein, D. (2009). A DSS for integrated distribution of empty and full containers. *Decision Support Systems, 47*(4), 383-397.

Carpenter, M. A. (2006). The Box: How the Shipping Container Made the World Smaller and the World Economy Bigger. [Book Review]. *Administrative Science Quarterly, 51*(4), 656-658.

Cheumo, R. K., & Chen, C.-y. (1998). A Two-Stage Stochastic Network Model and Solution Methods for the Dynamic Empty Container. [Article]. *Transportation Science, 32*(2), 142.

Choong, S. T., Cole, M. H., & Kutanoglu, E. (2002). Empty container management for intermodal transportation networks. *Transportation Research Part E: Logistics and Transportation Review, 38*(6), 423-438.

Colegrave, F., Simpson, M., & Denne, T. (2008). Economic Impact of POAL. Retrieved 3rd November, 2010, from http://www.poal.co.nz/news_media/publications/POAL_economic_impact_report_covec_2008.pdf

Collaboration key for success. (2008). [Article]. *New Zealand Management, 55*(7), 8-8.

Crainic, T. G., Gendreau, M., & Dejax, P. (1993). Dynamic and stochastic models for the allocation of empty containers. *Operations Research, 41*(1), 102-126.

Dawson, R. Freight and Logistics. Retrieved 4th November, 2010, from http://www.exportandtrade.co.nz/webfiles/Adrenalin/files/Freight__Logistics_1.pdf

Deidda, L., Di Francesco, M., Olivo, A., & Zuddas, P. (2008). Implementing the street-turn strategy by an optimization model. [Article]. *Maritime Policy & Management, 35*(5), 503-516.

Economic infrastructure. (1996). (Country Report No. 02695618): EIU: Economist Intelligence Unit.

Foreign Investment Climate. (2010). [Article]. *New Zealand Country Review*, 58-61.

Handbook of quantitative supply chain analysis : modeling in the e-business era. (2004). Boston : Kluwer, c2004.

Jayne, V. (2009). FACE TO FACE: Jens Madsen - Berth Controller. [Article]. *New Zealand Management, 56*(4), 40-44.

Jula, H., Chassiakos, A., & Ioannou, P. (2006). Port dynamic empty container reuse. [Article]. *Transportation Research: Part E, 42*(1), 43-60.

Lam, S. W., Lee, L. H., & Tang, L. C. (2007). An approximate dynamic programming approach for the empty container allocation problem. *Transportation Research Part C: Emerging Technologies, 15*(4), 265-277.

Madsen, J. (2010a). Driving Greater Economic Productivity with Targeted Infrastructure Investments. Retrieved 08 August, 2010, from http://www.poal.co.nz/news_media/speeches_presentations/20100428_TransportInfrastructureandEconomicProsperitySummitspeech.pdf

Madsen, J. (2010b). The Role of Ports and Shipping in Stimulating Economic Prosperity. Retrieved 03 August, 2010, from http://www.poal.co.nz/news_media/speeches_presentations/20100428_TransportInfrastructureandEconomicProsperitySummitPresentation.pdf

Nagar, S., & Enderwick, P. (2010). India: The next big opportunity for New Zealand business? [Article]. *University of Auckland Business Review, 12*(1), 1-11.

Port energy savings repay significant capital outlay. (2009). [Article]. *New Zealand Management*, 5-5.

Port Facts. (2000). [Article]. *New Zealand Management, 47*(7), 8.

Ports of Auckland. Rail Exchange. Retrieved 4th November, 2010, from http://www.poal.co.nz/facilities_services/facilities/rail_exchange.htm

Ports of Auckland. (2008). Port Development Plan. Retrieved 4th November, 2010, from http://www.poal.co.nz/news_media/publications/POAL_port_development_plan_2008.pdf

Research methodologies in supply chain management. (2005). Heidelberg ; New York : Physica-Verlag, 2005.

Shen, W. S., & Khoong, C. M. (1995). A DSS for empty container distribution planning. *Decision Support Systems, 15*(1), 75-82.

Smith, N. (2010). Ports - There will be blood. Retrieved 3rd November, 2010, from http://www.nzherald.co.nz/economy/news/article.cfm?c_id=34&objectid=10638742

Steenken, D., Voß, S., & Stahlbock, R. (2004). Container terminal operation and operations research - a classification and literature review. [Article]. *OR Spectrum, 26*(1), 3-49.

The Tioga Group. (2002). Empty Ocean Container Logistics Study. Retrieved 5th November, 2010, from http://www.scag.ca.gov/goodsmove/pdf/Final_Empty_Containers_Report.pdf

Theofanis, S., & Boile, M. (2007). Investigating the Feasibility of Establishing a Virtual Container Yard to Optimize Empty Container Movement in the NY-NJ Region. Retrieved 4th November, 2010, from http://www.utrc2.org/research/assets/105/FinalReport1.pdf

Thun, J.-H. (2005). The Potential of Cooperative Game Theory for Supply Chain Management. In H. Kotzab, S. Seuring, M. Müller & G. Reiner (Eds.), *Research Methodologies in Supply Chain Management* (pp. 477-491): Physica-Verlag HD.

UNESCAP. (2006). Trade Structures, Chapter 4. Retrieved 02 November, 2010, from http://www.unescap.org/ttdw/publications/tfs_pubs/pub_2398/pub_2398_ch4.pdf

United Containers Limited. Depot Locations. Retrieved 06th November, 2010, from http://www.containershop.co.nz/locations.html

Optimal Ordering Policy with Inspection Errors and Learning Curve Consideration on Imperfect Quality Items

Divya Jain, K.K. Aggarwal
Department of Operational Research, Faculty of Mathematical Sciences, University of Delhi, Delhi 110007, India
divvya.jain@gmail.com

Abstract

This paper develops an economic ordering policy for imperfect quality items considering the effects of learning curve and inspection errors simultaneously. Many a times, companies' efforts in quality improvement lead to learning effects in the quality of the products. Thus, in this paper we have observed the effect of learning on the percentage defectives of the shipments received from the supplier. The items received from the supplier undergoes a 100% inspection process but the inspection process may commit two types of errors, namely, Type I error and Type II error. In this scenario, some of the defective items may get passed on to the customer, which are later received back from the market. The considered mathematical model aims at maximizing the net profit obtained through the sales of both perfect and imperfect quality items while incurring various costs. Finally, we have included a numerical example to demonstrate the applicability of the proposed model.

Key words: Inventory, Imperfect quality, Learning Curve, Inspection.

1 Introduction

Traditionally, inventory models were developed based on the assumption that the items are of perfect quality. However, product quality may not be always perfect. A proportion of the produced/ordered items can be found to be defective. This aspect has received attention from many researchers. Many authors have addressed the issue of lot sizing decision for imperfect quality items. Initially, Rosenblatt and Lee [12] considered that the presence of defective items resulted in smaller lot sizes. Porteus [11] considered an imperfect production process with significant relationship between quality and lot size. Schwaller [14] extended EOQ model by assuming that the incoming lots contain a fixed proportion of defectives and derived policy using fixed and variable inspection costs. Zhang and Gerchak [17] developed a joint lot sizing and inspection policy for an EOQ model where the lot was assumed to contain a random proportion of defective units. They considered a model where the defective units cannot be used and thus must be replaced by non-defective ones. Recently, Salameh and Jaber [13] developed an EOQ model considering the lot to contain a random fraction of imperfect quality items with a known probability distribution. They considered that the received lot undergoes a 100% inspection process and at the end of the inspection, defective items are sold as a single batch. Several researchers have extended the work of Salameh and Jaber [13]. Cárdenas-Barrón [2] observed a minor correction for the expression of optimal lot size.

Goyal and Cárdenas-Barrón [4] presented a simple approach which approximately determines the order quantity when a random proportion of units are defective. Papachristos and Konstantaras [10] examined the assumptions made for avoiding shortages in S & J's model. Recently, Khan et al. [8] extended the work of S & J by considering inaccuracy in the inspection process. They considered that an inspector, while classifying the items as defective and non-defectives could make misclassifications with fixed rates.

Moreover, many researchers have investigated the effect of learning curve. A learning curve is the phenomenon of improvement in the performance due to repetitions. It was introduced in 1936 by T.P Wright in an article "Factors affecting the cost of airplanes" in the Journal of Aeronautical Sciences. Wright is believed to be the first to quantify the learning curve function. He gave the power form to the learning curve model.

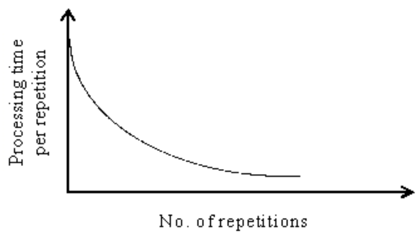


Fig.1. Wright's learning curve

Wright's learning curve is the most widely used and accepted and it has also been found to fit the empirical data quite appropriately in the studies done by Yelle [16] , Lieberman [9] etc. Thereafter, several authors have studied the effects of learning in inventory systems. Keachie and Fontana [7] examined the effects of learning on optimal lot size. Chand [3] studied the effects of learning in set-ups and process quality. Jaber and Bonney [5] extended the work of Chand [3] by assuming learning and forgetting to occur simultaneously in set-ups and process quality. Badiru [1] performed a multivariate analysis of the effect of learning and forgetting on product quality. He stated that learning affects worker performance, which ultimately can affect product quality. Recently, Jaber et al., [6] investigated the effects of learning in product quality for developing an economic production quantity model.

The author's survey of the relevant literature reveals that there is no published work that investigates the effect of learning in product quality and inspection errors simultaneously. Therefore, this paper develops an inventory model to determine the optimal order quantity in the presence of inspection errors and learning in product quality. This paper is organized as follows: 1. Introduction; 2. Notations and assumptions; 3. Mathematical model; 4. Concavity analysis of function 5. Numerical example; 6. Conclusion and 7. References.

2 Notations and Assumptions

The relevant assumptions and notations used to develop mathematical model in this paper are:

2.1 Assumptions

1. The demand rate is deterministic and constant.
2. Shortages are not allowed.
3. Lead time is negligible.
4. Replenishment is instantaneous

5. The received lot contains imperfect quality items and thus is screened for separating defective and non-defective items through a 100% inspection process.
6. The items found defective are sold at a discounted price.
7. The percentage of defectives per lot follows the Wright's learning curve.

2.2 Notations

d	Demand rate (units / unit time)
y_i	Order size of i^{th} cycle
c	Unit purchase cost
K	Ordering cost per order
h	Holding cost per unit per unit time
$p(i)$	Percentage of defectives in i^{th} shipment
E_1	Probability of Type I error
E_2	Probability of Type II error
β_{1i}	Effective rejection rate of i^{th} cycle
β_{2i}	Effective return rate of i^{th} cycle
x	Screening rate
c_s	Unit cost of inspection
c_a	Unit cost of accepting a defective item
c_r	Unit cost of rejecting a non-defective item
s	Unit selling price of non-defective items
v	Unit selling price of defective items $(v < s)$
T_i	Cycle time of i^{th} cycle
t_{1i}	Inspection time of i^{th} cycle
t_{2i}	Remaining time of i^{th} cycle $(= T_i - t_{1i})$

3 Formulation of the Mathematical Model

Consider a lot of size y_i received from the supplier in the beginning of i^{th} cycle. Each received lot contains $p(i)$ percentage defectives and thus it is subjected to a 100% inspection process. This inspection is carried out so that the defective items can be removed from the lot and sold at a discounted price as a single batch. When the lot is screened for the defective items, two types of error may be committed, namely Type I error and Type II error. Let E_1 be the probability of committing Type I error, i.e. probability of rejecting a non-defective item during inspection and let E_2 be the probability of committing Type II error, i.e. probability of accepting a defective item. Therefore, the number of items rejected from the lot of size y_i is the sum of incorrectly rejecting a non-defective item and correctly rejecting a defective item. We have,

Number of defective items in the i^{th} lot, $N_{d\,i} = y_i(1 - p(i))$;

Number of non-defectives in the i^{th} lot, $N_{nd\,i} = y_i\,p(i)$;

Number of items rejected from the i^{th} lot, $N_{r\,i} = E_1 y_i(1 - p(i)) + (1 - E_2)\,y_i\,p(i)$;

Therefore, the effective rejection rate of i^{th} cycle is,

$\beta_{1i} = \dfrac{N_{ri}}{y_i} = E_1(1 - p(i)) + (1 - E_2)p(i)$;

The items rejected during the inspection process are sold as a single batch at a discounted price of v per unit. Also due to inspection error, some of the defective items are identified as non-defectives during inspection and thus get passed on to the customers. These items are later returned from the market and sold as a single batch at a discounted price at the end of the cycle.

So, we have

Number of defective items not screened in inspection, $N_{mi} = E_2 y_i p(i)$;

Therefore, effective return rate of i^{th} cycle is, $\beta_{2i} = \dfrac{N_{mi}}{y_i} = E_2 p(i)$;

The behaviour of the inventory levels in a cycle is shown in figures 2(a), 2(b) and 2(c). The lot for i^{th} cycle is received at time zero, then till time t_{1i} the inventory is classified into defectives and non-defectives while serving the demand from the items classified as non-defective. Here, it is reasonable to assume that the rate of inspection is greater than the demand rate $(x > d)$. Also, due to inspection error, some of the defective items that are sold to the customers get returned by the market as shown in fig. 2(c). These returned items are replaced with non-defective ones. Thus, in an i^{th} cycle after time t_{1i} until the end of the cycle, inventory level as shown in fig. 2(a) decreases due to demand and the replacement of returned items.

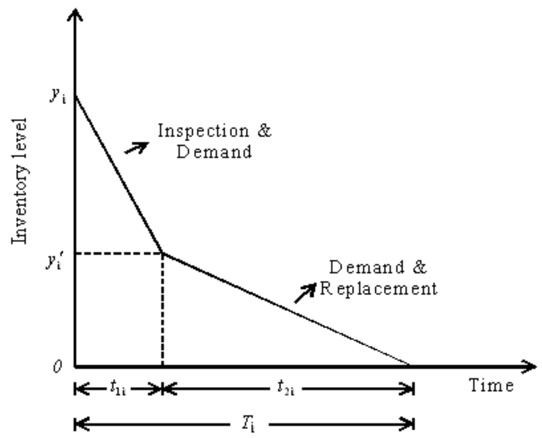


Fig.2(a). The inventory level of the i^{th} shipment

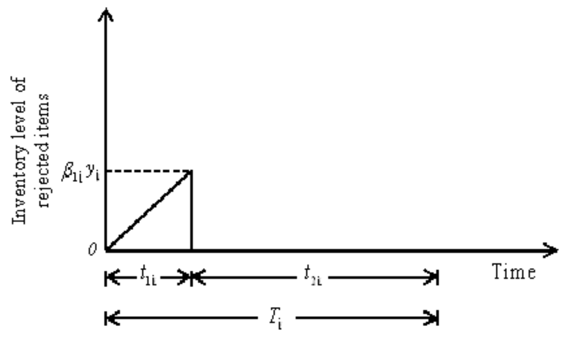


Fig. 2(b). The inventory level of rejected items of i^{th} cycle

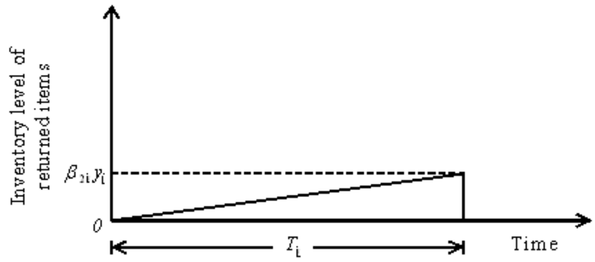


Fig. 2(c). The inventory level of items returned during i^{th} cycle

In this paper, the learning effect on the percentage of defectives per lot is assumed to follow Wright's power function formula and can be expressed as $p(i) = p_0 i^{-L}$ where p_0 is the initial percentage of defectives, L is the learning slope in the Wright's formulation for the learning curve $(0 \le L < 1)$ and i is the cumulative number of shipments. This form of learning curve for percentage defectives has earlier been described by Jaber [6] and found to be appropriate empirically for the learning phase of the learning curve.

To avoid shortages, it is reasonable to assume that the number of items inspected as non-defective is at least equal to the sum total of the actual demand and the number of units used to replace the returned ones, i.e.

$$y_i(1-\beta_{1i}) \ge dT_i + y_i\beta_{2i} \Rightarrow y_i(1-(\beta_{1i}+\beta_{2i})) \ge dT_i$$

For the limiting case in each cycle, we have

$$y_i(1-(\beta_{1i}+\beta_{2i})) = dT_i \Rightarrow T_i = \frac{y_i(1-(\beta_{1i}+\beta_{2i}))}{d} \tag{1}$$

The total revenue for an i^{th} cycle is the sum total of revenue generated from the sales of non-defectives and defectives and is given by

$$R_i = sy_i(1-\beta_{1i}) + vy_i(\beta_{1i}+\beta_{2i}) \tag{2}$$

and the total cost for an i^{th} cycle is the sum total of ordering cost, purchase cost, inspection cost, error costs and the holding cost and is given by

$$C_i = K + cy_i + c_s y_i + c_r y_i(\beta_{1i}+\beta_{2i} - p(i)) + c_a y_i \beta_{2i}$$

$$+ h\left\{ \underbrace{\frac{(y_i - y_i')t_{1i}}{2} + y_i' t_{1i} + \frac{y_i' t_{2i}}{2}}_{\text{Refer to fig 1(a)}} + \underbrace{\frac{y_i \beta_{1i} t_{1i}}{2}}_{\text{Refer to fig 1(b)}} + \underbrace{\frac{y_i \beta_{2i} T_i}{2}}_{\text{Refer to fig 1(c)}} \right\}$$

where $y_i' = y_i - y_i\beta_{1i} - dt_{1i} \Rightarrow y_i - y_i' = y_i\beta_{1i} + dt_{1i}$; $t_{1i} = \frac{y_i}{x}$ and $t_{2i} = T_i - t_{1i}$.

$$\Rightarrow C_i = K + cy_i + c_s y_i + c_r y_i(\beta_{1i}+\beta_{2i} - p(i)) + c_a y_i \beta_{2i}$$

$$+ \frac{h}{2}\left\{ \frac{y_i^2(1+\beta_{1i})}{x} + y_i(1-\beta_{1i}+\beta_{2i})T_i - \frac{dy_i T_i}{x} \right\} \tag{3}$$

Thus, the total profit per i^{th} cycle is

$$TP_i = R_i - C_i$$

$$= sy_i(1-\beta_{1i}) + vy_i(\beta_{1i}+\beta_{2i}) - K - cy_i - c_s y_i - c_r y_i(\beta_{1i}+\beta_{2i} - p(i)) \tag{4}$$

$$- c_a y_i \beta_{2i} - \frac{h}{2}\left\{ \frac{y_i^2(1+\beta_{1i})}{x} + y_i(1-\beta_{1i}+\beta_{2i})T_i - \frac{dy_i T_i}{x} \right\}$$

Here, for an infinite planning horizon, i shall be regarded as an input parameter, thereby leaving the total profit function TP_i a function of variable y_i. Hence, our objective is to obtain the optimal value of y_i that maximizes the net profit.

Now, the total profit per unit time for an i^{th} cycle is given by

$TPU(y_i) = \dfrac{R_i - C_i}{T_i}$ where $T_i = \dfrac{y_i(1-(\beta_{1i}+\beta_{2i}))}{d}$;

$$\therefore TPU(y_i) = \frac{d}{[1-(\beta_{1i}+\beta_{2i})]}\left\{s(1-\beta_{1i}) + v(\beta_{1i}+\beta_{2i}) - \frac{K}{y_i} - c - c_s - c_r(\beta_{1i}+\beta_{2i} - p(i)) - c_a\beta_{2i}\right\} - \frac{hy_i}{2[1-(\beta_{1i}+\beta_{2i})]}\left\{\frac{d(2\beta_{1i}+\beta_{2i})}{x} + \left[(1-\beta_{1i})^2 - \beta_{2i}^{\;2}\right]\right\} \tag{5}$$

The total profit per unit time is a concave function of y_i, as

$$\frac{dTPU(y_i)}{dy_i} = \frac{Kd}{y_i^{\;2}[1-(\beta_{1i}+\beta_{2i})]} - \frac{h}{2[1-(\beta_{1i}+\beta_{2i})]}\left\{\frac{d(2\beta_{1i}+\beta_{2i})}{x} + \left[(1-\beta_{1i})^2 - \beta_{2i}^{\;2}\right]\right\} \tag{6}$$

$$\Rightarrow \frac{d^2TPU(y_i)}{dy_i^{\;2}} = \frac{-2Kd}{y_i^{\;3}[1-(\beta_{1i}+\beta_{2i})]} \quad < \quad 0 \qquad \forall \;\; y_i > 0$$

Note that, $0 < 1-(\beta_{1i}+\beta_{2i}) \le 1$ as $1-(\beta_{1i}+\beta_{2i}) = (1-E_1)(1-p(i))$.

Therefore, the necessary condition for $TPU(y_i)$ to be maximum is $\dfrac{dTPU(y_i)}{dy_i} = 0$.

Hence, the optimal value of y_i can be obtained by differentiating the function $TPU(y_i)$ w.r.t. y_i and equating the result to zero, i.e. by setting $\dfrac{dTPU(y_i)}{dy_i} = 0$.

$$\Rightarrow \qquad y_i^{\;*} = \sqrt{\frac{2Kd}{h\left\{\dfrac{d(2\beta_{1i}+\beta_{2i})}{x} + \left[(1-\beta_{1i})^2 - \beta_{2i}^{\;2}\right]\right\}}} \quad \text{for each } i \tag{7}$$

where $\beta_{1i} = E_1(1-p(i)) + (1-E_2)p(i)$; $\beta_{2i} = E_2 p(i)$ and $p(i) = p_0 i^{-L}\ (0 \le L < 1)$

Now, let us consider a finite planning horizon of n cycles in which at the beginning of each cycle, a lot of fixed size y is received from the supplier and $\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} y = ny = Q(say)$. Then the total revenue and the total cost functions over the finite planning horizon are given by

$$\sum_{i=1}^{n} R_i = sQ - \frac{Q}{n}(s-v)\sum_{i=1}^{n}\beta_{1i} + v\frac{Q}{n}\sum_{i=1}^{n}\beta_{2i} \tag{8}$$

$$\sum_{i=1}^{n} C_i = nK + (c+c_s)Q + c_r.\frac{Q}{n}\left\{\sum_{i=1}^{n}\beta_{1i} - \sum_{i=1}^{n} p(i)\right\} + (c_r + c_a).\frac{Q}{n}\sum_{i=1}^{n}\beta_{2i} + \frac{h}{2d}.\frac{Q^2}{n^2}\left\{\frac{d}{x}\sum_{i=1}^{n}(2\beta_{1i}+\beta_{2i}) + \sum_{i=1}^{n}\left[(1-\beta_{1i})^2 - \beta_{2i}^{\;2}\right]\right\} \tag{9}$$

From $\beta_{1i} = E_1(1-p(i)) + (1-E_2)p(i)$; $\beta_{2i} = E_2 p(i)$ and $p(i) = p_0 i^{-L}\ (0 \le L < 1)$

we can obtain, $\sum_{i=1}^{n}\beta_{1i} = nE_1 + (1-E_1-E_2)\sum_{i=1}^{n} p(i)$; $\sum_{i=1}^{n}\beta_{2i} = E_2\sum_{i=1}^{n} p(i)$ and

$$\sum_{i=1}^{n} p(i) = \sum_{i=1}^{n} p_0 i^{-L} \cong \int_0^n p_0 u^{-L} du = \frac{p_0 n^{1-L}}{1-L} \tag{10}$$

Similarly, we can also obtain

$$\sum_{i=1}^{n}\left[(1-\beta_{1i})^2-\beta_{2i}{}^2\right]=n(1-E_1)^2-2(1-E_1)(1-E_1-E_2)\sum_{i=1}^{n}p(i)+(1-E_1)(1-E_1-2E_2)\sum_{i=1}^{n}p^2(i)$$

and $\sum_{i=1}^{n}p^2(i)=\dfrac{p_0{}^2.n^{1-2L}}{1-2L}$.

Then Equations (8) and (9) reduces to

$$\sum_{i=1}^{n}R_i = sQ(1-E_1)+vQE_1-Q\{(s-v)(1-E_1)-sE_2\}.\frac{p_0n^{-L}}{1-L} \tag{11}$$

and

$$\begin{aligned}\sum_{i=1}^{n}C_i = nK+(c+c_s)Q+c_rQE_1\left(1-\frac{p_0n^{-L}}{1-L}\right)+c_aQE_2.\frac{p_0n^{-L}}{1-L}+\frac{h}{2d}.\frac{Q^2}{n}\Bigg\{(1-E_1)^2+\frac{2E_1d}{x}\\ +\left[\frac{d}{x}(2-2E_1-E_2)-2(1-E_1)(1-E_1-E_2)\right].\frac{p_0n^{-L}}{1-L}+(1-E_1)(1-E_1-2E_2)\frac{p_0{}^2n^{-2L}}{1-2L}\Bigg\}\end{aligned} \tag{12}$$

The total profit over n cycles is given by $\sum_{i=1}^{n}TP_i=\sum_{i=1}^{n}R_i-\sum_{i=1}^{n}C_i$ which is a function of two variables Q and n, where n being the number of replenishments is a discrete variable. Moreover, $\sum_{i=1}^{n}TP_i$ is an increasing function of Q and concave in n (Refer to Theorem A in section 4), so we find the optimal number of shipments $n*$ that maximizes the total profit function at a fixed value of Q, say Q_0 by searching for $n*$ such that $\sum_{i=1}^{n*-1}TP_i(Q_0,n*-1)<\sum_{i=1}^{n*}TP_i(Q_0,n*)$ and $\sum_{i=1}^{n*+1}TP_i(Q_0,n*+1)<\sum_{i=1}^{n*}TP_i(Q_0,n*)$.

4 Concavity Analysis of the Total Profit Function over n Cycles

Theorem A: The total profit function, $\sum_{i=1}^{n}TP_i(Q_0,n)$ is a concave function of n.

Proof: To simplify the analysis and without loss of generality, let us assume that the function $\sum_{i=1}^{n}TP_i(Q_0,n)$ is a continuous and differentiable function over n and n is a real number, $n>0$.

$$\begin{aligned}\sum_{i=1}^{n}TP_i(Q_0,n)=[s(1-E_1)+vE_1-c-c_s-c_rE_1].Q_0-nK\\ -[(s-v)(1-E_1)-sE_2-c_rE_1+c_aE_2].Q_0.\frac{p_0n^{-L}}{1-L}-\frac{h}{2d}.\frac{Q_0{}^2}{n}\Bigg\{(1-E_1)^2+\frac{2E_1d}{x}\\ +\left[\frac{d}{x}(2-2E_1-E_2)-2(1-E_1)(1-E_1-E_2)\right].\frac{p_0n^{-L}}{1-L}+(1-E_1)(1-E_1-2E_2)\frac{p_0{}^2n^{-2L}}{1-2L}\Bigg\}\end{aligned}$$

(13)

$$\frac{d\left(\sum_{i=1}^{n} TP_i(Q_0,n)\right)}{dn} = -K + \left[(s-v)(1-E_1) - sE_2 - c_r E_1 + c_a E_2\right].\frac{Q_0}{n}.\left(\frac{L}{1-L}\right).p_0 n^{-L}$$
$$+\frac{h}{2d}.\frac{Q_0^{\ 2}}{n^2}\left\{(1-E_1)^2 + \frac{2E_1 d}{x} + \left[\frac{d}{x}(2-2E_1-E_2)\right.\right.$$
$$\left.\left.-2(1-E_1)(1-E_1-E_2)\right].\left(\frac{1+L}{1-L}\right).p_0 n^{-L} + (1-E_1)(1-E_1-2E_2).\left(\frac{1+2L}{1-2L}\right).p_0^{\ 2} n^{-2L}\right\} \tag{14}$$

and

$$\frac{d^2\left(\sum_{i=1}^{n} TP_i(Q_0,n)\right)}{dn^2} = -\left[(s-v)(1-E_1) - sE_2 - c_r E_1 + c_a E_2\right].\frac{Q_0}{n^2}.\left(\frac{L(1+L)}{1-L}\right).p_0 n^{-L}$$
$$-\frac{h}{d}.\frac{Q_0^{\ 2}}{n^3}\left\{(1-E_1)^2 + \frac{2E_1 d}{x} + \frac{1}{2}.\left[\frac{d}{x}(2-2E_1-E_2) - 2(1-E_1)(1-E_1-E_2)\right].\right.$$
$$\left.\left(\frac{(1+L)(2+L)}{1-L}\right).p_0 n^{-L} + (1-E_1)(1-E_1-2E_2)\left(\frac{(1+L)(1+2L)}{1-2L}\right).p_0^{\ 2} n^{-2L}\right\} \tag{15}$$

Now, in order to obtain the optimal value of n, we set $\frac{d\left(\sum_{i=1}^{n} TP_i(Q_0,n)\right)}{dn} = 0$

$$\Rightarrow \left[(s-v)(1-E_1) - sE_2 - c_r E_1 + c_a E_2\right].\frac{Q_0}{n}.\left(\frac{L}{1-L}\right).p_0 n^{-L}$$
$$= K - \frac{h}{2d}.\frac{Q_0^{\ 2}}{n^2}\left\{(1-E_1)^2 + \frac{2E_1 d}{x} + \left[\frac{d}{x}(2-2E_1-E_2)\right.\right.$$
$$\left.\left.-2(1-E_1)(1-E_1-E_2)\right].\left(\frac{1+L}{1-L}\right).p_0 n^{-L} + (1-E_1)(1-E_1-2E_2)\left(\frac{1+2L}{1-2L}\right).p_0^{\ 2} n^{-2L}\right\} \tag{16}$$

Multiplying both sides of Eq. (16) by $\frac{(1+L)}{n}$ and substituting the value of its L.H.S in Eq. (15) we get,

$$\frac{d^2\left(\sum_{i=1}^{n} TP_i(Q_0,n)\right)}{dn^2} = -\frac{(1+L)K}{n} - \frac{h}{2d}.\frac{Q_0^{\ 2}}{n^3}\left\{\left[(1-E_1)^2 + \frac{2E_1 d}{x}\right].(1-L)\right.$$
$$+\left[\frac{d}{x}(2-2E_1-E_2) - 2(1-E_1)(1-E_1-E_2)\right].\left(\frac{1+L}{1-L}\right).p_0 n^{-L} \tag{17}$$
$$\left.+(1-E_1)(1-E_1-2E_2)\left(\frac{(1+L)(1+2L)}{1-2L}\right).p_0^{\ 2} n^{-2L}\right\}$$

Since $0 \le L < 1$, $0 \le E_1 < 1$, $0 \le E_2 < 1$ and $n > 0$ we conclude that

$\frac{d^2\left(\sum_{i=1}^{n} TP_i(Q_0, n)\right)}{dn^2} < 0$ at the optimal value of n that can be obtained from equation (16) provided $L \neq \frac{1}{2}$. Hence, $\sum_{i=1}^{n} TP_i(Q_0, n)$ is a concave function of n.

Also, based on the parametric values given in the numerical example, the following graph depicts the concavity of the function $\sum_{i=1}^{n} TP_i(Q_0, n)$.

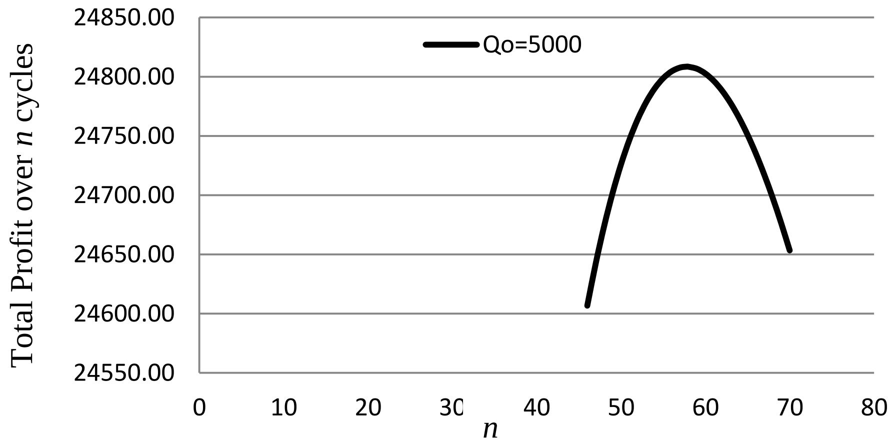


Fig.3. Concavity of the Total Profit Function

5 Numerical Example

5.1 Example 1. Let $K = \$100$, $d = 40{,}000$ units per year, $x = 87{,}600$ units per year, $h = \$5$ per unit per year, $s = \$50$, $v = \$20$, $c = \$25$, $c_s = \$2$, $c_r = \$40$, $c_a = \$80$, $p_0 = 0.35$, $E_1 = 0.2$, $E_2 = 0.3$ and $L = 0.4$. Then for an infinite planning horizon, using equations (7) and (5), the percentage defectives per cycle, the optimal lot size and the corresponding net profit per unit time is given as follows:

Cycle no. (i)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
p(i)	**0.35**	**0.265**	**0.226**	**0.201**	**0.184**	**0.171**	**0.161**	**0.152**
y^*_i	**1441.49**	**1433**	**1428.35**	**1425.28**	**1423.04**	**1421.3**	**1419.89**	**1418.73**
$TPU^*(y_i)$	**8558.05**	**151696.09**	**207985.08**	**239939.18**	**261169.4**	**276581.95**	**288427.99**	**297902.91**

5.2 Example 2. Given the same parametric values as mentioned in example 1, consider a finite planning horizon in which a fixed number of $Q_0 = 5000$ units are to be delivered in n equal lot sizes. By applying the computational procedure mentioned at the end of section 3, we get $n^* = 57.78 \approx 58$ as the optimal number of replenishments and the total profit over n cycles is then given by $\sum_{i=1}^{n^*} TP_i(Q_0, n^*) = \$24808.47$.

6 Conclusion

In this paper, we developed a mathematical model to determine the optimal order quantity of imperfect quality items which are subjected to 100% inspection on their arrival, but the inspection process is not error-free. Moreover, we also investigated the effect of learning on percentage of defectives in each lot. To examine the behaviour of learning, power function formula of Wright's learning curve model has been used. In this study, it has been found that the influence of learning and inspection errors is significant in determining the optimal lot size.

7 References

[1] Badiru, A.B., 1995. "Multivariate analysis of the effect of learning and forgetting on product quality." *International Journal of Production Research* **33**, 777–794.

[2] Cárdenas-Barrón, L.E., 2000. "Observation on: Economic production quantity model for items with imperfect quality." *International Journal of Production Economics* **67** (2), 201.

[3] Chand, S., 1989. "Lot sizes and setup frequency with learning and process quality." *European Journal of Operational Research* **42**, 190–202.

[4] Goyal, S.K., Cárdenas-Barrón, L.E., 2002. "Note on: Economic production quantity model for items with imperfect quality— a practical approach." *International Journal of Production Economics* **77** (1), 85–87.

[5] Jaber, M.Y., Bonney, M., 2003. "Lot sizing with learning and forgetting in set-ups and in product quality." *International Journal of Production Economics* **83**, 95–111.

[6] Jaber, M.Y., Goyal, S.K., Imran, M., 2008. "Economic production quantity model for items with imperfect quality subject to learning effects", *International Journal of Production Economics* **115,** 143– 150.

[7] Keachie, E.C., Fontana, R.J., 1966. "Effects of learning on optimal lot size." *Management Science* **32** (2), 12-108.

[8] Khan, M., Jaber, M.Y., Bonney, M., 2010. "An economic order quantity (EOQ) for items with imperfect quality and inspection errors". *International Journal of Production Economics* (Article in Press) Available online, 2nd February 2010.

[9] Lieberman, M.B., 1987. "The learning curve, diffusion, and competitive strategy." *Strategic Management Journal* **8** (5), 441– 452.

[10] Papachristos, S., Konstantaras, I., 2006. "Economic ordering quantity models for items with imperfect quality." *International Journal of Production Economics* **100** (1), 148–154.

[11] Porteus, E. L., 1986. "Optimal lot sizing, process quality improvement and setup cost reduction." *Operations Research* **34**, 137-144.

[12] Rosenblatt, M.J., Lee, H.L., 1986. "Economic production cycles with imperfect production processes." *IIE Transactions* **18**, 48–55.

[13] Salameh, M.K., Jaber, M.Y., 2000. „Economic production quantity model for items with imperfect quality." *International Journal of Production Economics* **64** (1–3), 59–64.

[14] Schwaller, R.L., 1988. "EOQ under inspection costs." *Production and Inventory Management* **29**(3), 22.

[15] Wright, T.P., 1936. "Factors affecting the cost of airplanes." *Journal of Aeronautical Sciences* **3**, 122–128.

[16] Yelle, L.E., 1979. "The learning curve: historical review and comprehensive survey." *Decision Sciences* **10**, 302–328.

[17] Zhang, X., Gerchak, Y., 1990. "Joint lot sizing and inspection policy in an EOQ model with random yield." *IIE Transactions* **22**(1), 41.

Optimisation of Mould Filling Parameters of the Compression Resin Transfer Moulding Process

Sherry Hsu, Matthias Ehrgott and Piaras Kelly
Department of Engineering Science
The University of Auckland
New Zealand
m.ehrgott@auckland.ac.nz

Abstract

The Compression Resin Transfer Moulding Process (CRTM) is a popular type of Liquid Composite Moulding Process (LCM) commonly used for manufacturing composite materials. In this paper we consider the optimisation of the manufacturing processing time and the machine tooling force for the CRTM process. Since this process requires large forces during compression, force evaluation and prediction provides great advantages for the industry as it enables structural analysis of the moulds. Not only it does lead to cost effective tooling design, it also allows for proper selection of cost effective moulds and supporting equipment. The tooling force, moreover, is in conflict with manufacturing time, which is another objective of particular interest in the industry.

In recent years, the advancement of CRTM simulation software allows accurate prediction of the processing objectives thus making it unnecessary to run through the expensive experiments physically. In this process, we use such a simulation software called SimLCM and combine it with a popular NSGA-II evolutionary multi-objective optimisation (EMO) algorithm to optimise maximum tooling force and processing time with respect to three manufacturing parameters. The EMO algorithm uses SimLCM as a black box to evaluate the objective function values for a population of solutions. We report results on a simple rectangular plate model (for calibration) and an industrial example.

Key words: Composite materials, compression resin transfer moulding, finite elements, multi-objective optimisation, genetic algorithm.

1 Introduction

1.1 The Compression Resin Transfer Moulding Process for Composite Materials Manufacturing

Composite materials are commonplace in everyday life: Concrete, milk bottles, sporting equipment, car bodies and spacecraft are just a few examples where composite materials are used. Composite materials are made by combining two or more materials which have different properties, but where the resultant material has more desirable properties than either of the individual constituents. The two types of material that make up composites are called *matrix* and *reinforcement*. In this paper we consider the manufacturing of polymer composites by a liquid composite moulding process. For more information on composite materials the reader is referred to Polymer Science Learning Center, The University of Southern Mississippi (2005)

Although many materials can be used as reinforcement, glass fibres are by far the most common. While glass is usually brittle, it is strong and flexible when spun into fibres. For some composite parts that must be able to sustain extremely high stress such as aircraft parts, stronger but more expensive reinforcements such as Kevlar and carbon fibres are used. A reinforcement material that has undergone initial shaping but has not been processed into the final part is called a preform.

The matrix holds fibre reinforcements together and adds toughness to the fibres as it can absorb energy by deforming under stress and two common types of polymer resins are used as the matrix, namely *thermosets* and *thermoplastics*. Thermoset resins are liquid at room temperature. They solidify after the chemical/thermal activation known as curing. The process cannot be reversed so that the material will not return to a liquid state even under high temperature. Thermoplastics are hard at room temperature and need to be melted above their crystallisation temperature in order for them to flow. Thermoplastics are highly viscous fluids once melted. They are cooled to solidify after the composite parts are made. The process is thus reversible.

The Liquid Composite Moulding Process (LCM) is a type of manufacturing process that is able to mass produce composite parts. Initially, the fibrous reinforcement is manufactured into a perform and placed inside a mould. The mould is then closed and compacted to allow resin to be injected. After the injection process finishes, the resin is allowed to cure and the part is demoulded once sufficient rigidity is reached Kelly and Bickerton (2009). We consider two of the variants of LCM, namely Compression Resin Transfer Moulding (CRTM) and its special case Resin Transfer Moulding (RTM).

The CRTM process proceeds in the following stages.

- Stage 1: Preform Manufacture and Lay-up. First, the reinforcement is manufactured into perform and laid inside the mould.

- Stage 2: Initial Dry Compaction. The mould is closed to some desired height. At the end of dry compaction, the mould usually remains partially open. This will allow resin to flow through the mould with relative ease due to the low fibre volume fraction.

- Stage 3: Resin Injection. The resin is injected into the mould with constant pressure or constant flux. The injection process will stop once the required volume has been reached.

- Stage 4: Final Wet Compaction. In the wet compaction stage, the mould is closed down fully until the final desired part thickness is reached. The wet compaction stage is usually carried out with constant velocity or constant force. The resin inside the cavity will be forced through the remaining dry region of the preform, fully saturating the part.

- Stage 5: Curing and Removal. The part is cured and demoulded once sufficient rigidity has been attained.

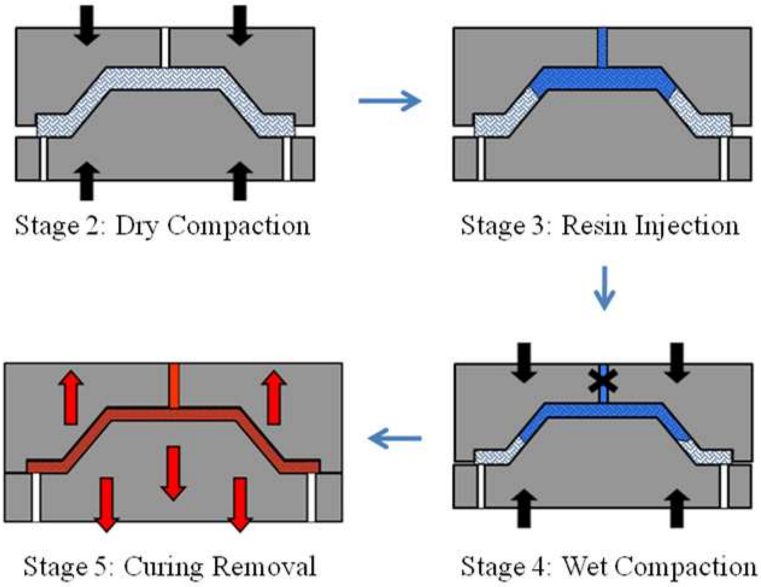


Figure 1: LCM process: Stages 3 to 5.

Stages 2 to 4 are the main focus of the optimisation of the CRTM process. Figure 1 shows the CRTM process from Stage 2 to Stage 4.

The Resin Transfer Moulding (RTM) process is a special case of the CRTM process. However, in Stage 2 of the RTM process, the mould is always closed to the final part thickness. This results in a high fibre volume fraction inside the cavity before resin injection starts and thus a typical RTM process requires more processing time than a CRTM process.

1.2 Simulation Software

To simulate the CRTM and RTM processes we use a generic LCM process simulation software called SimLCM developed at the University of Auckland Kelly and Bickerton (2009). SimLCM uses the finite element methods to solve the differential equations of the mixed elastic compaction model that describe the fluid flow through the mould, the stresses taken by the preform reinforcements and the permeability of the material. During simulation, the finite element meshes of the composite part are input into SimLCM. SimLCM simulates the moulding process according to the specified manufacturing parameters as well as the given part material and geometry. As a result of the simulation, quantities of interest such as the process time and maximal tooling force can be observed.

2 Optimising the CRTM Process

2.1 The Optimisation Problem

Optimisation of the moulding process may consist of a variety of processing objectives and parameters. Common objectives are minimising processing time, tooling force, void content and injection pressure. Processing parameters include the injection pressure, compression velocity, gate location and materials etc.

In this paper we consider the objectives processing time T (because shorter process time obviously allows higher production rates) and the maximal tooling force

F_{max} required to close the mould during the process (because this determines the required capabilities of the manufacturing equipment and therefore its cost). It is desirable to keep both T and F_{max} at a minimum. Unfortunately, these goals contradict each other, because shorter process times can be achieved by increasing the tooling force.

T and F_{max} are largely determined by three important manufacturing parameters, namely the injection pressure P_{inj}, the wet compaction velocity V_{wet} and the mould height H before the wet compaction stage. Although the dry compaction velocity also has an influence it was found that it can be neglected. For any given values of P_{inj}, V_{wet} and H, SimLCM can be run to obtain the values of T and F_{max}. The values of the three parameters need to be restricted to reasonable ranges. In this paper we use typical values that occur in practice, i.e. P_{inj} between 100 and 450 kPa, V_{wet} between 0.000008333 and 0.000416667 m/s and H any value between the final part thickness and the initial preform thickness.

In an abstract form we therefore want to solve the following bi-objective otpimisation problem:

$$\begin{array}{rl}
\text{minimise} & (T, F_{max} = f(P_{inj}, V_{wet}, H) \\
\text{subject to} & 100 \leq P_{inj} \leq 450, \\
 & 8.333 \times 10^{-6} \leq V_{wet} \leq 4.167 \times 10^{-4}, \\
 & H_I \leq H \leq H_f.
\end{array}$$

Note that in the special case of the RTM process the preform is compressed to the final part thickness in the dry compaction phase. This means that $V_{wet} = 0$ and $H = H_f$ and hence the optimisation has only one variable, namely P_{inj}.

There is not much literature on the optimisation of LCM manufacturing processes. Lin *et al.* (2000) investigate two cases of RTM process optimisation. In the first one they determine the optimal gate location to minimise the filling time and in the second one they vary the permeability of layers to minimise resin waste in addition to filling time. A genetic algorithm (GA) and the quasi-Newton method were tested. Lin *et al.* (2000) show that the GA has a poor rate of convergence compared with the quasi-Newton method. Kim *et al.* (2000) also investigate minimising RTM filling time by finding optimal gate locations. Numerical simulation and optimisation are conducted on a complex geometry, an automobile bumper core. It is shown that the filling time can be reduced by about 100 seconds using a GA. Na (2008) formulates the CRTM process optimisation problem using the same variables and (a weighted sum combination of the) objectives as in this paper. The study reveals that the optimisation problem is non-convex and a global optimisation heuristic is created from the problem pattern analysis. Kam (2009) in a preliminary study uses a GA to solve the CRTM process optimisation problem (1). While the algorithm is not fine-tuned for the problem, he demonstrates that the GA is able to find a set of efficient solutions.

2.2 Multi-objective Optimisation with Genetic Algorithms

2.3 Multi-objective Optimisation

As we have seen in Section 1.1, the optimisation of the CRTM process can be formulated as a bi-objective optimisation problem. In this section, we give a brief introduction to multi-objective optimisation and genetic algorithms for their solution, as well as the framework we use to solve the CRTM optimisation problem (1).

A mutli-objective optimisation problem can be written as

$$\begin{aligned} \text{minimise } f(x) &= (f_1(x), \ldots, f_p(x)) \\ \text{subject to } x &\in X, \end{aligned}$$

where X is some feasible set. We say a solution $x^* \in X$ is efficient if there is no other $x \in X$ such that $f(x) \leq f(x^*)$ and $f(x) \neq f(x*)$. Hence the goal of multi-objective optimisation is to find efficient solutions. The objective function vectors $y^* = f(x^*)$ of efficient solutions are called non-dominated points. $Y = f(X)$ is the feasible set in objective space and $Y_N = \{y^* = f(x^*) : x^* \text{ is an efficient solution}\}$ is the set of all non-dominated points. Since multiple efficient solutions can map to the same non-dominated point, the goal of multi-objective optimisation is usually more precisely defined as finding Y_N, and for each $y \in Y_N$ some solution x with $y = f(x)$.. Multi-objective optimisation algorithms therefore focus on the objective space, i.e. finding Y_N. For further details on multi-objective optimisation see Ehrgott (2005).

In the case that X is a continuous set and f is a continuous function, Y_N is usually infinite and in all but the most basic cases (X is a polyhedron and f is linear) it is not possible to obtain an exact description of Y_N. This problem is aggravated if f is not convex. Hence approximation methods or heuristics have to be used to solve multi-objective optimisation problems. Such algorithms aim to find a finite set of solutions $X' \subset X$ such that the finite set $Y' \subset Y$ has two properties:

1. Each element of Y' is close to Y_N.

2. The set Y' covers the entirety of Y_N.

While approximation algorithms such as Ehrgott (2005) do have some guarantee on their performance with respect to these principles, heuristics do not.

In many practical applications an additional problem arises, namely that the objective function f is not known analytically, but only through simulation or other black box evaluation that may require long computation times. This is of course the case in our problem (1), where for given P_{inj}, V_{wet} and H a run of SimLCM is needed to obtain T and F_{max}. In this case, mathematical optimisation methods that require information about f are not applicable and heuristics are used. The most popular heuristics for multi-objective optimisation are evolutionary algorithms, also called evolutionary algorithms.

2.4 Evolutionary Multi-objective Optimisation

An evolutionary algorithms (EAs) mimics the principles of biological evolution. It works with a population of solutions of size N and, in each iteration, evaluates the "fitness" of the individuals in the population, modifies the individuals ("parents") to create "offspring" through mutation and recombination operators, and decides which of the parent and child individuals survive into the next generation according to the principle of survival of the fittest. A typical EA proceeds as shown in 1.

In this paper we use the very popular NSGA-II (Non-dominated Sorting Genetic Algorithm) described in Deb *et al.* (2002). The algorithm is defined by the specific choices in each of the steps of Algorithm 1. Fitness assignment is done as follows. Given a population of parents and offspring, the individuals that are not dominated by any other individual are assigned rank 1 (the fittest). These individuals are then removed and the procedure is repeated to find individuals of rank 2 etc. For selection of individuals for the next generation the so-called crowding distance is used in

Algorithm 1 Pseudocode for a typical EA.

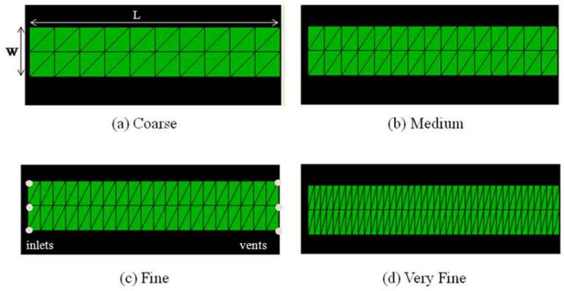
```
(Randomly) generate initial population
repeat
    Assign fitness values
    Select parent individuals
    Variate parent individuals to produce offspring individuals
    Select N fittest individuals as the population for the next generation
until termination criteria met
```

addition. This is the average distance of an individual to its nearest neighbour over all objectives. Individuals are compared according to rank and crowding distance, where individuals with lower rank are preferred (fitter), and in case of ties the larger crowding distance is preferred, as individuals with larger crowding distance are in "less crowded" areas of the non-dominated set. Individuals are added to the population for the next generation until the population size N is reached. This procedure is aimed at preserving the fittest individuals while at the same time ensuring diversity of the population.

Parent selection follows a roulette-wheel selection process. Then three different variation operators are used. Since we cannot describe them in detail here, we refer to the literature. We use two recombination operators, namely uniform crossover (Zitzler *et al.*, 2000), simulated binary crossover (SBX) (Deb and Agrawal, 1994), and one mutation operator, the polynomial mutation operator proposed by Deb and Goyal (1996).

Finally, we need to mention the termination criteria. We imposed a maximum number of generations. The algorithm will stop as soon as this maximum m is reached. In order to be able to control convergence, we also implemented two common measures to compare populations of solutions. The hypervolume indicator (Zitzler and Thiele, 1999) measures the dominated region of the objective space that is bounded by a reference point which is at least weakly dominated by the non-dominated front. In our problem, the larger the bounded hypervolume, the better the non-dominated front. The algorithm is terminated if the percentage increase of the hypervolume indicator falls below some threshold. We found the highest F_{max} value by using SimLCM with the highest P_{inj}, V_{wet} and lowest H value and the highest T value by suing SimLCM with opposite settings. The second measure is the binary epsilon indicator of Zitzler *et al.* (2003). Given two sets of individuals A and B, where A dominates B, it calculates the smallest value ϵ by which each element of B can be multiplied so that every point in B is weakly dominated by some element of A. We use it to compare the non-dominated elements of subsequent generations and terminate as soon as it is close enough to 1. During initial test we noticed that Using the epsilon indicator and hypervolume measures as convergence criteria may cause problems during the first few generations, when the set of non-dominated individuals (those with rank 1) does not change between two generations. We avoid this by imposing a minimum number of generations.

The implementation was carried out in the PISA (Platform and programming language independent Interface for Search Algorithms) framework of Bleuler *et al.* (2003). This framework allows to separate problem specific components of the implementation, such as the variation mechanisms from the problem independent ones such as selection mechanisms, which only require information about objective values. PISA divides the implementation of an optimisation method into the *Selector module* and the *Variator module*. The Selector module contains the selection mech-



(a) Coarse (b) Medium

(c) Fine (d) Very Fine

Figure 2: Four meshes for the flat rectangular plate model used in CRTM simulation.

anisms and algorithms while the Variator module contains the problem specific part: the representation of individuals, the generation of new individuals and the calculation of objective function values Zitzler *et al.* (2004). The two modules are compiled independently and communicate via a text-based interface.

3 Case Studies

In this section we report results of two case studies, first a simple rectangular plate and second an industrially manufactured object, namely a fireman's helmet.

3.1 Rectangular Plate

A convergence analysis was carried out observing the fluid force and comparing it with the analytical solution, which is known for a simple rectangular plate model. After running SimLCM with four different grades of mesh (coarse, medium, fine, and very fine, see Figure 2) and observing that the finer the mesh, the more accurate the simulation works, but the higher the computation time it was decide to work with the fine mesh.

We first report results for the RTM process. Recall that here $H = H_f$ and $V_{wet} = 0$, so there is only a single variable P_{inj}. We included V_{dry} to estimate its influence on T and F_{max}. The set-up of SimLCM used a fine mesh with 63 nodes and 80elements, three resin injection nodes (inlets) at one end and three vents on the opposite side (see Figure 2) and appropriate material parameters. The initial thickness of the glass fibres is $0.0063m$ and they will be compressed to the final thickness of $0.0035m$.

The range for P_{inj} is set between 100 kPa and 450 kPa and the range for V_{dry} is set between 0.5 mm/min and 25 mm/min and the GA parameters are a maximum of 60 iterations with population size 40. In each iteration, 20 individuals are selected as parents and 20 offspring will be generated. The tolerance for the hypervolume and epsilon indicator values are set to be 0.001 and 0.9, respectively.

The GA optimisation run terminates after 10 generations when the hypervolume indicator value is reduced to 0.00257 and the epsilon indicator value is 0.977. We compared the results of the GA with runs of SimLCM with different P_{inj} from 100 kPa to 450 kPa with 50 kPa step and fixed V_{dry}. The results are plotted on top of the population plot at the 10th generation of the GA in Figure 3).

This shows that $P_{inj}$is the key determining variable in the RTM process and that the same force is induced regardless of the compaction speed V_{dry}.

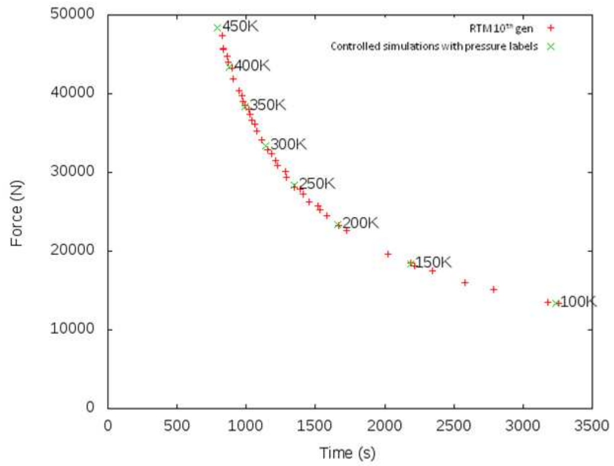


Figure 3: Plot of population at the 10th generation and the RTM simulations with 9 different pressure values.

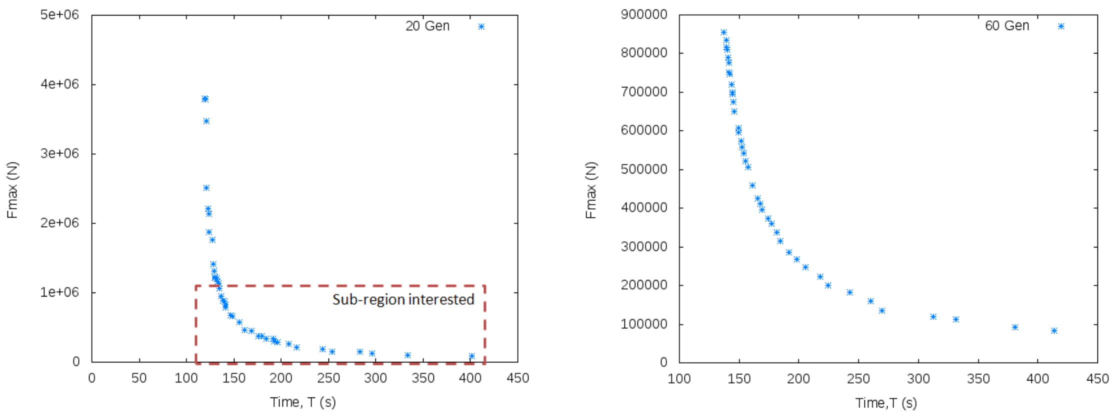


Figure 4: Populations of the 20th and 60th (final) generations of Tests 1 (left) and 2 (right).

Next, we report the results of the CRTM optimisation with the same rectangular plate and the same material parameters as before. Also, the GA parameters are the same as before. We perform two tests with different variable ranges. In both P_{inj} varies between 100 and 450 kPa. In the first, we have $0.5 \leq V_{wet} \leq 25mm/min$ and $0.05 \leq H \leq 2.85mm$ and in the second one $0.5 \leq V_{wet} \leq 5.4mm/min$ and $1 \leq H \leq 2.85mm$.

The first test terminates after the 20th generation with the hypervolume indicator and epsilon indicator values 0.001213 and 0.94. The population of the 20th generation is displayed on the left of Figure 4. Some of the individuals in this population have F_{max} values up to $4 \times 10^6 N$. In real life, it is unlikely that flat plate is manufactured with such a high force value. The sub-region with lower F_{max} and higher T values is of a higher interest. The individuals around that region are observed to have high P_{inj}, high H and low V values. Therefore we run the second test with the parameters specified before. The program terminates at the maximum number of 60 generations. The plot of the final generation from the second optimisation is displayed on the right of Figure 4.

We also compared the optimisation results with runs of SimLCM with the lowest

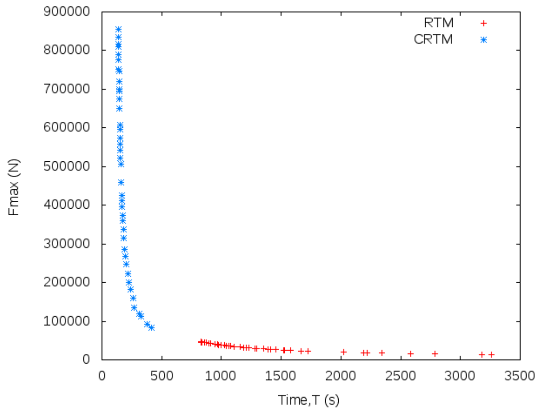


Figure 5: Plot of final CRTM and RTM populations.

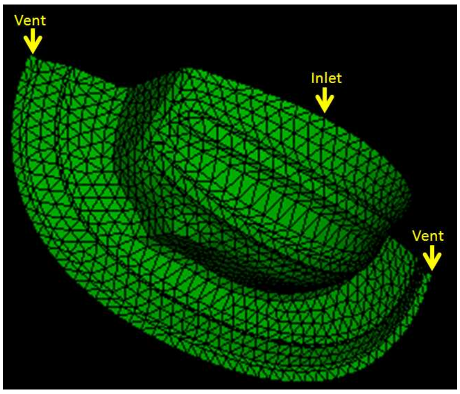


Figure 6: The medium mesh of a fireman's helmet.

P_{inj}, lowest V_{wet} and five different H values as well as with the highest P_{inj}, lowest V_{wet} and the same five values of H. In every case, the optimisation found solutions dominating all of the fixed variable simulations, clearly demonstrating the value of optimisation.

To conclude this section, we show the combined results for the RTM and CRTM processes. As expected, these confirm that RTM is a special case of CRTM, see Figure 5.

3.2 Fireman's Helmet

This study looks at a fireman's helmet manufactured by Pacific Helmets Ltd, of Wanganui, New Zealand. In this case we used a medium mesh to reduce computation time for SimLCM. This mesh has 988 nodes and 1862 elements. A run of SimLCM with $P_{inj} = 100\ kPa$, $V_{wet} = 10\ mm/min$ and $H = 1\ mm$ requires about 238 seconds on a computer with Intel® Xeon® 2.67 GHz processor to achieve an accuracy within 3 % of that of a fine mesh of 1707 elements requiring 1284 seconds.

The initial total preform thickness is 0.00714 m and the final part thickness at the end of moulding process is 0.002 m. The helmet has a length of 0.42 m, width of 0.32 m, and a depth of 0.17 m. The resin injection node is at the center top of the helmet and there are two vents at the sides (see Figure 6).

The variable ranges and GA parameters are the same as those used in Section 3.1. The termination criteria are met when the hypervolume indicator value reaches

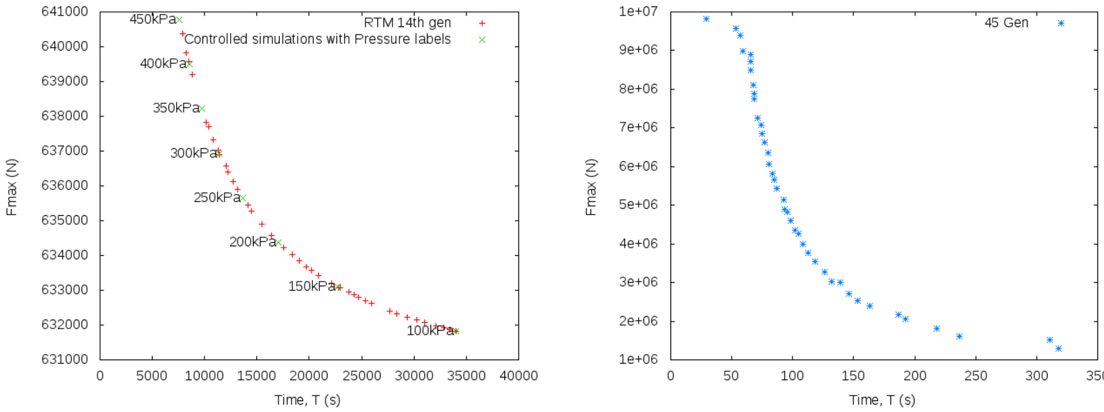


Figure 7: Populations of the 14th and 45th (final) generations of the RTM (left) and CRTM (right) optimisations.

0.0001 and when the epsilon indicator value reaches 0.95.

The procedure was as in Section 3.1. We optimised the RTM process first, comparing the results with runs of SimLCM with fixed P_{inj} to verify the optimisation. The optimisation process terminates after the 14th generation. The hypervolume indicator value and the epsilon indicator values are 0.998856 and 0.000008, respectively. The final population is shown on the left of Figure 7.

Second, we optimised the CRTM process using the same parameters as before once again. At termination of the optimisation, the hypervolume indicator value equals 0.000048 and the epsilon indicator value reaches 0.97. The plot of the 45th and final generation is given on the right of Figure 7.

4 Conclusion

In this paper we have described a framework for the optimisation of liquid composite manufacturing processes. We have focused on the RTM and CRTM processes and chosen the objectives of minimising the maximal tooling force F_{max} and the processing time T depending on the variables of injection pressure P_{inj}, wet compaction velocity V_{wet} and injection height H as variables. The optimisation was carried out using an evolutionary muti-objective algorithm using the finite element simulation software SimLCM for function evaluations. This was implemented within the PISA framework for ease of use and future modification. We have demonstrated that optimisation is valuable, producing better solutions than simulation with fixed variable values alone. Moreover, the illustration of a set of non-dominated solutions provides valuable information to manufacturers for setting up their processes and possibly for investment decisions. Further analysis of the solutions obtained provides valuable insights into the LCM processes and on how the objectives depend on the variables.

References

Bleuler, S., Laumanns, M., Thiele, L., and Zitzler, E. (2003). PISA – A platform and programming language independent interface for search algorithms. In C. Fonseca, editor, *Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, volume 2632 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin.

Deb, K. and Agrawal, R. B. (1994). Simulated Binary Crossover for Continuous Search Space. Technical Report IITK/ME/SMD-94027, Indian Institute of Technology Kanpur.

Deb, K. and Goyal, M. (1996). A combined genetic adaptive search (GeneAS) for engineering design. *Computer Sciences and Informatics*, **26**(4), 30–45.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6**(2), 182 –197.

Ehrgott, M. (2005). *Multicriteria Optimization*. Springer Verlag, Berlin, 2nd edition.

Kam, W. (2009). Optimisation of Mould Filling Parameters during the Compression Resin Transfer Moulding Manufacturing Process. Final Year Report, University of Auckland.

Kelly, P. and Bickerton, S. (2009). A comprehensive filling and tooling force analysis for rigid mould LCM process. *Composites Part A: Applied Science and Manufacturing*, **40**, 1685–1697.

Kim, B., Nam, G., Ryo, H., and Lee, J. (2000). Optimisation of filling process in RTM using genetic algorithm. *Korea-Australia Rheology Journal*, **12**(1), 83–92.

Lin, M., Murphy, M., and Hahn, H. (2000). Resin transfer molding process optimisation. *Composites Part A: Applied Science and Manufacturing*, **31**, 361–371.

Na, S. (2008). *Global Optimisation of Mould Filling Parameters during the Constant Speed Injection Compression Moulding Process*. Master's thesis, University of Auckland.

Polymer Science Learning Center, The University of Southern Mississippi (2005). Composites. `http://pslc.ws/macrog/composit.htm`.

Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, **3**(4), 257–271.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, **8**(2), 173–195.

Zitzler, E., Thiele, L., Foneseca, C., and Grunert da Fonseca, V. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, **7**(2), 117–132.

Zitzler, E., Laumanns, M., and Bleuler, S. (2004). A tutorial on evolutionary multiobjective optimisation. In X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'Kindt, editors, *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*, pages 3–38. Springer Verlag, Berlin.

Branch-and-Price Guided Search

Mike Hewitt
Rochester Institute of Technology

George Nemhauser
Georgia Institute of Technology

Martin Savelsbergh
CSIRO Mathematics, Informatics, and Statistics
Martin.Savelsbergh@csiro.au

Jin-Hwa Song
ExxonMobil Research and Engineering

Abstract

We present an exact solution approach for integer programs in which well-chosen restrictions are solved to produce high-quality solutions early in the search. Column generation is used for generating the restrictions and for producing bounds on the value of an optimal solution. A local search scheme is embedded to explore neighbours of the current best solution. The approach is designed to be implemented on a multi-processor architecture. The efficacy of the approach in demonstrated on the integer multi-commodity fixed-charge network flow problem and a complex maritime inventory routing problem with varying storage capacities and production/consumption rates at facilities.

Regression Spline Fitting with Applications

C. G. Walker, M. J. O'Sullivan, Y. Zhu, S. Hsu
Department of Engineering Science
University of Auckland
New Zealand
cameron.walker@auckland.ac.nz

M. L. MacKenzie, C. R. Donovan
School of Mathematics and Staistics
University of St Andrews
Scotland

V. Rajagopal
Auckland Bioengineering Institute
University of Auckland
New Zealand

Abstract

Deciding how best to fit regression splines to data is a difficult non-linear optimization problem. In this paper we present a method for determining good fits using a spacially adaptive local smoothing algorithm (SALSA). We present results that show our method generates models that fit as well as those generated by techniques using smoothing splines, and discuss an application of our technique that enables the automatic landmarking of certain object boundaries.

Key words: Regression Splines, Shape Analysis, Landmarks, Clustering, PCA.

1 Introduction

Splines are a useful function-type to use in Regression when the relationship between a response and a set of covariates is not known *a priori*. Their benefits are well documented (see for example (DeBoor 1978; Schumaker 1993)). In particular, they share many of the nice mathematical properties of polynomials, without the global behaviour that can be problematic when fitting a regression model (often it is required that the change to a value in one region should not influence the fit in another region). The use of splines within the regression framework has been largely influenced by the work of Hastie and Tibshirani (Hastie and Tibshirani 1986; Hastie and Tibshirani 1990) on Generalized Additive Models (GAMs).

Approaches in spline-based regression for balancing fit to the signal (the mean) and fit to the noise (the data variation from the mean) include controlling the number and location of knots (Wold 1974) or including a penalty term in the fitness criteria (Eilers and Marx 1996; Wood 2000; Ruppert 2002). Recent work on the second approach has concentrated on implementating locally adaptive smoothing parameters (Ruppert and Carroll 2000; Baladandayuthapani, Mallick, and Carroll 2005; Crainiceanu et al. 2007; Krivobokova, Crainiceanu, and Kauermann 2008; Wood et al. 2008). Historically the first approach to regression spline fitting, adaptively placing knots, has involved a computer–intensive search. This includes stepwise forward and backward knot selection (TURBO (Friedman and Silverman 1989), MARS (Friedman 1991)), often with guided-search techniques included to reduce the solve time (SARS, (Zhou and Shen 2001)). Bayesian approaches (using the reversible-jump Metropolis-Hastings version of Markov Chain Monte Carlo simulation) have also been implemented in BARS (DiMatteo, Genovese, and Kass 2001; Behseta, Kass, and Wallstrom 2005; Behseta and Kass 2005) and cBARS (Kaufman, Venture, and Kass 2005).

In this paper we first describe a spatially-adaptive local smoothing algorithm (SALSA) (Walker et al. 2010), which automatically chooses the location and number of knots in the spline regression model. This heuristic includes local-search and a restricted forward/backward regression step that significantly reduces the number of models to be evaluated at each iteration, compared to the standard approach (Friedman and Silverman 1989). It performs as well as current alternatives in the literature on established benchmark functions.

Next we explain how this algorithm can be used to automatically determine landmarks on object boundaries, for use in shape analysis. Our approach is demonstrated in two examples. In the first we landmark animal tracks in the ocean (1-dimensional curves in 3-space). In the second we landmark the boundary of nuclei in cardiac cells (2-dimensional surfaces in 3-space).

The paper is set out as follows. In Section 2 we describe how SALSA works, and evaluate it's performance against standard benchmarks functions in the literature in section 3. In section 4 we describe how we have extended SALSA in a couple of applications to perform automatic landmarking, and give some concluding remarks in section 5.

2 Details of SALSA

In this section we provide details of a spatially adaptive local smoothing algorithm (SALSA) for fitting regression splines to data. Our goal is to use a spline to approximate the mean of the response variable at each value (or combination of values) of the explanatory variable(s). This involves deciding the number and location of the knots, as well as the coefficients of the polynomial sections making up the spline. Determining each knot location adds a level of complexity - the minimax polynomial approximation problem can be modelled as a linear program, as opposed to the nonlinear mixed-integer program necessary for a regression spline. Although it is possible to position a knot anywhere in the domain when fitting a regression spline, we consider only data point locations as potential sites, which is standard in practice (see, for example, (Wold 1974; Hastie and Tibshirani 1990)). The fitness measure we use in this paper is the Bayesian Information Criterion (BIC), which can be calculated for a model fit to n data points from the log-likelihood L and the number of parameters p by $-2 \times L + k \times p$, where $k = log(n)$. In this calculation both p and L are variable, depending on the number and location of the knots. The BIC balances improving the fit to the data against increasing the number of knots used.

SALSA iteratively determines regression splines that better fit the data using three steps. The first is a global exchange step, which enables the movement (addition) of a knot to (at) the worst fit data point in the domain. The second step is local, moving knot positions to neighbouring datapoints. The final step simplifies the model by removing knots from the regression spline.

SALSA:

Calculate s equally-spaced locations E between first and last data points
Initialize knots K with s data locations minimizing $\sum_{i=1}^{s} |K_i - E_i|$
Repeat
    Repeat Exchange step *While* (fit measure is improved)
    Repeat Improvement step *While* (fit measure is improved)
    If ($|K| >$ `minKnots`)
        Perform Simplification step
    End If
While (an improvement in fit measure is made by one of the above steps)

Figure 1: Pseudocode outlining the structure of SALSA

The structure of the algorithm is given in figure 1. We have included a number of parameters in the implementation of our algorithm to increase the user's ability to control basic characteristics of the final model.

Two of the parameters we include are:

`maxKnots`	maximum allowable number of knots;
`minKnots`	minimum allowable number of knots.

2.1 Algorithm Details

The specific details of each algorithm step are given in the remainder of this section.

Exhange Step. In the exchange step the location of the data point furthest from the fitted curve is identified, and regression splines are fit by shifting each existing internal knot from its current location to the position of this point. Where the spline contains less than `maxKnots` knots, a further model is fitted, retaining the current knot locations and including a further knot at this new location. All new models are evaluated, and, where an improvement is obtained, the current model is replaced with the best new model. This step is similar to the forward addition step described by Friedman and Silverman (Friedman and Silverman 1989), but restricting the new knot location to a single data point. This approach requires significantly fewer model evaluations per iteration than the standard forward regression approach. The exchange step has worked well on the benchmark functions we have used for testing, despite the restriction of considering only one new knot location.

Improvement Step. This is a local step, which considers relocating each knot, in turn, to each of its neighbouring data points (where possible). Where the best of these new models is better than the current model, the current model is updated accordingly.

Simplification Step. In this step, new models are obtained from the current model by removing each knot in turn and refitting. Where this results in a better fit, the current model is replaced with the best of these new regression splines. This step is just the standard backward deletion (Friedman and Silverman 1989), and is performed only if the regression spline includes more than `minKnots` knots.

3 Performance of SALSA on Benchmark Functions

In this section, we summarise SALSA's ability to fit to known benchmark functions that have low, moderate and high spatially-variable smoothness.

The performance of SALSA on benchmark functions of low and high spatially variable smoothness was evaluated using functions (Equation 1) proposed by Rupert and Carroll (Ruppert and Carroll 2000) with $n = 400$ equally spaced x's on [0,1] and a signal-to-noise (S/N) ratio of approximately 7 ($\sigma_\epsilon = 0.2$). We set $j = 3$ and $j = 6$ (Figure 2) to represent low and high spatial heterogeneity respectively. These benchmarks are subsequently referred to as the $RC_{j=3}$ and $RC_{j=6}$ functions.

$$f(x_i, j) = \sqrt{x(1-x)}\sin\left\{\frac{2\pi(1+2^{(9-4j)/5})}{x+2^{(9-4j)/5}}\right\} \quad (1)$$

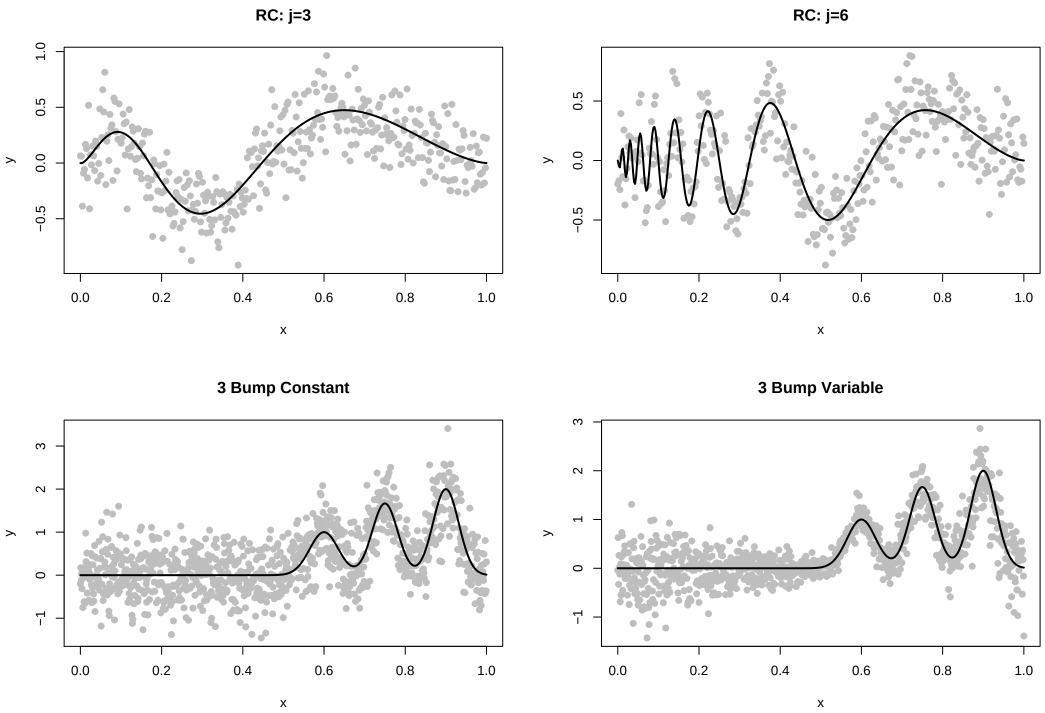


Figure 2: $RC_{j=3}$, $RC_{j=6}$, Three-Bump Constant and Three-Bump Variable data sets. The original function is a solid line, and the simulated data are the points.

The performance of SALSA for moderate spatial heterogeneity was examined using the 'three bump' function (Equation 2, Figure 2). These data sets were generated using equally spaced x's on [0,1] ($n = 1000$) and with error variance $\sigma_\epsilon = 0.5$. We refer to these functions as the 'Three-Bump Constant' and 'Three-Bump Variable' models.

$$f(x_i) = \exp\{-400(x-0.6)^2\} + \frac{5}{3}\exp\{-500(x-0.75)^2\} + 2\exp\{-500(x-0.9)^2\} \quad (2)$$

3.1 Algorithm performance

We generated 100 data sets for each benchmark function and ran the algorithm under two schemes: one with a fixed number of knots; one where the number of knots can vary. The variable-knot scheme included knot addition, deletion and relocation while the fixed-knot scheme only allowed knot relocation (achieved by setting `maxKnots` to be equal to `minKnots`). We used the B-spline basis (DeBoor 1978) in fitting our model. We permitted both quadratic and cubic bases, using the Bayesian Information Criterion (BIC) to select the degree of the spline and the number and location of knots.

The Average Squared Error (ASE, Equation 3) was used to measure the fidelity of the fitted model to the underlying function for each data set and the mean of these ASE values (MASE) across the 100 data sets was calculated.

$$ASE = \frac{1}{n}\sum_{j=1}^{n}\left(f(x_i) - \hat{f}(x_i)\right)^2 \quad (3)$$

Table 1: Mean Average Square Error (MASE) using the three algorithm settings.

Function	Fixed-knot	Variable-knot
$RC_{j=3}$ ($\sigma_\epsilon = 0.2$)	0.00110	0.00127
$RC_{j=6}$ ($\sigma_\epsilon = 0.2$)	0.00452	0.00465
Three-Bump Constant	0.00467	0.00534
Three-Bump Variable	0.00284	0.00332

Under the fixed-knot scheme, SALSA performed at least as well as current spatially adaptive methods for functions with low spatial heterogeneity (Table 1). Our results for the $RC_{j=3}$ function (Equation 1) were very similar to Ruppert and Carroll (Ruppert and Carroll 2000) and Crainiceanu et al. (Crainiceanu et al. 2007), which reported values of 0.0011 and 0.0012 respectively. In contrast, under the variable-knot scheme SALSA returned a MASE that was about 13% larger than when used under the fixed-knot alternative.

Regardless of scheme, SALSA outperformed adaptive alternatives for functions with high spatial heterogeneity. For instance, under the fixed-knot scheme the algorithm returned a MASE for the $RC_{j=6}$ function that was 26% smaller than the algorithms described in (Ruppert and Carroll 2000) and (Crainiceanu et al. 2007) (these reported MASE values of 0.0061 and 0.0065 values respectively). Under the variable-knot scheme our algorithm gave a MASE score 24% smaller than these alternatives.

Under the fixed-knot scheme SALSA also gave smaller MASE values for the Three-Bump Constant function, returning a MASE between 16% and 25% smaller than those reported in (Ruppert and Carroll 2000) (MASE=0.0054), (Crainiceanu et al. 2007) (MASE=0.0055) and (Baladandayuthapani, Mallick, and Carroll 2005)

(MASE=0.0061). Under the variable-knot scheme SALSA gave closer MASE scores to these alternatives.

4 Applications

In this section we describe how to extend the original version of SALSA to fit splines to 1-dimensional curves in 3-dimensional space, and apply this new version of SALSA to automatically landmark sea mammal dives and cell nucleus boundaries.

4.1 SALSA for landmarking curves parametrized by time

Each sea mammal dive is a 1-dimensional curve in 3-dimensional space, so it is necessary to extend the algorithm to accommodate this sort of curve. This is done by parametrizing the x, y and z co-ordinates of the dive by a fourth parameter t (which can be thought of as time in this application, although for an arbitrary closed or open curve this approach will also work). Each of these curves, $x(t)$, $y(t)$ and $z(t)$, is then fitted using the SALSA algorithm, with the regression splines used for the 3 parametric curves constrained to have knots at the same values of t. To fit 3 curves at once the fitness criterion needs to be updated to incorporate a fitness measure from each curve. It is possible to use the residual sums of squares from the 3 paramteric curves to compute a BIC value for the single curve in 3D, but we have found the sum of the BIC values for each paramteric curve to be an effective criterion. The exchange step of SALSA is also updated to identify the value of t that yields the largest magnitude residual across all 3 parametric curves. An example of the approach for the sea mammal dive shown in figure 4 is presented in figure 3.

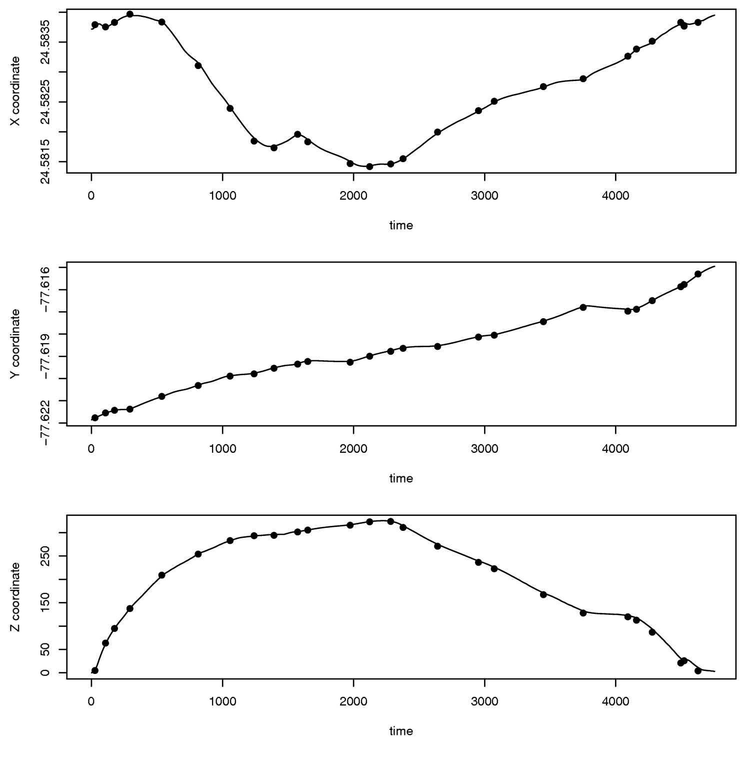


Figure 3: Parametrization of a sea mammal dive - x, y, and z corordinate given in terms of time. A fit is performed using SALSA with degree 1 splines. The solution places 26 knots (shown here as dots), which can be used as landmarks for the original curve in 3D.

The fitted values of the degree d regression splines for $x(t)$, $y(t)$, and $z(t)$ give the coordinates of a degree d spline for the original curve in 3 dimensions. Figure 4 shows a solution for a particular sea mammal dive. The grey curve is the original sea mammal dive, and the black curve is the fitted regression spline, with the black dots showing the location of the knots.

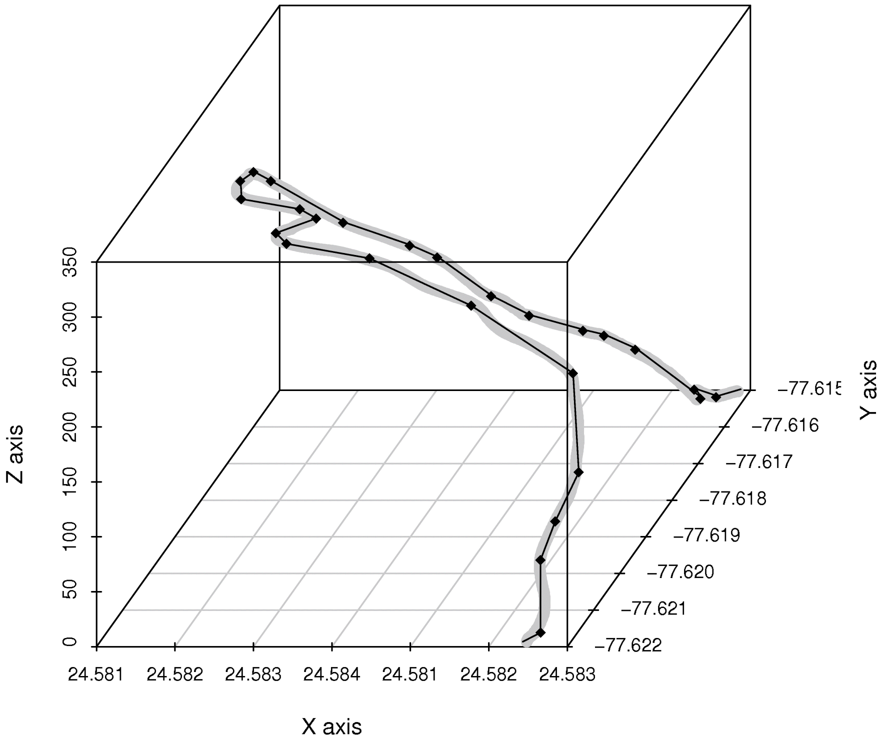


Figure 4: A sea mammal dive (in grey) and the 1d spline fitted using SALSA (in black). The knots for the spline (shown as black dots) can be used as landmarks for a PCA analysis.

The knot points of the regression spline fitted by the parametric SALSA can be used as landmarks for an analysis of the dive shape. SALSA aims to fit the model with the lowest BIC, so it is extremely unlikely that all dives being compared will generate the same number of landmarks (a requirement for the subsequent PCA). To deal with this we do an initial fit for all dives to determine the maximum number of knots K fitted across all the resulting regression splines. SALSA allows the user to define the maximum number (**max**) and minimum number (**min**) of knots to be permitted when fitting a spline. By computing a second fit for all dives with both **max** and **min** set to K we are able to produce a comparable set of landmarks across all the dives. This approach ensures we have chosen the minimum number of landmarks for all dives while still capturing sufficient shape information to satisfy the BIC criterion. For some dives we will capture more shape information than is required, but for our application this is not a problem - one could implement other approaches if desired, such as fixing the number of knots to be the minimum or average across all the fits from the first application of SALSA.

4.2 SALSA for slicing and landmarking 3d objects

To landmark the boundary of a 3-dimensional object, such as the nucleus of a cardiac cell, we have adapted SALSA further. The nucleus is aligned along a “major” axis and then the boundary segmented by intersecting n planes with the object boundary. The first plane is chosen to include the major axis (and an orthoganol "minor" axis), and each subsequent plane is determined from its predecessor by rotating $\frac{180}{n}$ degrees clockwise around the major axis. This process results in $2n$ equally spaced curves on the object boundary, running from the top of the object (with reference to the minor axis) to the bottom. We use our further adapted version of SALSA to fit degree 1 splines to these $2n$ curves simultaneously, each with knots at the same k heights. Note: it is possible that the object shape may be such that a given boundary curve has more than one point at a given domain value (height) – a situation not possible when a curve is parametrized by time. Thus, although each spline will have knots positioned at the same k heights, a given spline may have more than k knots, and hence the $2n$ splines need not all have the same number of knots. This process is performed to identify k heights at which to take 2-dimensional cross-sections of the object. These heights should capture the shape well (althought the equal spacing of the boundary segments means the choice will not be "optimal"). The approach described in subsection 4.1 is then used to landmark the boundaries of the k 2-dimensional cross-sections. An example of landmarked nucleus is shown in figure 5.

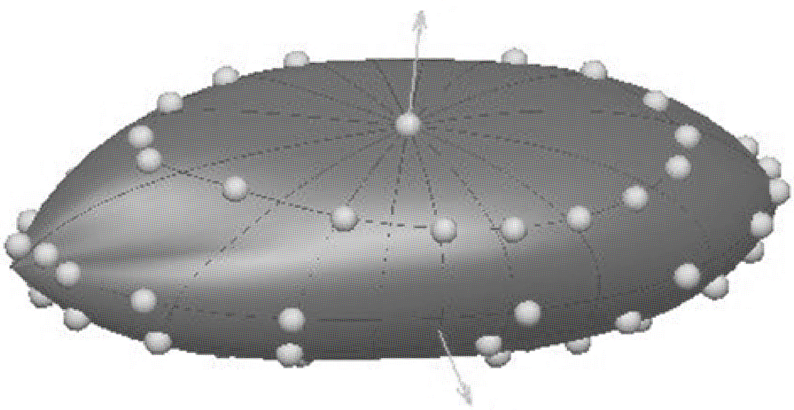


Figure 5: A nucleus from a cardiac cell with landmarks determined using SALSA. The algorithm determined 5 heights at which to slice the nucleus, and fitted splines with 16 knots to each cross-sectional slice.

5 Conclusion

In this paper we have described an algorithm for fitting regression splines to data, and described how extensions of this algorithm can be used to automatically generate landmarks for object boundaries, or 1-dimensional curves (such as animal tracks) in 3-dimensions. In the applications discussed landmarking was the initial step in a more comprehensive study. For the analysis of animal tracks a Principle Components Analysis (PCA) was carried out on the landmarks for a sample of animal dives, and clustering analysis performed on the PCA scores to cluster the dives into types. Of particular interest was identifying whether ensonification affected animal behaviour. For the analysis of cardiac cell nuclei PCA was also used to identify the main modes of variation across a sample of nuclei. The longer term goal is to determine whether there is a difference in nucleus shape for cells from healthy and diabetic tissue.

In both cases landmarking was performed by fitting degree 1 splines using extensions of SALSA (Walker et al. 2010). In all applications the number of landmarks could be determined by the algorithm, given a measure of fit to optimize (for example, the BIC). Alternatively, if only a certain number of landmarks are wanted, parameters could be set to obtain the best fit using only that number of knots in the resulting regression splines.

References

Baladandayuthapani, V., B. K. Mallick, and R. J. Carroll. 2005. "Spatially Adaptive Bayesian Penalized Regression Splines (P-splines)." *Journal of Computational and Graphical Statistics* 14 (2): 378–394.

Behseta, S., and R. E. Kass. 2005. "Testing equality of two functions using BARS." *Statistics in Medicine* 24:3523–3534.

Behseta, S., R. E. Kass, and G. L. Wallstrom. 2005. "Hierachical models for assessing variability among functions." *Biometrika* 92:419–434.

Crainiceanu, C. M., D. Ruppert, R. J. Carroll, A. Joshi, and B. Goodner. 2007. "Spatially Adaptive Bayesian Penalized Splines With Heteroscedastic Errors." *Journal of Computational and Graphical Statistics* 16 (2): 265–288.

DeBoor, C. 1978. *A Practical Guide to Splines*. New York: Springer-Verlag.

DiMatteo, I., C. R. Genovese, and R. E. Kass. 2001. "Bayesian curve-fitting with free-knot splines." *Biometrika* 88:1–67.

Eilers, P. H. C., and B. D. Marx. 1996. "Flexible smoothing with -splines and penalties." *Statistical Science* 11 (2): 115–121.

Friedman, J. H. 1991. "Multivariate adaptive regression splines." *The Annals of Statistics* 19:1–67.

Friedman, Jerome H., and Bernard W. Silverman. 1989. "Flexible Parsimonious Smoothing and Additive Modeling." *Technometrics* 31 (1): 3–29.

Hastie, T., and R. Tibshirani. 1986. "Generalized Additive Models." *Statistical Science* 1 (3): 297–310.

———. 1990. *Generalized Additive Models*. New York: Chapman & Hall.

Kaufman, C. G., V. Venture, and R. E. Kass. 2005. "Spline-based non-parametric regression for periodic functions and its application to directional tuning of neurons." *Statistics in Medicine* 24:2255–2265.

Krivobokova, T., C. M. Crainiceanu, and G. Kauermann. 2008. "Fast adaptive penalized splines." *Journal of Computational and Graphical Statistics* 17:1–20.

Ruppert, D. 2002. "Selecting the Number of Knots for Penalized Splines." *Journal of Computational and Graphical Statistics* 11 (4): 735–757.

Ruppert, D., and R. J. Carroll. 2000. "Spatially-Adaptive Penalties For Spline Fitting." *Australian and New Zealand Journal of Statistics* 42 (2): 205–223.

Schumaker, Larry L. 1993. *Spline Functions: Basic Theory*. Malabar, Fla.: Krieger.

Walker, C. G., M. L. Mackenzie, C. R. Donovan, and M. J. O'Sullivan. 2010. "SALSA a spatially adaptive local smoothing algorithm." *Journal of Statistical Computation and Simulation* http://www.informaworld.com/10.1080/00949650903229041:1–13.

Wold, Svante. 1974. "Spline Functions in Data Analysis." *Technometrics* 16 (1): 1–11.

Wood, S. A., R. C. Kohn, R. Cottet, W. Jiang, and M. Tanner. 2008. "Locally adaptive nonparametric binary regression." *Journal of Computational and Graphical Statistics* 17:352–372.

Wood, S.N. 2000. "Modelling and Smoothing Parameter Estimation with Multiple Quadratic Penalties." *J. R. Statist. Soc. B* 62 (2): 413–428.

Zhou, S., and X. Shen. 2001. "Spatially Adaptive Regression Splines and Accurate Knot Selection Schemes." *Journal of the American Statistical Association* 96 (453): 247–259.

Optimal Paths in Real Multimodal Transportation Networks: An Appraisal Using GIS Data from New Zealand and Europe

Felipe Lillo, Frederik Schmidt
School of Computing and Mathematical Sciences
AUT University
Auckland, New Zealand
{flillovi,fschmidt}@aut.ac.nz

Abstract

The most convenient route to connect two locations is often a mix of different transportation systems. For instance, a user can make intercity trips by selecting a combination of transport modes such as car, rail, ship or airplane. In this case the transportation system is said to be multimodal. In this paper, real multimodal transportation systems are experimentally analyzed. Real road–rail networks from Denmark, Hungary, Spain, Norway and New Zealand are built based on a set of digitized transportation maps obtained from several GIS libraries. These networks are modelled as coloured–edge graphs to be used as main input by a multimodal Dijkstra's algorithm that computes a set of optimal paths. The cardinality of the resulting set is at the core of the approach tractability. It is concluded that vertices connectivity and network shape considerably affect the total number of optimal paths.

Key words: Multimodal network, coloured–edge graph, GIS libraries, transportation networks.

1 Introduction

A multimodal network (MMN) is a transportation system that considers two or more transport choices for connecting locations (vertices) in a network. Freight and urban transportation are two application fields in which multimodal networks are extensively found. In (Bontekoning, Macharis, and Trip 2004) freight transportation is thoroughly reviewed for identifying possible research trends. The authors here conclude research in multimodal freight transportation is still in a pre-paradigmatic phase. Urban transportation deals with the movement of passengers in urban areas considering variables such as congestion levels (flow), public fares, transport modes (bus, subway, private car or bicycle), service demand and user behaviour. Some overviews about urban transportation are found in (Boyce 2007), (Lee 1994), (Wegener 1994) and (Nagurney 1984). Despite their popularity as multimodal research field, freight and urban transportation are not the only areas where multimodal networks can be used as a modelling tool. Application studies into computer networks (Nigay and Coutaz 1993), biomedicine (Heath and Sioson 2007) and manufacturing (Medeiros et al. 2000) can be noticed in the literature too.

One problem associated to MMN is the determination of a shortest path from an origin to a destination. The shortest path problem has been extensively studied in both practice and theory. Likewise, a large number of algorithms have been developed by different fields of enquire such as operation research and computing sciences. When dealing with algorithms for multimodal networks, researchers and practitioners typically opt for techniques based on variations of classical approaches such as Dijkstra's algorithm or Bellman–Ford method (a description of these approaches is found in (Lawler 2001)). However, a direct application of these approaches generates a shortest path that does not consider the multimodal traits of the network as part of the analysis. Thereby, an optimal combination of means of transport is an outcome of the shortest path itself. Consequently, a multimodal network cannot be treated as a "unimodal" because each edge includes an additional variable (the mode) that has to be included as part of the analysis.

An appraisal is carried out in this paper to analyze the comportment of the shortest path problem in real MMNs. The employed modelling approach takes a coloured–edge graph that represents modes, cities and intercity links by colours, vertices and edges. This graph is used by a generalization of Dijkstra's algorithm that computes a set of optimal paths. This set of paths is the key of the model's tractability so that experiments are set for tracing its value. Unlike others approaches, the presented modelling and computational techniques are able to keep the multimodal traits of the network throughout the analysis. The results indicate that the total number of optimal paths is significantly influenced by the shape and connectivity of the network.

The remainder of this paper is organized in four further sections. First the modelling approach and the algorithm are described in section 2. The experimental setup is described in section 3. Results are given in section 4. Finally, section 5 provides conclusions and future work in this research field.

2 Model and Algorithm

2.1 The Coloured–Edge Graph Model

The coloured–edge graph is a modelling tool introduced by (Ensor and Lillo 2009) for the modelling of MMNs. This graph modelling approach labels edges with colours for representing a specific attribute such as a transport mode. In a real transportation system, two locations might be connected by several modes. For this case, the coloured–edge graph allows the use of multiple edges. These edges can be directed or undirected as well as weighted or unweighted. In this research weighted coloured–edge digraphs are utilized to model real multimodal transportation networks from several countries.

In their work (Ensor and Lillo 2009) formally define a weighted coloured–edge graph as $G = (V, E, \omega, \lambda)$ which is a directed graph (V, E) with vertex set V and edge set E, a weight function $\omega\colon E \to \mathbb{R}^+$ on edges, and a colour function $\lambda\colon E \to M$

on edges. M is a finite set of colours with $k = |M|$. Each edge $e_{uv} \in E$ joining vertices u and v has a positive weight $\omega(e_{uv})$ and a colour $\lambda(e_{uv})$. For any colour $i \in M$ and for any path p_{uv} between two vertices u and v, the path weight $\omega_i(p_{uv})$ in colour i is defined as $\omega_i(p_{uv}) = \sum_{e_{uv} \in p, \lambda(e_{uv})=i} \omega(e_{uv})$. The total path weight is represented as a k-tuple $(\omega_1(e_{uv}), \ldots, \omega_i(e_{uv}), \ldots, \omega_k(e_{uv}))$, giving the total weight of the path in each colour.

Only limited research has been conducted in the area of coloured–edge graphs. (Climaco, Captivo, and Pascoal 2010) studied the number of spanning trees in a weighted graph whose edges are labelled with a colour. This work defines weight and colour as two conflicting criteria. Hence, the proposed algorithm generates a set of nondominated spanning trees. The computation of coloured paths in a weighted coloured–edge graph is investigated by (Xu et al. 2009). The main feature of their approach is a graph reduction technique based on a priority rule. This rule basically transforms a weighted coloured–edge multidigraph into a coloured–vertex digraph by applying algebraic operations upon the adjacent matrix. Additionally, (Xu et al. 2009) provide an algorithm to identify coloured source–destination paths. Nevertheless, the algorithm is not intended for general instances because it just works with the number of edges as a path weight. Furthermore, only paths having consecutive edges with distinct colours are considered.

2.2 Multimodal Dijkstra's Algorithm

A weighted coloured–edge graph can produce a factorial number of source–destination paths. For instance, the total number of paths in a complete coloured–edge graph is $O\left(k^{n-1}(n-2)!\right)$. This can be proved by applying a basic counting argument. However, the question is how many of these paths are optimal (shortest). At first glance, traditional shortest path algorithms might be able to provide an answer. Nevertheless, these procedures were originally designed for "unimodal" networks. Thus, a direct application of such algorithms on a coloured–edge graph produces outcomes that do not take transport modes into consideration. This section explain a procedure designed by (Ensor and Lillo 2009) that extracts a set of optimal path from a weighted coloured–edge graph base on a general version of the well–known Dijkstra's algorithm. Firstly, some notation need to be introduced.

Let $\mathcal{M}_{uv}$ be the set of shortest paths joining two vertices u and v in a weighted coloured–edge graph. This set can be extracted from a weighted coloured–edge graph by defining a partial order relation on weight tuples. As a result, a smaller set of paths is obtained. Each k–tuple in $\mathcal{M}_{uv}$ is Pareto efficient which means that a weight in a tuple cannot be improved (or worsened) without worsening (or improving) the same weight of another. The cardinality of $\mathcal{M}_{uv}$ is a key factor in determining the tractability of the model.

The computation of $\mathcal{M}_{uv}$ is performed in (Ensor and Lillo 2009) by a generalization of Dijkstra's algorithm. Unlike its classic counterpart, the multimodal Dijkstra's algorithm (MDA) has a partially ordered data structure to manage path weights. The algorithm takes as input a coloured–edge network G and a source vertex s. It commences at s with the empty path and relaxes each edge that is incident from the source vertex s, adding the single edge paths to the queue. At the front of the queue will be a shortest path estimate to some vertex v adjacent to s. Next, the algorithm determines the shortest paths to v (note that there are more than one) and relaxes edges incidents to v. This iterative process finishes when the queue has no more paths to be compared.

3 Experimental Setup

The experimental study collected vector data information about Denmark, Hungary, Spain, Norway and New Zealand from a GIS library (Geofabrik 2010). These countries were selected based on their similarities in shape and number of locations. For example, New Zealand has resemblances with Norway in shape and number of vertices. Both countries have a long shape and a number of locations between 100 and 200. In addition, one interest of this paper was to establish the extent in which a european multimodal transportation system differs from the New Zealand one.

The multimodal networks were stored and maintained as a set of vertices and bidirectional links. A network dataset for each mode was generated by firstly snapping vertices (towns and cities) to network features according to a tolerance radius. Secondly, a connectivity map was created by an ad-hoc algorithm that iterates itself through vertices. As a spin–off, this algorithm also calculated the real intercity distances as decimal geographic degrees. Additionally, airways were added as a third mode for Norway and New Zealand. Straight distances between airports were used as edge length in this case so that airports had to be snapped to cities to build a connectivity map. Airway data was obtained from (OpenFlights 2010). Characteristics of the resulting networks are shown in Table 1.

Table 1: Characteristics of the Networks

Network	Country	Vertices	Edges	Modes
1	Denmark	124	1284	Road,Rail
2	Hungary	305	7418	Road,Rail
3	Spain	901	5326	Road,Rail
4	Norway	122	641	Road,Rail,Airways
5	New Zealand	183	1436	Road,Rail,Airways

As an illustration, Figure 1 displays the Hungary roadway system which is composed by motorways and primary roads. Likewise, Figure 2 yields a view of New Zealand airway system.

The reported runtimes corresponds to CPU times by computing the total number of shortest path trees ($\mathcal{M}_{uv}$ cardinality) from a source vertex. Each of these trees can be composed by any number of transport modes. Networks were all tested on a standard double core desktop computer of 1.86 GHz and 1.99 GB of RAM.

The calculation of $\mathcal{M}_{uv}$ cardinality was settled upon two different source vertex scenarios. Scenario 1 considered the capital city of each country as source vertex whereas Scenario 2 uses an extreme city as source. For instance, Wellington and Invercargill were picked as source vertices for scenarios 1 and 2 respectively in the New Zealand's case. Besides, the algorithm was responsible by reporting the total number of processing paths (total number paths taken by the iterative subroutine of the MDA) as well as average and maximum $\mathcal{M}_{uv}$ cardinality. Average cardinality is calculated by averaging all vertices' cardinalities whereas the maximum cardinality corresponds to the largest value of $\mathcal{M}_{uv}$ cardinality among vertices.

Figure 1: Hungary roadway system.

4 Results

Results for Scenario 1 are shown in Table 2. CPU times are given in seconds.

One fact that Table 2 evidences is that just a small number of paths become optimal in comparison to the number of processing paths taken by the algorithm. No more than 2% of such paths became optimal for the studied multimodal networks. This is a promising result from a tractability viewpoint.

Spain and New Zealand obtained the largest values of $\mathcal{M}_{uv}$ cardinality. What these countries have in common is a high level of network overlap between road and rail as well as a high number of cities located along such overlaps. These features

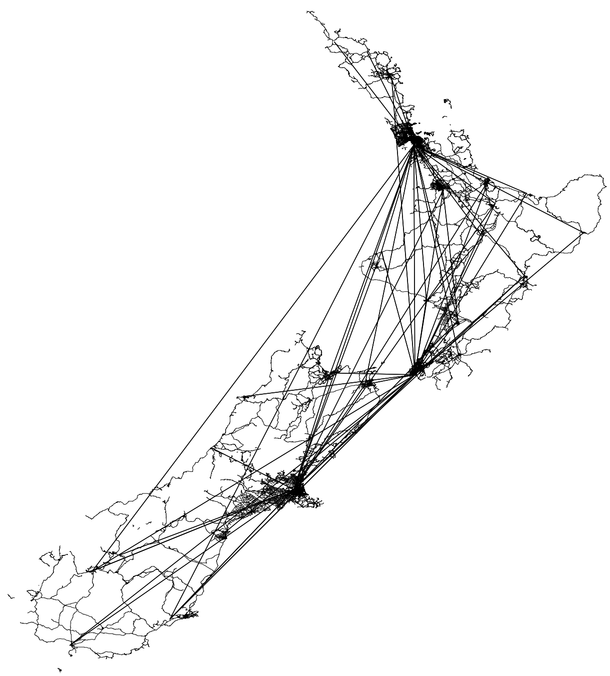


Figure 2: New Zealand airway system.

Table 2: Results for multimodal networks from vector data: Scenario 1

Network	Source city	Average $\mathcal{M}_{uv}$ cardinality	Maximum $\mathcal{M}_{uv}$ cardinality	Processing Paths	CPU time
1	Copenhagen	56	171	10704	0.515
2	Budapest	69	227	53587	6.024
3	Madrid	133	1039	181976	32.216
4	Oslo	41	147	6248	0.158
5	Wellington	759	9342	611230	311.184

together induce a high number of optimal paths because some network sections resemble a coloured–edge chain. Coloured–edge chains are important subgraphs in a general coloured–edge graph because they are able to elicit a worst case scenario. In their work (Ensor and Lillo 2010) show the worst case of a weighted coloured–edge graph occurs when a hamiltonian coloured–edge path (or coloured–edge chain) concentrates most of the lowest weights. In addition, weights in this chain have to satisfy an special equality condition. When these two requirements come together, the total number of optimal paths is $O(k^{n-1})$. In practical terms, those cities (or towns) that require a greater number of intermediate connections to be reached are prone to generate an elevated number of shortest coloured–edge paths. To envision the concept of network overlap, Figure 3 shows road and rail networks for New Zealand.

On the other hand, the rich variety of network links presented in Denmark, Hungary and Norway reduce overlap so that the number of optimal paths tend to be lower. Moreover, maximal cardinalities were found in remote cities (or towns) with no direct link from the sources. For example, Frederiksharn and Rakamaz were the locations reporting the maximum number of optimal paths for Denmark and Hungary, respectively. This indicates the number of optimal paths is affected

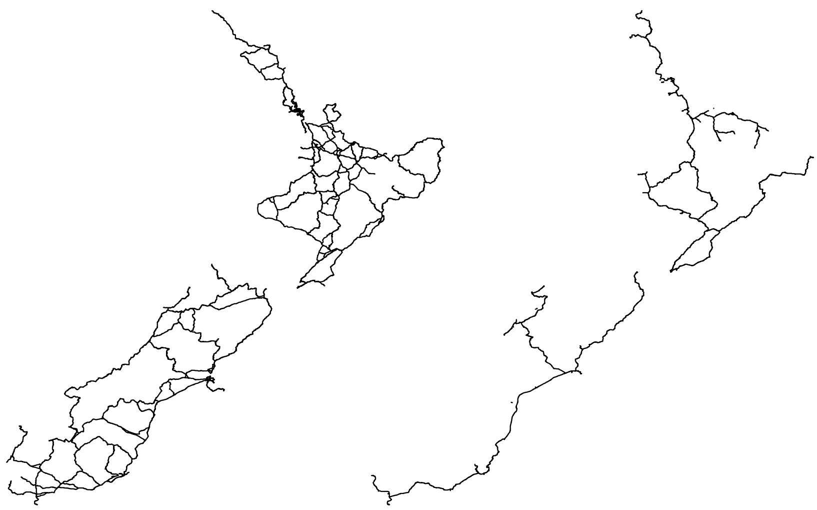


Figure 3: New Zealand road (left) and rail (right) networks. Note the high level of overlap in the east side of the country.

whether the location of the source is changed. A special case is a vertex located at one of the country extremes.

Scenario 2 was set to analyze the impact that source location has on the cardinality of $\mathcal{M}_{uv}$. Table 3 summarizes the corresponding results. The selected location were extreme points situated at one of the four cardinal points. Spain and New Zealand again concentrates the highest numbers of paths. Here, maximum cardinalities were obtained in San Fernando (Spain) and Riverton (New Zealand). Observe that the number of transport modes is not changing the cardinality pattern. Although more modes do increase $\mathcal{M}_{uv}$ cardinality, the overlap is maintained since the configuration of the networks remains unchanged in each mode.

Table 3: Results for multimodal networks from vector data: Scenario 2

Network	Source city	Average $\mathcal{M}_{uv}$ cardinality	Maximum $\mathcal{M}_{uv}$ cardinality	Processing Paths	CPU time
1	Hanstholm	76	332	20009	1.262
2	Csenger	149	600	135218	68.423
3	Tarroella	423	1864	606609	507.657
4	Ergersund	78	245	11674	0.332
5	Kaitaia	5969	33246	1644768	20572.52

5 Conclusions

Real–world multimodal networks were computationally investigated in this work. The experiments were based on digitized road, rail and airways maps from Denmark, Hungary, Spain, Norway and New Zealand. Each map was pre–processed to be used as input by a multimodal version of Dijkstra's algorithm that produces a set of optimal paths ($\mathcal{M}_{uv}$). The cardinality of this set was the main variable to be analyzed because of its influence on the model's tractability. Such cardinality turn out to be higher and significantly concentrated on cities situated far away from the sources in those countries whose transportation systems exhibited a greater level of overlap. Overlap produces that certain sections of a multimodal network resemble a coloured–edge chain. These chains were proved by (Ensor and Lillo 2010) to be the cause of an exponential number of optimal paths in a coloured–edge graph.

Computational times were reasonable considering that the multimodal Dijkstra's algorithm was implemented with a basic data structure (priority queue). Multimodal networks in Scenario 2 required longer runs than Scenario 1 due to the relocation of the sources. This relocation effect indicates higher number of optimal path could be needed to reach distant cities (or towns) when the source vertex is located at the very extreme of a country. Scenario 2 is besides attesting that country shapes are able to alter the number of optimal paths. Longer and slimmer shapes are thus closer to behave as a coloured–edge chain. Hereby, the special shape of New Zealand is also accounting for the elevated number of optimal path found for its extreme locations. Although Norway has a shape resemblance with New Zealand, the lower number of optimal paths is explaining by different circumstances: (1) Norway's rail system is not able to connect the entire country. Rail roughly covers just 20% of the territory. (2) The multimodal transportation system of Norway goes from dense to very sparse as a user moves from the south to the north. Road is predominantly defining the connectivity in the north. (3) The number of airways is much lower than in New Zealand. New Zealand has about 116 different air connections whereas Norway has just 62.

There were not remarkable differences in using two or three transport modes. This suggests that shape and connectivity are more determining factors in the tractability of these networks rather than the value of k.

Future work in this field can tackle the analysis of $\mathcal{M}_{uv}$ cardinality on larger multimodal transportation datasets. This is particularly useful in assessing the correlation between country shape and tractability. Another issue is related to the distribution of weights. For instance, a worst case can be built by taking a coloured–edge chain with pure Euclidean distances as edge weights. However, real distances are far to be pure Euclidean in real transportation systems. Thereby, real distances could be causing a reduction in the number of processing paths.

Improvements on the algorithm are required to speed calculations up. Therefore, faster computational techniques such as parallel computing and ad-doc data structures could result in much faster implementations of the multimodal Dijkstra's algorithm.

Acknowledgments

This research is partly supported by Catholic University of Maule, Talca–Chile, through the project MECESUP–0205.

References

Bontekoning, Y. M., C. Macharis, and J. J. Trip. 2004. "Is a new applied transportation research field emerging?–A review of intermodal rail-truck freight transport literature." *Transportation Research Part A: Policy and Practice* 38 (1): 1–34 (January).

Boyce, D. 2007. "Forecasting Travel on Congested Urban Transportation Networks:Review and Prospects for Network Equilibrium Models." *Networks and Spatial Economics* 7 (2): 99 – 128.

Climaco, Joao, Maria Captivo, and Marta Pascoal. 2010. "On the bicriterion - minimal cost/minimal label - spanning tree problem." *European Journal of Operational Research* 204 (2): 199 – 205.

Ensor, Andrew, and Felipe Lillo. 2009. "Partial order approach to compute shortest paths in multimodal networks." *23rd European Conference on Operational Research.*

———. 2010, July. "Tight Upper Bound on the Number of Optimal Paths in Weighted Coloured-Edge Graphs." *24th European Conference on Operations Research.*

Geofabrik. 2010, http://www.geofabrik.de/. Europe shapefiles.

Heath, Lenwood, and Allan Sioson. 2007. "Multimodal Networks: Structure and Operations." *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 99 (1): 1–19.

Lawler, E. 2001. *Combinatorial Optimization, Networks and Matroids.* New York: Dover Publication, INC.

Lee, D. B. 1994. "Retrospective on large-scale urban models." *Journal of the American Planning Association* 60:35 – 40.

Medeiros, D. J., M. Traband, A. Tribble, R. Lepro, K. Fast, and D. Williams. 2000. "Simulation based design for a shipyard manufacturing process." *Simulation Conference Proceedings.* 1411 – 1414.

Nagurney, A. B. 1984. "Comparative Test of Multimodal Traffic Equilibrium Methods." *Transportation Research B* 18 (6): 469 – 485.

Nigay, Laurence, and Joëlle Coutaz. 1993. "A design space for multimodal systems: concurrent processing and data fusion." *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems.* New York, NY, USA: ACM, 172–178.

OpenFlights. 2010, http://openflights.org/data.html. Airport, airline and route data.

Wegener, M. 1994. "Operational Urban Models: State of the Art." *Journal of the American Planning Association* 6:17 – 30.

Xu, Haiyan, Kevin W. Li, D. Marc Kilgour, and Keith W. Hipel. 2009. "A matrix-based approach to searching colored paths in a weighted colored multidigraph." *Applied Mathematics and Computation* 215 (1): 353 – 366.

Network Design Model with Evacuation Constraints

Yuval Hadas
Department of Civil and Environmental Engineering
University of Auckland, New Zealand
and
Department of management, Faculty of Social Sciences
Bar-Ilan University, Israel
y.hadas@auckland.ac.nz, yuval.hadas@biu.ac.il

Amir Laor
Department of management, Faculty of Social Sciences
Bar-Ilan University, Israel
amir1607@bezeqint.net

Abstract

In recent years terrorism activities have been increasing in scope worldwide as well as the global warming process has had a direct impact on the weather in various climates, subjecting countries around the world to unusually severe storms. Thus for the policy maker it is not only a matter of network design (roads and facilities) in terms of costs and level of service for commuting, but also the matter of network design in terms of costs and evacuation time. This paper focuses on the later – the development of a model for the design of an optimal network in terms of minimizing both evacuation time and network constructions costs. However, the optimal model's complexity does not allow solution within a reasonable timeframe. Therefore, a fast heuristic model was developed based on the minimum-cost algorithm, using the unimodular properties of the model for obtaining integral results. The heuristic algorithm was compared to the optimal algorithm (both based on ILOG CPLEX) on various network scenarios and produced on average 10% higher construction costs than the optimal algorithm. On the other hand, the execution time of the heuristic algorithm was significantly faster than the optimal algorithm.

Key words: Evacuation, Network Design, Facility Location, Optimization, Heuristics.

1 Introduction

In recent years terrorism activities have been increasing in scope worldwide, with the events of 9/11 constituting a significant turning point in the global perception of security needs. The idea that terrorists have gained global reach enabling them to strike anywhere in the world, at any given time, has raised the level of alertness and readiness. Concurrently, the global warming process has had a direct impact on the weather in various climates, subjecting countries around the world to unusually severe storms. Hurricane Katrina, which had nearly wiped out the city of New Orleans, the cyclone storm in Myanmar, and the earth quake in Haiti has been painful reminders of humanity's frailty in the face of the forces of nature.

Such recent events have taught us that in order to reduce damages and minimize casualties we must prepare for the worst. It is our responsibility to design our facilities and infrastructure accordingly, both in terms of access routes and in terms of location, in order to be prepared to evacuate them quickly at short notice.

Thus, for the policy maker it is not only a matter of network design in terms of costs and level of service for commuting, but also the matter of network design in terms of costs and evacuation time, as it is imperative to minimize casualties with quick and efficient evacuation, as well as to utilize the infrastructure for logistic support (Sheu, 2007).

Evacuation planning can be related to the network design problem, the facility location problem, and network flow models. The network design problem (Magnanti and Wong, 1984) is a set of problems designed to construct networks with different objective functions in mind, given the flow can be served by a network constrained by capacity. Facility location problems (Nagy and Salhi, 2007), aims at locating a set of facilities, both serving and being served, in a network, in order to achieve an objective function with a set of constraints (Avella and Boccia, 2009). Models that combine facility location and vehicle routing problems (Bozkaya et al., 2010, Yi and Özdamar, 2007) were also developed, as clearly integrating two interrelated models together can increase efficiency and reduce costs of distribution systems.

Most evacuation planning models are dealing with predetermined networks, such as the model by Sherali et al. (1991), a location-allocation model that minimizes evacuation time but disregards costs, or a model by Xie et al. (2010) that increases network capacity for evacuation by lane reversal and optimizing crossing. Other models were developed to locate relief facilities in a known network (Balcik and Beamon, 2008), but none were found to suggest the structure of the network. Network flow models, such as the maximum-flow and minimum-cost problems (Hillier and Lieberman, 2005) are well known problems that find the total flow from origin to destination (the former), or the minimal cost for flow from origin to destination, given costs associated with arcs and nodes (the later). These models assume costs per unit, rather than construction costs associated with network design problems and facility location problems.

This paper focuses on the development of a multi-objective model for the design of an optimal network in terms of minimizing both evacuation time and network constructions costs. Multi-objective models (Coello Coello et al., 2002) are dealing with models that optimize problems with more than one objective function, for example, modeling delivery system with three objective functions: cost, time, and satisfaction (Tzeng et al., 2007), or optimizing both efficiency and equity measures for school-bush routing (Bowerman et al., 1995).

The paper is organized as follows: a multi-objective evacuation network design problem is introduced: an optimal formulation (section 2), a simplified, single objective model (section 3), and an efficient heuristic based on the minimum-cost problem (section 4). Section 5 provides evaluation of the heuristic algorithm, and section 6 presents conclusions.

2 Network evacuation design problem

Assume a network, as illustrated in Figure 1, with: a) set of origin nodes designed to serve as warehouses, facilities, populated districts, stadium sections, etc., each with estimated construction cost and capacity, b) set of destination nodes designed to serve as evacuation areas (assembly areas, shelters, safe zones, etc., each with estimated construction cost and capacity as well, and c) set of possible transportation infrastructure (roads, aisles, pathways, etc.), each with estimated construction cost and capacity. We are looking for a recommendation for the location of origin nodes, destination nodes, and a network connecting those nodes that will minimize the evacuation time with minimal costs, taking into account the total demand (population, commodities, military mobile-resources, spectators, etc.).

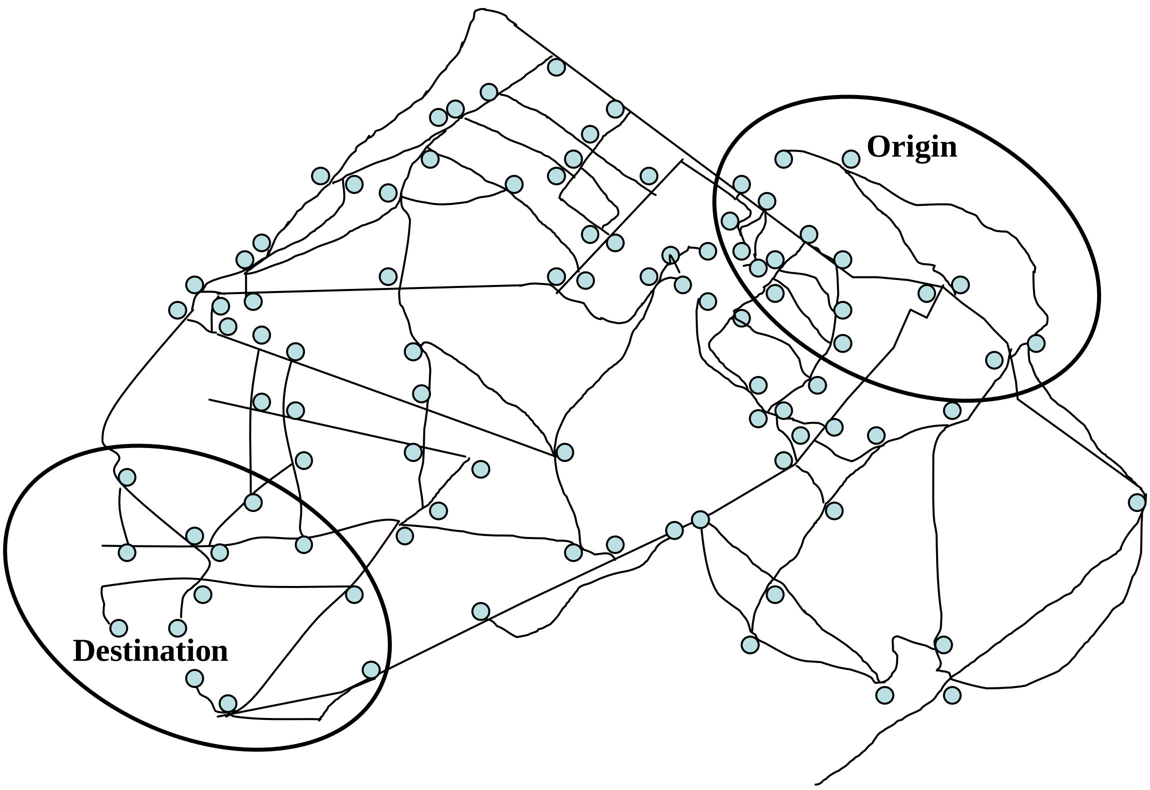


Figure 1 - Illustrative evacuation network

Thus, the Network Evacuation Design Problem (NEDP) can be formulated as follows.

Let $G(N,A)$ be a graph, with N nodes and A arcs, $\{O\} \in N$ origin candidate set (demand), $\{D\} \in N$ destination candidate set (supply). Also let $\{(i,j)\} \in A$ arc candidate set, with $i, j \in [1 \ldots N]$.

Parameters

$Ua_{i,j}$ = capacity of arc (i,j).

Un_i = the capacity of node i.

$Ca_{i,j}$ = construction cost of arc *(i,j)*.

Cn_i = construction cost of node *i*.

TD = total demand

Decision variables

$f_{i,j}$ = flow along arc *(i,j)*

b_i = quantity of demand allocated to node i (positive value – supply, negative value – demand).

T = evacuation time.

Model NEDP1

(1) $$Minimize \sum_{(i.j)\in A} Ca_{i,j} \cdot Ya_{i,j} + \sum_{i\in N} Cn_i \cdot Yn_i$$

(2) $$Minimize\ T$$

Subject To

(3) $$0 \le b_i \le Un_i \cdot Yn_i \quad \forall i \in O$$

(4) $$0 \le -b_i \le Un_i \cdot Yn_i \quad \forall i \in D$$

(5) $$\sum_{i\in O} b_i = TD$$

(6) $$\sum_{i\in D} b_i = -TD$$

(7) $$b_i = 0 \quad \forall i \notin O \cup D$$

(8) $$f_{i,j} \le Ua_{i,j} \cdot Ya_{i,j} \cdot T \quad \forall (i,j) \in A$$

(9) $$\sum_{j=1}^{n} f_{i,j} - \sum_{j=1}^{n} f_{i,j} = b_i \quad \forall i$$

(10) $$f_{i,j} \ge 0,\ f_{i,j} \in \mathbb{Z} \quad \forall (i,j) \in A$$

(11) $$Ya_{i,j} \in \{0,1\} \quad \forall (i,j) \in A$$

(12) $$Yn_i \in \{0,1\} \quad \forall i \in N$$

(13) $$T > 0$$

Objectives (1) and (2) represent the construction costs and evacuation time respectively. Constraints (3) and (4) restrict demand to facility capacity, constraints (5) and (6) enforce that total demand is met, constraint (7) defines transshipment nodes, constraint (8) defines arcs' capacity over time, constraint (9) defines conservation of flow, constraint (10) defines integral flow, constraints (11) and (12) define binary variables, and constraint (13) enforces positive evacuation time.

However, the NEDP1 complexity does not allow solution within a reasonable timeframe due to: a) Multi-objective problem: solution cost and evacuation time, b) The use of integer variables, and c) Integral flow.

Therefore, in order to decrease complexity, a single objective algorithm was constructed, as well as a heuristic Model that is based on the minimum-cost algorithm.

3 Single objective model

Multi-objective problems are usually difficult to solve (Coello Coello, Van Veldhuizen and Lamont, 2002, Current et al., 1990), thus it is sought, if possible, to construct a single objective function. The NEDP1 is unsuitable for that, as it is impossible to convert time into costs. On the other hand, decision variable T can be isolated, as it is relevant only to constraint (8). Thus an iterative algorithm was constructed by removing objective function (2), substituting constraint (13) with

(14) $T = T'$

We define NEDP2 as a single objective variation of the model with T' as an input value. Then, given a predetermined set of possible evacuation times $\{T'\}$ and an optimal costs set $\{C\}$, resulted from executing ENDP2 $|\{T'\}|$ time it is possible to construct the pareto front for NEDP1, as illustrated in Figure 2.

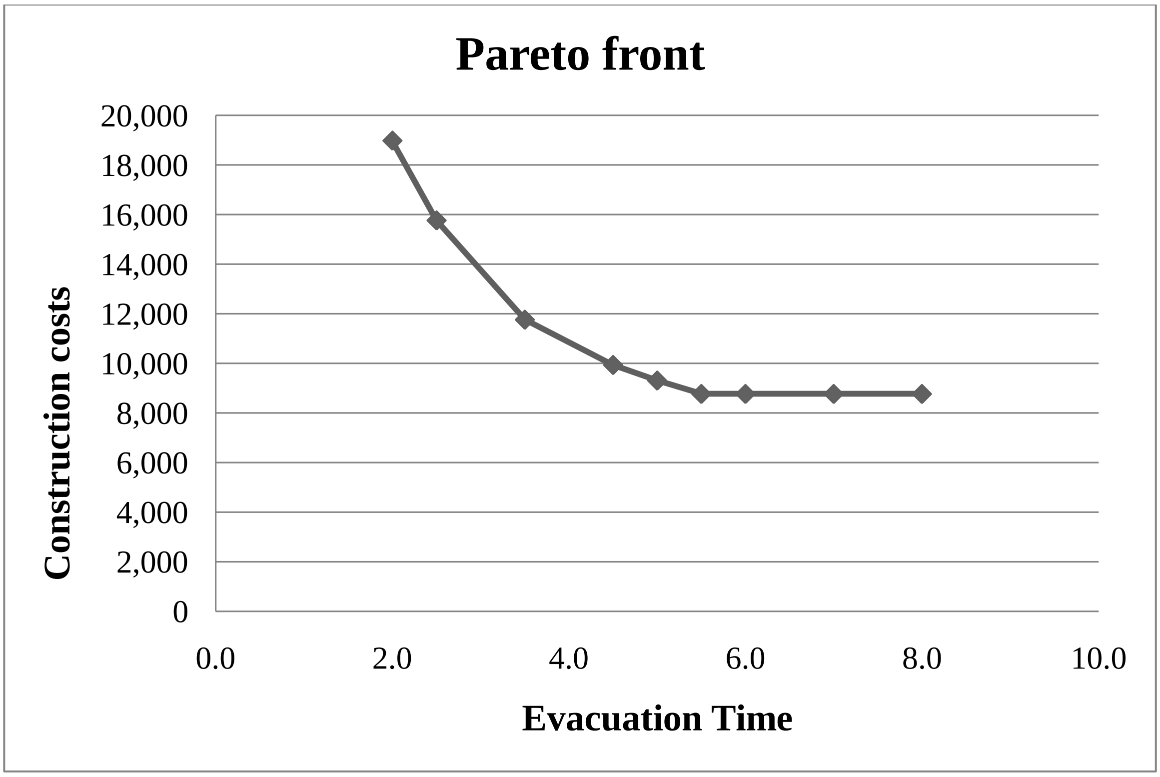


Figure 2 - Pareto front resulted from NEDP2

Even though the NEDP2 model is single objective, the MIP properties retain its NP-complete properties(Schrijver, 1986).

4 Minimum-cost heuristic model

In order to cope with the complexity of the model an efficient, reliable and quickly resolvable heuristic algorithm was developed – NEDP3. The heuristic model is based on the minimum-cost problem which was researched extensively in the past and as a result many efficient algorithms were developed (Ahuja et al., 1992), and the unimodular properties of the minimum-cost problem (Schrijver, 1986, Schrijver, 2003), meaning that if all parameters (*C*, *A*, *b*) of the problem are integral, so is the result vector X.

In order to transform the model to a minimum-cost model, we do the following: a) *Demand and supply nodes transformation:* each possible demand or supply node *i* is transformed to an arc (*i,i''*), with $Ca_{i',i} = Cn_i, Ua_{i',i} = Un_i$. b) *Unit cost transformation:* As the cost $Ca_{i,j}$ is the construction cost of an arc rather than the unit cost $C_{i,j}$ required by the minimum-cost model, an estimated unit cost $C_{i,j}$ was calculated:

$$C_{i,j} = round\left(\frac{Ca_{i,j}}{Ua_{i,j}} \cdot 10^m\right) \tag{15}$$

Where *m* is the precision level, with *m*=2 practically sufficient.

4.1 Minimum-cost formulation

NEDP3 can be formulated as follows:

$$Minimize \sum_{(i.j)\in A} C_{i,j} \cdot f_{i,j} \tag{16}$$

$$b_i \ge 0 \quad \forall i \in O \tag{17}$$

$$b_i \le 0 \quad \forall i \in D \tag{18}$$

$$\sum_{i\in O} b_i = TD \tag{19}$$

$$\sum_{i\in D} b_i = -TD \tag{20}$$

$$b_i = 0 \quad \forall i \notin O \cup D \tag{21}$$

$$f_{i,j} \le U_{i,j} \cdot T \quad \forall (i,j) \in A, i \notin O \wedge j \notin D \tag{22}$$

$$f_{i,j} \le U_{i,j} \quad \forall (i,j) \in A, i \in O \vee j \in D \tag{23}$$

$$\sum_{j=1}^{n} f_{i,j} - \sum_{j=1}^{n} f_{i,j} = b_i \quad \forall i \tag{24}$$

$$f_{i,j} \ge 0 \tag{25}$$

Where constraint (22) restrict the flow along road sections to the capacity over time, while constraint (23) restrict the flow to the facility capacity.

4.2 Heuristic algorithm:

1. *best*=∞, *k*=1
2. Execute NEDP3
3. If *solution*<*best* then *best*=*solution*, retain $\{f\}$

4. In case the flow is less than capacity, it is necessary to preserve $Ca_{i,j} = C_{i,j} \cdot f_{i,j} \quad \forall\, 0 < f_{i,j} < U_{i,j}$, thus the unit cost is updated as follows:

$$(26) \qquad C_{i,j} = \begin{cases} round\left(\dfrac{Ca_{i,j}}{f_{i,j}} \cdot 10^m\right) & \forall\, 0 < 2 \cdot f_{i,j} < U_{i,j} \\ C_{i,j} & Otherwise \end{cases}$$

5. k=k+1
6. If k<(maximum number of iterations) then go to step 2, otherwise terminate.

5 Algorithm evaluation

For the evaluation of NEDP3, in terms of execution time and quality of the results, synthetic networks were generated, with the same structure, as illustrated in Figure 3, and different sizes. Assume a $n \times n$ grid of candidate bi-directional road sections, each assigned construction cost and capacity, $n \times o$ a set of possible origin nodes (nodes to evacuate), and $n \times d$ a set of possible destination nodes (evacuating areas). Both origin and destination nodes are connected to the grid with arcs that holds the construction and capacity of the nodes.

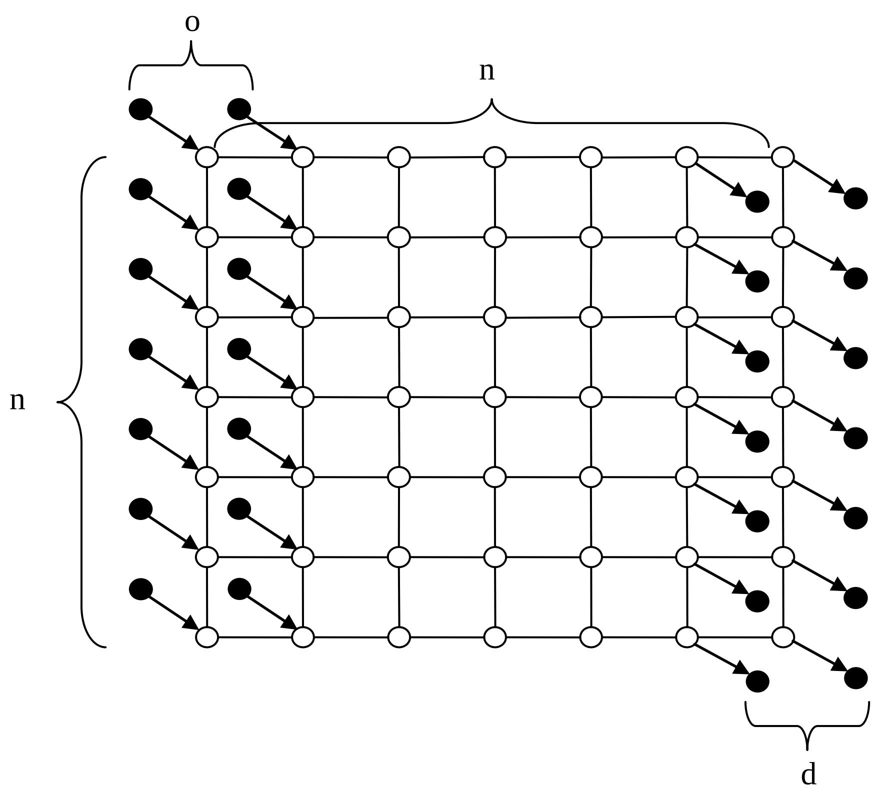


Figure 3 - Synthetic evacuation network

Executions were carried out for a single iteration of NEDP2 and NEDP3 with ILOG CPLEX 12.2 (IBM, 2010), on Intel M540 @ 2.53Ghz based computer with 4GB memory, running windows 7 64bit. The results of NEDP2 (O) versus NEDP3 (H) are summarized in Table 1. For each network the costs and execution times are presented for both models, and the gap between the optimal and heuristic solution.

"best O @ H time" provides the best solution found by NEDP2 at the time NEDP3 was terminated, while "O time @ best H" is the time it takes NEDP2 to reach a solution similar to NEDP3. These indicators are valuable as it further compares the search logic of NEDP2 with NEDP3.

Table 1 - Results of NEDP2 and NEDP3

	Cost		Execution time [hh:mm:ss]		Gap			
Nodes	**O**	**H**	**O**	**H**			**best O @ H time**	**O time @ best H**
60	9,314	10,406	00:00:01	00:00:01	12%			
140	14,598	14,861	00:00:07	00:00:01	2%			
2,700	124,097	128,823	01:56:35	00:00:06	4%	*	125,458	00:00:05
10,400	245,225	249,868	03:03:00	00:00:29	2%	*	581,150	01:20:00
40,800	991,517	988,926	03:09:00	00:02:39	0%	*	47,604,200	
91,200	1,356,894	1,488,099	08:59:00	00:07:59	10%	*	4,283,797	05:03:00
161,600		2,632,201		00:15:50		*		

* - Out of memory for NEDP2, best results presented, if available.

6 Conclusions

1. A new multi-objective model for network design, which is evacuation oriented was developed. This model enables the policy makers to analyze different network designs in terms of construction cost and evacuation time. Such a model is especially relevant for facilities sensitive to mass evacuation (military depot, civilian population close to high-risk areas, stadiums, etc.).
2. As the model is NP-complete, an efficient heuristic was developed, which is both fast, easy to construct, and performs well, when compared to the optimal algorithm.
3. The use of the well known minimum-cost problem as the basis for the heuristic, assures an effective implementation for very large problems, as many efficient variations of the minimum-cost problem exist.
4. Further research of existing networks can be carried out to investigate network improvement for different evacuation scenarios.

References

R.K. Ahuja, Goldberg, A.V., Orlin, J.B., Tarjan, R.E. 1992. Finding minimum-cost flows by double scaling. *Mathematical Programming*. **53**(1) 243-266.

P. Avella, Boccia, M. 2009. A cutting plane algorithm for the capacitated facility location problem. *Computational Optimization and Applications*. **43**(1) 39-65.

B. Balcik, Beamon, B.M. 2008. Facility location in humanitarian relief. *International Journal of Logistics: Research & Applications*. **11**(2) 101-121.

R. Bowerman, Hall, B., Calamai, P. 1995. A multi-objective optimization approach to urban school bus routing: Formulation and solution method. *Transportation Research Part A: Policy and Practice*. **29**(2) 107-123.

B. Bozkaya, Yanik, S., Balcisoy, S. 2010. A GIS-Based Optimization Framework for Competitive Multi-Facility Location-Routing Problem. *Networks and Spatial Economics*. **10**(3) 297-320.

C.A. Coello Coello, Van Veldhuizen, D.A., Lamont, G.B. 2002. *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic, New York.

J. Current, Min, H., Schilling, D. 1990. Multiobjective analysis of facility location decisions. *European Journal of Operational Research*. **49**(3) 295-307.

F.S. Hillier, Lieberman, G.J. 2005. *Introduction to operations research*. McGraw-Hill Higher Education, Boston.

IBM. 2010. *ILOG CPLEX Optimization Studio*. City.

T.L. Magnanti, Wong, R.T. 1984. Network Design and Transportation Planning: Models and Algorithms. *Transportation Science*. **18**(1) 1.

G. Nagy, Salhi, S. 2007. Location-routing: Issues, models and methods. *European Journal of Operational Research*. **177**(2) 649-672.

A. Schrijver. 1986. *Theory of linear and integer programming*. Wiley, Chichester ; New York.

A. Schrijver. 2003. *Combinatorial optimization : polyhedra and efficiency*. Springer, Berlin ; New York.

H.D. Sherali, Carter, T.B., Hobeika, A.G. 1991. A location-allocation model and algorithm for evacuation planning under hurricane/flood conditions. *Transportation Research Part B: Methodological*. **25**(6) 439-452.

J.-B. Sheu. 2007. An emergency logistics distribution approach for quick response to urgent relief demand in disasters. *Transportation Research Part E: Logistics and Transportation Review*. **43**(6) 687-709.

G.-H. Tzeng, Cheng, H.-J., Huang, T.D. 2007. Multi-objective optimal planning for designing relief delivery systems. *Transportation Research Part E: Logistics and Transportation Review*. **43**(6) 673-686.

C. Xie, Lin, D.-Y., Travis Waller, S. 2010. A dynamic evacuation network optimization problem with lane reversal and crossing elimination strategies. *Transportation Research Part E: Logistics and Transportation Review*. **46**(3) 295-316.

W. Yi, Özdamar, L. 2007. A dynamic logistics coordination model for evacuation and support in disaster response activities. *European Journal of Operational Research*. **179**(3) 1177-1193.

Biobjective Air Traffic Flow Management

Kenneth D. Kuhn
Civil and Natural Resources Engineering
University of Canterbury, Christchurch, New Zealand
kenneth.kuhn@canterbury.ac.nz

Andrea Raith
Engineering Science
University of Auckland, Auckland, New Zealand
a.raith@auckland.ac.nz

Abstract

Air traffic flow management seeks to extenuate delay created by congestion in the air traffic control system while ensuring equitable access to air transportation system resources. Mathematical programming formulations of the air traffic flow management problem typically minimize delay costs, ignoring evidence that equity is a critical concern in practice. Recently, authors have adjusted classical formulations, adding terms to the objective function to penalize various results deemed unfair. This work reformulates the air traffic flow management problem as a formal multiobjective optimization problem. We are able to find all Pareto-optimal solutions trading off efficiency and equity, without having to select and parameterize a model of the costs of inequity.

Key words: air traffic flow management, multiobjective optimization, equity, air traffic control.

1 Introduction

There often arise situations where the numbers of aircraft that are scheduled to use certain airports or fly through certain sections of airspace over five to fifteen minute periods of time cannot be safely and efficiently accommodated. The numbers of aircraft that can safely land and take off at airports are largely determined by visibility, wind, and weather conditions that can change relatively rapidly. It is worth noting that in the United States air carriers often set schedules based on assumed optimal weather conditions. In certain busy sections of airspace, local demand for air traffic control services can threaten to outstrip the safe and efficient operating capacity of the system. As past researchers have noted, this happens in western Europe "on an almost routine basis" (Lulli and Odoni 2007). Air traffic flow management (ATFM) involves strategically altering flight schedules to avoid and mitigate local system capacity deficits.

Ground delay programs (GDPs) are commonly used tools of air traffic flow management. During GDPs, flights destined for capacity constrained airports are delayed on the ground at their origin airports prior to take off. The costs, per unit time, of delaying a flight on the ground are significantly lower than the costs of delaying that same flight once airborne (Bertsimas and Patterson 1998). Airspace-flow programs (AFPs) control the rate at which aircraft arrive in capacity constrained sections of airspace. Setting appropriate GDPs and AFPs to avoid multiple potential local air traffic control demand-capacity imbalances across a network is clearly a challenging task. ATFM decision making today follows heuristic procedures, but past researchers have noted that "computer-based decision support systems might improve ATFM performance significantly" (Lulli and Odoni 2007).

There have been significant research efforts formulating and solving various mathematical programming formulations of ATFM problems. Much of the available research focuses on managing air traffic arriving at a single airport. Considering capacity constraints associated with multiple airports and sections of airspace, while allowing for en-route speed adjustments complicates the problem. The most commonly cited formulation of "the air traffic flow management problem" (Bertsimas and Patterson 1998) considers such a problem. Stochastic formulations have addressed the issue of uncertainty in airspace and airport capacities. Researchers have extended formulations to capture the dependence between the arrival and departure capacities of airports, as well as to consider the possibility of rerouting aircraft. Many researchers have investigated the computational performance of various formulations and solution strategies for ATFM problems (Bertsimas and Patterson 1998; Hoffman and Ball 2000).

The "fundamental principle" of air traffic flow management in the United States today is *Ration by Schedule (RBS)* (Gupta and Bertsimas 2010). According to RBS, aircraft are assigned slots at a capacity constrained airport according to a schedule that preserves the order with which the aircraft were originally scheduled to land. The Collaborative Decision-Making (CDM) paradigm, also in widespread use today, holds that ATFM decisions are made with significant authority and responsibility given to individual air carriers.

Practical air traffic flow management has been noted to require "a careful balance between equity and efficiency" (Fearing et al. 2009). Equity in this case refers to ensuring that the costs incurred as a result of ATFM activities do not disproportionally fall on certain airlines or flights. It is worth noting that the twin goals of ATFM are not typically complimentary; there is a "fundamental conflict that may arise between the objectives of efficiency and equity" (Lulli and Odoni 2007). There is a significant gap in the research literature, particularly in the area of network-level air traffic flow management, regarding the lack of consideration of equity as a goal of ATFM activities. It has been suggested that this gap is a key reason past research has not been fully adopted in practice (Gupta and Bertsimas 2010).

There have been recent efforts to bridge the gap between research and practice identified above. One group of researchers has developed a 'fairness metric' to estimate the difference between a given schedule and the first-scheduled, first-served alternative schedule (Fearing et al. 2009). The authors have gone on to incorporate their fairness metric in a mathematical programming formulation of the air traffic flow management problem. Terms penalizing unfair outcomes are weighted and added to the traditional delay cost objective function of air traffic flow management problems. A related research effort proposes an alternate fairness metric, and extends the analysis by allowing airlines to swap landing slots if desired (Gupta and Bertsimas 2010). Again, the work focuses on bringing consideration of equity into a classical air traffic flow management problem formulation. Again a term is added to the delay cost minimizing objective function, weighted to reflect the importance of equity in relation to delay costs.

In this work we reformulate the air traffic flow management problem as a formal biobjective optimization problem. There has not been a formal multiobjective formulation of an air traffic management problem before, to the best of our knowledge, although previous authors have used the language of multiobjective optimization. One paper notes "since there will typically be a trade-off between aggregate system delay and any flight-based fairness criterion, [a new] formulation should essentially consider a bi-criterion approach, enabling the efficient study of the trade-off curve between the two" (Fearing et al. 2009). Another paper notes that in the United States "a primary objective of the [Federal Aviation Administration's Air Traffic Management] functions is provide fair and equitable access" (Vossen et al. 2010).

There are several reasons why a formal bicriteria approach that treats equity and efficiency objectives separately may be preferable to recently introduced formulations based on objective functions that minimize a weighted summation of different objective functions.

Inefficiency and inequity metrics are fundamentally incompatible, and it's not clear what a weighted summation of such terms represents. Decision makers must select and then parameterize a model combining various incompatible terms when using a weighted-summation approach. The selection of the 'optimal' solution will be very sensitive to the weights used when combining the different objectives, yet decision makers will typically have little confidence in a given set of weights (Ehrgott 2005). Arguably the biggest problem associated with weighted-summation approaches is that such approaches are only able to generate a certain class of 'optimal' solutions: those that are found on the boundary of the convex hull of the feasible region of solutions in the multi-dimensional space of the various objective functions (Ehrgott 2005). There exists no intuitive reason for decision makers to restrict themselves to consideration of such solutions. Researchers investigating decision making in complex situations often look for Pareto-optimal policies (Ehrgott 2005). In the context of the air traffic flow management problem, a policy is Pareto optimal if no distinct policy exists that performs better with regards to either equity or efficiency and at least as well with regards to the other objective. The approach introduced in this paper, unlike prior work, is able to identify all Pareto-optimal solutions to the air traffic flow management problem.

2 Mathematical formulations

2.1 Delay cost minimization

The approach introduced in this paper builds off prior work, particularly the formulation of "the air traffic flow management problem" of (Bertsimas and Patterson 1998). That formulation is introduced here, modified somewhat where helpful. In this formulation, a set of flights F is to be scheduled so as to avoid local system capcity deficits. Each flight f in F has a flight plan consisting of N_f ordered elements including an origin airport, sections of airspace sectors, and a destination airport. The flight plans are referenced two separate ways. The unordered set ρ_f includes the N_f elements in flight f's flight plan, while the function P is used to keep track of the trajectory of the aircraft. For any flight f, $P(f,1)$ evaluates to the flight's origin airport, $P(f,N_f)$ the destination airport, and $P(f,n)$ terms (for values of n which are integers between 1 and N_f) the airspace sectors the flight will travel through arranged in the order with which the sections will be flown through.

Capacities are described here by first discretizing time and then noting the numbers of aircraft that can land at and take off from each airport, as well as fly through each airspace sector, in discrete time slices. Note that the formulation is ideal for considering problems like fog reducing airport throughput at San Francisco International Airport during certain (somewhat predictable) hours of the morning. Let $\mathcal{T}$ be the set of all time slices considered, $\mathcal{A}$ the set of all airport considered, and $\mathcal{S}$ the set of all sectors considered. For any airport k in $\mathcal{A}$ and time slice t in $\mathcal{T}$, $D_{k,t}$ and $A_{k,t}$ are defined as the airport departure and arrival capacities. Similarly $S_{k,t}$ is the capacity of sector k ($k \in \mathcal{S}$) during time t ($t \in \mathcal{T}$). Let $T_{f,k}$ be the set of times when flight f may be scheduled to depart from, fly through, or land at k when k is flight f's origin airport, a sector within f's flight plan, or f's destination airport, respecitvely.

The formulation of (Bertsimas and Patterson 1998) is innovative in its definition of decision variables. $x_{f,k,t}$ terms are binary decision variables that are to take on a value of 1 if and only if flight f in F has departed from/flown through/arrived at origin airport/airspace sector/destination airport k *before the end of* time slice t. Certain dummy variables are helpful when setting up the problem. For all flights f in F and for all k in ρ_f, $x_{f,k,t}$ terms are set to 0 for all $t \leq \min T_{f,k} - 1$ and set to 1 for all $t \geq \max T_{f,k}$. Given these definitions, the expression $x_{f,k,t} - x_{f,k,t-1}$ is 1 if and only if flight f has departed from / flown through / arrived at k during time slice t. Similarly, the expression $x_{f,P(f,n),t} - x_{f,P(f,n+1),t}$ is 1 if and only if flight f is in/at $P(f,n)$ during time slice t. Furthermore, the expression $\sum_{t \in T_{f,k}} t(x_{f,k,t} - x_{f,k,t-1})$ will yield the time slice when flight f has departed from / flown through / arrived at k.

Let c_f^g be the cost of delaying flight f on the ground (before the flight takes off) per discrete unit of time. Let c_f^a be the unit cost of delaying flight f once it is in the air. Assume the scheduled (and earliest possible) departure and arrival times of flight f are given as d_f and a_f. Then the total cost incurred holding aircraft at origin airports is $\sum_{f \in F} c_f^g \Big[\sum_{t \in T_{f,P(f,1)}} t(x_{f,P(f,1),t} - x_{f,P(f,1),t-1}) - d_f \Big]$. It is a bit trickier to determine the airborne delay cost since it is essential not to (re)count ground delays. The total airborne delay cost is $\sum_{f\in F} c_f^a\Big[\sum_{t\in T_{f,P(f,N_f)}} t(x_{f,P(f,N_f),t} - x_{f,P(f,N_f),t-1}) - \sum_{t\in T_{f,P(f,1)}} t(x_{f,P(f,1),t} - x_{f,P(f,1),t-1}) - (a_f - d_f)\Big]$. The objective function of the air traffic flow management problem, as defined in (Bertsimas and Patterson 1998), minimizes the sum of delay costs, as in expression (1) below.

$$\min \sum_{f\in F}\Bigg[c_f^a\Big(\sum_{t\in T_{f,P(f,N_f)}} t[x_{f,P(f,N_f),t} - x_{f,P(f,N_f),t-1}] - a_f\Big) + (c_f^g - c_f^a)\Big(\sum_{t\in T_{f,P(f,1)}} t[x_{f,P(f,1),t} - x_{f,P(f,1),t-1}] - d_f\Big)\Bigg] \quad (1)$$

It is worth noting that the above expression references a number of model parameters and dummy variables. Taking out such references actually yields a simpler, and in some ways more intuitive, objective function. The refined objective function, which to this author's knowledge has not appeared in the research literature to date, is shown as expression (2) below.

$$\min \sum_{f\in F}\Big[(-c_f^a)\sum_{t\in T_{f,P(f,N_f)}} x_{f,P(f,N_f),t} + (c_f^a - c_f^g)\sum_{t\in T_{f,P(f,1)}} x_{f,P(f,1),t}\Big] \quad (2)$$

Given an initial solution, setting one additional $x_{f,P(f,N_f),t}$ decision variable to 1 implies reducing f's flight time one time unit and thus reduces delay costs by c_f^a. Setting one additional $x_{f,P(f,1),t}$ term to 1 implies scheduling flight f to take off from its origin airport one time unit earlier. If the arrival time remains unchanged, the flight incurs one less unit of time ground delay but one more unit of time airborne delay and total costs go up by $(c_f^a - c_f^g)$.

Airport and sector capacity constraints, modified to match this paper's terminology, are presented in expressions (3), (4), and (5) below.

$$\sum_{f:P(f,1)=k} (x_{f,k,t} - x_{f,k,t-1}) \le D_{k,t} \qquad \forall k \in \mathcal{A}, t \in \mathcal{T} \quad (3)$$

$$\sum_{f:P(f,N_f)=k} (x_{f,k,t} - x_{f,k,t-1}) \le A_{k,t} \qquad \forall k \in \mathcal{A}, t \in \mathcal{T} \quad (4)$$

$$\sum_{f:P(f,i)=k, i<N_f} (x_{f,k,t} - x_{f,P(f,i+1),t}) \le S_{k,t} \qquad \forall k \in \mathcal{S}, t \in \mathcal{T} \quad (5)$$

In order for the decision variables to be consistent, connectivity constraints are required. For example, if an $x_{f,k,t-1}$ term is set to 1, then $x_{f,k,t}$ must also be set to 1. This is known as connectivity in time as is captured by expression (6) below.

$$x_{f,k,t} - x_{f,k,t-1} \ge 0 \qquad \forall f \in F, k \in \rho_f, t \in T_{f,k} \quad (6)$$

Similarly, the variables must be consistent in terms of individual aircraft trajectories. Let $\beta_{f,k}$ be the minimum number of time units it takes flight f to pass through k. Expression (7) below captures connectivity between sectors.

$$x_{f,P(f,i),t} - x_{f,P(f,i-1),t-\beta_{f,P(f,i-1)}} \leq 0 \qquad \forall f \in F, 2 \leq i \leq N_f, t \in T_{f,P(f,i)} \quad (7)$$

Indivual aircraft will actually fly multiple flights over the course of a day. Delays propogate as the day goes on. It is important to capture this effect to describe a realistic instance of an air traffic flow management problem. Let $\mathcal{C}$ be the set of all pairs of flights (f_1, f_2) where an individual aircraft flies flight f_2 immediately following flight f_1. χ_{f_2} is the (given) minimum time it takes to turnaround the aircraft prior to flight f_2. Then the so-called airport connectivity constraints can be represented as in expression (8).

$$x_{f_2,P(f_2,1),t} - x_{f_1,P(f_1,N_{f_1}),t-\chi_{f_2}} \leq 0 \qquad \forall (f_1, f_2) \in \mathcal{C}, t \in T_{f_2,P(f_2,1)} \quad (8)$$

The final constraint that our decision variables be binary is respresented by expression (9).

$$x_{f,k,t} \in \{0, 1\} \qquad \forall f \in F, k \in \rho_f, t \in T_{f,k} \quad (9)$$

Objective function (2) together with constraint sets (3) through (9) defines the base air traffic flow management problem, as proposed and studied previously (Bertsimas and Patterson 1998).

2.2 Inequity minimization

The formulation of the air traffic flow management problem proposed above is here modified to consider a second objective of minimizing inequity. *Ration by Schedule* is, at least in the United States, “the industry accepted notion of fairness, endorsed by the primary stakeholders, i.e., the [Federal Aviation Administration] and the airlines” (Fearing et al. 2009). Thus, here different schedules are evaluated in terms of how much they deviate from an RBS ideal.

Let’s begin by focusing on a situation where arrival throughput at airports is the major concern, as is common in the United States. Let $\mathcal{R}$ be the set of all ordered pairs of flights (f_1, f_2) where f_1 and f_2 are destined for the same airport with f_1 initially scheduled to arrive before f_2. r_{f_1,f_2} terms are binary decision variables which are to be set to 1 if and only if f_2 arrives before f_1 in the schedule obtained when solving the air traffic flow management problem (the schedule implied by $x_{f,k,t}$ terms). In other words, r_{f_1,f_2} capture *reversals* in the schedule. Such variables were previously proposed (Gupta and Bertsimas 2010), but here such variables are incorporated into separate objective functions for the first time. One example objective function minimizing inequity is shown in expression (10).

$$\min \sum_{(f_1,f_2)\in\mathcal{R}} r_{f_1,f_2} \quad (10)$$

Note that we are counting the number of reversals in order to measure the deviance from an ideal RBS option.

In order to ensure the new decision variables take on values consistent with their desired interpretation, it is necessary to add a constraint set of the formulation. For any pair of flights (f_1, f_2) in $\mathcal{R}$, if f_2 lands before f_1 then r_{f_1,f_2} must be 1. This yields expression (11), which was previously proposed (Gupta and Bertsimas 2010).

$$x_{f_2,P(f_2,N_{f_2}),t} - x_{f_1,P(f_1,N_{f_1}),t} - r_{f_1,f_2} \leq 0 \qquad \forall (f_1,f_2) \in \mathcal{R}, t \in T_{f_2,P(f_2,N_{f_2})} \quad (11)$$

If there are important constraints on capacity within airspace sectors, it makes some sense to generalize the definition of a reversal. Such a generlization is easily accomplished. Let $\mathcal{R}'$ be the set of triples (f_1, f_2, k) where flights f_1 and f_2 are to fly through k (an airport or airspace sector) with f_1 initially scheduled to arrive before f_2. $r'_{f_1,f_2,k}$ terms generalize the previously introduced r_{f_1,f_2} terms. The objective function minimizing inequity and the constraint keeping decision variables consistent are formulated as in expressions (12) and (13).

$$\min \sum_{(f_1,f_2,k)\in\mathcal{R}'} r'_{f_1,f_2,k} \quad (12)$$

$$x_{f_2,k,t} - x_{f_1,k,t} - r'_{f_1,f_2,k} \leq 0 \qquad \forall (f_1,f_2,k) \in \mathcal{R}', t \in T_{f_2,k} \quad (13)$$

Previous authors have noted that applying RBS concurrently for multiple air transportation system resources can yield inefficient results (Fearing et al. 2009). For instance, consider an example where a sizable portion of the flights passing through one airspace sector are destined for a severly capacity constrained destination airport. Forcing all aircraft to go through the sector in the originally scheduled order ensures the sector throughput is reduced to reflect constraints at the troublesome airport. Using a biobjective approach to the air traffic flow management problem allows decision makers to gain greater insight into the trade-offs between efficiency and fairness (in the RBS sense) for particular problem instances.

The metrics described here (from Gupta and Bertsimas, 2010) are not perfect. They count the number of reversals in a schedule but do not take into account the magnitudes of the delays being assigned. Note that a reversal at a busy airport may result in an aircraft being delayed as little as two or three minutes, while another reversal in an infrequently used section of airspace may result is hours of delay. In addition, no accounting is made of the distribution of reversals / delay across the sets of flights and air carriers being managed. Our motivation for considering equity noted the importance of ensuing costs are spread relatively evenly amongst flights and air carriers. An entropy-based objective function could make sense here.

Further research to develop alternate equity maximizing objective functions for the air traffic flow management problem is warranted. More generally, multi-objective optimization could prove quite useful for considering factors rarely mentioned in the current air traffic flow management literature. In particular, it would be possible to develop metrics specifically focused on noise or pollutant emissions and minimize these without having to reduce everything to monetary costs using questionable models or conversion factors.

3 Solving the biobjective problem

The ϵ-constraint method is arguably the best-known approach for solving multiobjective problems (Ehrgott 2005) and is used here. The approach was originally introduced by (Haimes, Ladson, and Wismer 1971). Here an objective functions minimizing inequity, expression (12) above, is converted to yield a constraint, expression (14) below.

$$\sum_{(f_1,f_2,k)\in\mathcal{R}'} r'_{f_1,f_2,k} \leq \epsilon \tag{14}$$

Objective function (2) with constraints identified by expressions (3) through (9), (13), and (14) constitutes a typical (single-objective) air traffic flow management problem which can be solved in a reasonable amount of time for realistic problem instances. Varing ϵ in expression (14) allows us to find all Pareto-optimal solutions for the biobjective air traffic flow management problem (Ehrgott 2005).

For this particular problem, the inequity measure is a count of the number of reversals. For Pareto-optimal solutions, this metric must take on integer values between 0 and the number of reversals generated by maximizing efficiency and ignoring inequity. Note that each reversal necessarily involves delaying the aircraft originally scheduled to access the shared resource first. Thus, there is some reason to be optimistic that there will not be an unreasonably large number of reversals in an efficiency-maximizing schedule. We start by finding the efficiency-maximizing schedule. We then take the number of reversals in this schedule, subtract one, and set ϵ equal to this value in expression (14). We maximize efficiency alone, with the addition of constraint (14). The result is another Pareto-optimal schedule. We then repeat the process, counting the number of reversals in the latest schedule, subtracting one, setting ϵ equal to this value, and resolving. We stop when the problem becomes infeasible or we reach a situation where a schedule involving no reversals is generated. Along the way, we have found all the Pareto-optimal solutions to the biobjective air traffic flow management problem.

In order to test the defined algorithm, we have run computational studies with randomly generated biobjective air traffic flow management problems. In the generated problems, there were 20 airports, 200 airspace sectors, 168 discrete periods of time, and each flight was assumed to fly through 5 airspace sectors between its origin and destination airport with flight delays of between 0 and 6 time periods considered feasible. The chosen parameter values were taken from (Bertsimas and Patterson 1998), which describes the values as typical for realistic-sized problem instances. 2,000 flight paths were randomly generated, along with changing airport and airspace sector capacity constraints.

Figure 1 gives an example of obtained results, focusing on the trade-off between the efficiency and equity objectives. The x-axis shows the number of reversals in obtained solutions, while the y-axis shows the value of the delay minimization objective function. It is worth noting that the reformulation of the objective function used here, expression (2) introduced above, yields negative delay costs. Obtained values can be interpretted as the difference in delay costs between a given solution and a worst-case / maximum delay solution.

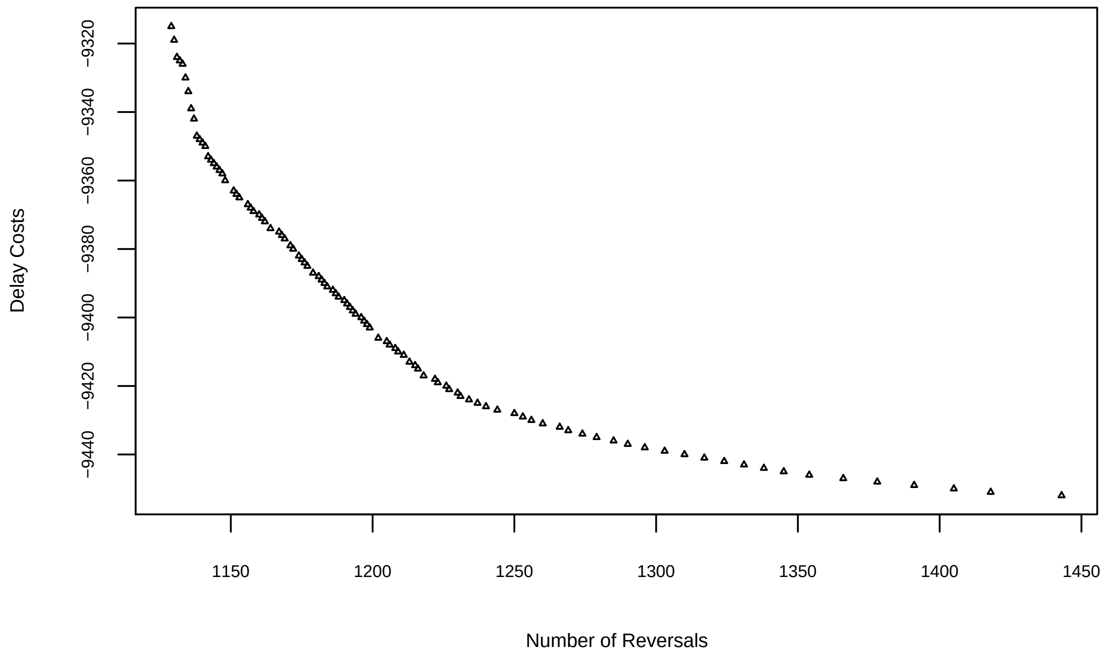


Figure 1: The trade-off between efficiency and equity.

Figure 1 makes clear that there is a choice to be made between efficiency and equity when scheduling flights. The general shape of the relationship shown is convex, indicating that focusing on only one of the objectives may yield very poor performance with regards to the other objective. Although the general shape of the relationship is convex, there are some points that would not be on the boundary of the convex hull of feasible points. In other words, some Pareto-optimal schedules were found that would not have been found using an approach minimizing a weighted summation of delay and inequity costs.

4 Conclusion

The air traffic flow management problem was here extended to a multi-objective optimization problem minimizing inefficiency and inequity. This is an important contribution given that past researchers have identified the failure of past formulations to consider equity concerns as the primary reason prior resarch results have not been adopted in practice in air traffic control. Computational studies show realistic sized biobjective air traffic flow management problems can be solved in reasonable amounts of time, and yield Pareto-optimal solutions not found using distinct approaches based on minimizing a weighted summation of delay and inequity costs. Further work is warranted to define additional objective functions for air traffic flow management problems, and to devise strategies for efficiently solving such problems. In particular, environmental concerns would be worth investigating. It would also be interesting to consider stochastic formulations to address the issue of uncertainty in airspace and airport capacity estimates, or to consider formulations that allow for dynamic rerouting of aircraft.

Acknowledgments

The authors would like to thank Lorenzo Castelli and Charles-Antoine Robelin for drawing the authors' attentions to the air traffic flow management problem.

References

Bertsimas, D., and S. Stock Patterson. 1998. "The air traffic flow management problem with enroute capacities." *Operations Research* 46:406–422.

Ehrgott, M. 2005. *Multicriteria Optimization.* Springer.

Fearing, D., C. Barnhart, D. Bertsimas, and C. Caramanis. 2009. "Equitable and efficient coordination of traffic flow management programs." *International Symposium for Mathematical Programming.* Chicago.

Gupta, S., and D. Bertsimas. 2010. "A proposal for network air traffic flow management incorporating fairness and airline collaboration." *submitted to Operations Research.* available online at http://www.agifors.org/award/AVMedal_submissions.htm.

Haimes, Y., L. Ladson, and D. Wismer. 1971. "On a bicriterion formulation of the problems of integrated system identification and system optimization." *IEEE Transactions on Systems, Man, and Cybernetics* 1:296–297.

Hoffman, R., and M. O. Ball. 2000. "A comparison of formulations for the single-airport ground-holding problem with banking constraints." *Operations Research* 48:578–590.

Lulli, G., and A. Odoni. 2007. "The European air traffic flow management problem." *Transportation Science* 41:431–443.

Vossen, T., M. Ball, R. Hoffman, and M. Wambsganss. 2010. "A general approach to equity in traffic flow management and its application to mitigating exemption bias in ground delay programs." *Air Traffic Control Quarterly* 11:277–292.

Shipping Stem Generation for the Hunter Valley Coal Chain

Natashia Boland and Hamish Waterer
School of Mathematical and Physical Sciences
University of Newcastle, Australia
hamish.waterer@newcastle.edu.au

Martin Savelsbergh
CSIRO Mathematics, Informatics, and Statistics
Australia

Extended Abstract

The Hunter Valley Coal Chain (HVCC) refers to the inland portion of the coal export supply chain in the Hunter Valley, New South Wales, Australia. The HVCC essentially follows the path of the Hunter River traveling south-east from the mining areas in the Hunter Valley to Newcastle. The Port of Newcastle is the world's largest coal export port. In 2008 port throughput was around 92 million tonnes, or more than 10 percent of the world's total trade in coal.

Most of the coal mines in the Hunter Valley are open pit mines. The coal is mined and stored either at a railway siding located at the mine or at a coal loading facility (used by several mines). The coal is then transported to one of the terminals at the Port of Newcastle, almost exclusively by rail. (Some coal is transported to the port by truck.) The coal is dumped and stacked at a terminal to form stockpiles. Coal brands are a blended product, with coal from different mines having different specifications "mixed" in a stockpile to meet the specifications of the customer. Blends and hence stockpiles are most often make-to-order, however the use of dedicated stockpiles for high volume products is increasing. Once the vessel for which the coal is destined arrives at a berth at the terminal, the coal is loaded onto the vessel. The vessel then transports the coal to its destination.

In 2003, the Hunter Valley Coal Chain Logistics Team (HVCCLT) was established to improve the movement of coal from Hunter Valley mines to the port's coal loaders and then to markets across the globe. HVCCLT pools the resources of port operators, railway operators, and railway infrastructure managers into one logistics team. In 2009, when the HVCC went through a major restructuring, the Hunter Valley Coal Chain Coordinator Limited (HVCCC) was incorporated as a new legal entity and formally replaced the HVCCLT (See `http://www.hvccc.com.au/` and Vandervoort [3] for more information on the HVCCC). The HVCCC's mission is to plan and coordinate the cooperative daily operation and long term capacity alignment of the HVCC. Its strategic objectives include, among others:

- To plan and schedule the movement of coal through the HVCC in accordance with the agreed collective needs and contractual obligations of producers and service providers;
- To ensure minimum logistics cost and maximum throughput through the provision of appropriate analysis and advice on capacity constraints (whether physical, operational or commercial) affecting the efficient operation of the HVCC; and
- To advocate positions to other stakeholders and governments on issues relevant to efficient operation, in order to maximize opportunities for improved coordination and/or further expansion of the coal chain.

An important decision support tool employed by HVCCC is a detailed simulation of the HVCC (Welgama and Oyston [4]). The tool is used, among others, for the analysis of the impact of forecast mine production on operations and the analysis of the impact of possible infrastructure expansions on throughput. As the HVCC can be viewed as operating as a pull-system, in which the arrival of a vessel at the port triggers the production, transportation, storage, and ultimately the loading of coal onto the vessel, the forecast mine production needs to be mapped into a stream of vessel arrivals at the port. Such a stream of vessel arrivals is referred to as a shipping stem. Each vessel arrival, referred to as a trip, is characterized by an arrival time, the terminal at which the vessel is to be loaded, a cargo-profile, which specifies the various brands of coal and their tonnage that make up the vessel's cargo, the associated brand-recipes, which specify the various coal components, and thus the mines, that make up a brand (or blend) and their tonnage.

Our work is concerned with this conversion of forecast mine production into a shipping stem, which we refer to as *stem generation*. Generating a shipping stem that matches forecast mine production and that resembles an historic shipping stem is challenging in itself, but the process is complicated by the fact that new mines are brought on line, existing mines are (temporarily) shut down, new brands and new brand-recipes are introduced, and new terminals may start their operations. This information, in particular regarding new brands, recipes, and the nature of demand expected for these, is largely unknown by HVCCC at the time the stem needs to be generated. The blend specifications are regarded as highly sensitive commercial information by the producers, and forward planning needs to be done without it.

Shipping stems are currently produced manually, which is extremely time-consuming; creating a single shipping stem can take up to three weeks. We have developed a multi-phase approach for generating shipping stems that relies on quadratic and integer programming and sampling and produces a shipping stem in a matter of hours. It allows for the generation of multiple shipping stems for the same forecast mine production. The generated shipping stems have been validated by the HVCCC and our shipping stem generator is now an integral part of their analysis framework.

Accommodating future growth is one of the most pressing challenges, if not the most pressing challenge, facing the HVCC. Demand for coal is expected to more than double in the next decade. Thus, strategic capacity planning is a core activity for the HVCCC (Singh et al. [2], Boland and Savelsbergh [1]). Using optimization and simulation models as part of this strategic planning effort provides valuable insights that cannot be obtained, or are extremely hard to obtain, otherwise. However, developing and deploying quantitative models in such situations is non-trivial. Deciding the appropriate level of detail to include in the models is crucial, but often more an art than a science. And so is deciding the scenarios that need to be analyzed and understood. As these scenarios may be quite different from the present environment, obtaining or generating appropriate data for these scenarios can be an ordeal. But even the best strategic planning models are of little use, or, even worse, may inadvertently suggest wrong decisions, if they are populated with inaccurate data.

This is exactly the situation that HVCCC finds itself in. Given the anticipated substantial increase in demand for coal over the next decades, the HVCC needs to significantly increase its annual throughput. Increasing the annual throughput can be accomplished in two ways: (1) by expanding the infrastructure, and (2) by improving the operational efficiency. Understanding the impact of infrastructure expansions, changes in operational procedures, and demand characteristics on the achievable annual throughput is essential.

To be able to assess the achievable annual throughput with confidence, the HVCC needs to be modeled at a fairly detailed level, i.e., at a daily operational level. This implies that yearly mine production forecasts need to be converted to daily demands on the HVCC, i.e., a shipping stem. The importance of generating shipping stems that are representative of what the future may bring cannot be overemphasized. Furthermore, generating such shipping stems is not simply a matter of scaling, of "more of the same". The shipping stems have to reflect that the HVCC is changing, e.g., that new terminals will commence operations and that new mines will start producing and shipping coal.

Our work completely focuses on generating shipping stems, i.e., generating input data, and the challenges associated with doing so. Even though we consider a specific setting, namely the HVCC, similar challenges occur in strategic planning efforts in other industries in which blended products are shipped, such as wheat, fertilizer, gas and petroleum, and our approach may provide ideas and insights that are valuable for others facing similar situations.

References

[1] N.L. Boland and M.W.P. Savelsbergh. Optimizing the Hunter Valley Coal Chain. In *Managing Supply Disruptions*. H. Gurnani, A. Mehrotra, and S. Ray (eds.) Springer-Verlag London Ltd. To appear.

[2] G. Singh, D. Sier, A.T. Ernst, P. Welgama, R. Oyston, and T. Giles. "Long Term Capacity Planning for the Hunter Valley Coal Chain: Models and Algorithms", *20th National ASOR Conference*, Gold Coast, Australia, September 27th–30th, 2009.

[3] J. Vandervoort. “Hunter Valley Coal Chain Coordinator”, *CEDA Annual Conference*, Newcastle, Australia, September 9th, 2010, available at `http://www.hvccc.com.au/Communications/MiscellaneousPresentations.aspx`.

[4] P.S. Welgama and R. Oyston. “Study of options to increase the throughput of the Hunter Valley coal chain”, *Proceedings of MODSIM 2003*, Townsville, July 2003, pp. 1841–1846.

Speed-up of Labelling Algorithms for Biobjective Shortest Path Problems

Andrea Raith
Department of Engineering Science
University of Auckland
New Zealand
a.raith@auckland.ac.nz

Abstract

There is a range of algorithms available to solve biobjective shortest path problems. Here, we focus on biobjective labelling algorithms and propose an acceleration technique, which is easily implemented. We compare the performance of the algorithms with and without the proposed improvements on the basis of test instances with three different network structures. The usage of different data structures within label setting algorithms and their effect on run times is also discussed.

Key words: shortest path problem, label correcting algorithm, label setting algorithm, biobjective optimization, multiobjective optimization.

Introduction

We begin with a formal introduction of the biobjective shortest path problem. Let $G = (N, A)$ be a *directed network* with a set of nodes $N = \{1, \dots, n\}$ and a set of arcs $A \subseteq N \times N$. Two positive costs $c_{ij} = (c^1_{ij}, c^2_{ij}) \in \mathbb{N} \times \mathbb{N}$ are associated with each arc $(i, j) \in A$. A *path* p in G from node $i_0 \in N$ to node $i_l \in N$ is a sequence $p = \{(i_0, i_1), (i_1, i_2), \dots, (i_{l-1}, i_l)\}$ of arcs in A. We denote the set of all paths from s to t by $\mathcal{P}_{st}$. The *biobjective shortest path problem* (BSP) with *source node* $s \in N$ and *target node* $t \in N$ can be formulated as:

$$\begin{array}{ll} \min & z(p) = \begin{cases} z_1(p) = \sum_{(i,j)\in p} c^1_{ij} \\ z_2(p) = \sum_{(i,j)\in p} c^2_{ij} \end{cases} \\ \text{s.t.} & p \in \mathcal{P}_{st}. \end{array}$$

The aim is finding all *efficient* paths, i.e. those with the property that none of the two objectives can be improved without worsening the other one. We call path cost vectors of efficient paths *non-dominated.* We consider only solution approaches that generate a *complete* set of solutions, i.e. for all possible non-dominated path cost vectors, there has to be at least one efficient path.

In a recent paper Raith and Ehrgott (2009), we compared different approaches to solve the biobjective shortest path problem. An important class of solution algorithms are biobjective label correcting and label setting algorithms. We found labelling algorithms to be competitive with others such as the two phase method on some problem instances that were investigated.

There is a multitude of papers dedicated to biobjective or multiobjective shortest path problems, in particular about solving them with labelling algorithms. Label setting algorithms for the biobjective problem were discussed by (Hansen 1980; Tung and Chew 1988), whereas (Martins 1984; Tung and Chew 1992; Martins and Santos 2000; Guerriero and Musmanno 2001; Paixão and Santos 2007) tackle the multiobjective problem. Papers on label correcting algorithms include (Henig 1985; Brumbaugh-Smith and Shier 1989; Skriver and Andersen 2000) in the biobjective case and (Corley and Moon 1985; Hartley 1985; Martins and Santos 2000; Guerriero and Musmanno 2001; Sastry, Janakiraman, and Mohideen 2003; Paixão and Santos 2007) in the multiobjective case. For a more detailed discussion of related literature, refer to Raith and Ehrgott (2009) and the references therein.

In the literature, there is no mention of a bounded labelling algorithm such as the one presented in this paper: Here, we aim at further improving the efficiency of biobjective and multiobjective labelling algorithms by exploiting the fact that the cost vector of every enumerated path from source node s to target node t dominates other paths in the network. It may not always be necessary to extend a path to the target node to confirm that it is dominated. All s-t paths enumerated at any time while the algorithm runs, may dominate other paths at t but also at any other node of the network (as arc costs are assumed to be positive). This means that it may be possible to delete paths at an early stage of the algorithm rather than extending them to the target node. In a similar way, bounds derived from supported solutions found in phase 1 of the two phase method can significantly speed up labelling algorithms in phase 2 as demonstrated in Raith and Ehrgott (2009).

We also discuss how different data structures may impact the run times of biobjective label setting algorithms.

The remainder of the paper is organized as follows. Labelling algorithms are introduced in Section 1. We propose bounded biobjective labelling algorithms in Section 2. Different data structure for use in biobjective label setting algorithms are introduced in Section 3. Numerical results are presented in Section 2.2 and 3.2 respectively.

1 Biobjective Labelling Algorithms

All biobjective label correcting as well as label setting algorithms follow the same basic concept. We first introduce a biobjective label correcting algorithm and then highlight the differences to a label setting one.

Initially, the only labelled node is the source node s with its label set $Labels(s) = \{(0,0)\}$. All labels at a particular node i are extended along all outgoing arcs (i,j). Dominated labels are eliminated from those extended labels and the labels already present at the end node j. The remaining labels form the new label set at node j. Whenever the label set of a node changes, the node has to be marked for reconsideration. At reconsideration, the mark of the node is deleted. The algorithm terminates as soon as there are no more nodes marked for reconsideration.

When traversing an outgoing arc from a node with multiple labels, every label has to be extended along this arc and tested for dominance with the labels of the end node of the arc, which is called *merging*. Merging is the most expensive component of a biobjective label correcting algorithm. The label sets are ordered so that the first component is increasing to reduce computational effort of the merge operation.

The algorithm described here follows a *node-selection* strategy as in every iteration a marked node is selected and all its labels are extended. We refer to *biobjective label correcting with node-selection* as C.

Another strategy in label correcting algorithms is called *label-selection*. Here, labels are considered separately, not together with all other labels at the same node. Instead of keeping track of nodes marked for reconsideration, individual labels are marked whenever they change. In every iteration a single (marked) label is selected and extended as described above. Every label that changes is marked, i.e. entered into the list of marked labels. We just mention label correction with label-selection for completeness, but do not implement it here as we are only extending the labelling algorithms from our previous article (Raith and Ehrgott 2009).

A biobjective label setting algorithm always follows the label-selection principle as one has to ensure that the selected label corresponds to an efficient path from s to the node at which the label is situated. We can ensure this by selecting a lexicographically minimal label amongst all tentative labels. This label is then extended and fixed, as it corresponds to an efficient sub-path and is therefore guaranteed to remain non-dominated. We refer to *biobjective label setting* as S.

For more details on label correcting and label setting algorithms as well as pseudocode, the reader is referred to Raith and Ehrgott (2009).

2 Bounded Labelling

We first discuss modifications to the labelling algorithms and then present some numerical results.

2.1 Modification of Labelling Algorithms

C stops when there are no marked nodes any more, whereas S stops once all labels are fixed. In both cases, all efficient paths from s to all other nodes are obtained, including of course the ones to the target node t.

While a labelling algorithm runs, every label at t corresponds to the cost vector of a path from s to t that is currently not dominated. Therefore, this label dominates parts of the objective space as no label that is dominated by it can represent an efficient path. Labels at *any* node of the network that are dominated by a label at t can be deleted as path costs are non-negative so that once a label anywhere in the network is dominated by a label at t it remains dominated. It is therefore not necessary to extend this label until its path reaches t, it can be deleted as soon as dominance is detected.

We propose a bounded labelling algorithm by modifying any of the labelling algorithms as follows: The algorithm runs as described in Section 1 while there is no label at node t. Once there is at least one label at t, one can start checking bounds. For each newly generated label, one checks whether it is dominated by at least one label at target node t. The labels at the target node, $l_1 = (z_1^1, z_2^1), \ldots, l_m =$

(z_1^m, z_2^m), are sorted by increasing first objective value, i.e. $z_1^1 < z_1^2 < \ldots < z_1^m$ and $z_2^1 > z_2^2 > \ldots > z_2^m$. It is checked if the newly generated label $l = (z_1, z_2)$ is dominated:

set *dominated* = FALSE and $i = 1$
while (*dominated* == FALSE) and ($i \leqq m$) and ($z_1^i \leqq z_1$) **do**
    if $z_2^i \leqq z_2$ **then**
        set *dominated* = TRUE
    else
        $i = i + 1$
    end if
end while

Only labels that are not dominated by any of the labels at t are retained, all others are deleted. It is easy to implement this check into any labelling algorithm. The resulting algorithms are called *bounded labelling* algorithms and denoted by bC and bS. Although, we formulate the dominance check for the biobjective problem here, our idea is easily applicable to multiobjective problems.

It should be noted that biobjective labelling algorithms actually yield the shortest path from s to all other nodes, not just to the target node t. With our improvement the algorithm is restricted to finding the shortest paths from s to t. If the aim was obtaining shortest paths from s to a small number of target nodes $t_1, \ldots, t_l$, the bound check could be modified to only deleting a label if it is dominated by at least one label at *each* target node $t_1, \ldots, t_l$. But considering too many target nodes might diminish the effectiveness of the bounds.

In some cases it can be determined that a label cannot be dominated by any of the labels at t without actively checking all of them. Let the labels at the target node, $l_1, \ldots, l_m$, be sorted by increasing first objective value as above. A label $l = (z_1, z_2)$ at any node will clearly not be dominated by any of the labels $l_1, \ldots, l_m$ if $z_1 < z_1^1$ or $z_2 < z_2^m$. Note that checking $z_1 < z_1^1$ is included as part of the while loop of the dominance check. If $z_2 < z_2^m$, it is not necessary to check a label l against all labels $l_1, \ldots, l_m$ when it is clear a priori that l cannot be dominated. We implemented this additional condition, but achieved no further run time improvements, which is why the corresponding results are not reproduced here.

It should be noted that an approach related to the proposed bounding is A* search, whose multiobjective extension was introduced by Stewart and White (1991) and further extended by Mandow and Pérez de la Cruz (2003), Mandow and Pérez de la Cruz (2006). The concept of A* search is based on guiding the search towards the target by selecting a label based on its value plus some (heuristic) prediction of the label's distance to its destination. In the process dominance by existing labels at the target can also be tested. Here, we only compare existing labels to those at target nodes.

2.2 Numerical Experiments

All numerical experiments are conducted using the same problem instances as in Raith and Ehrgott (2009). The networks have three different basic structures. The first network type consists of road networks of the US, namely those of the three states Washington DC (DC), Rhode Island (RI), and New Jersey (NJ). For each state, nine instances with different source and target nodes were generated. We

Table 1: Average run times for biobjective labelling algorithms for road networks (DC, RI, NJ), grid networks and NetMaker networks (NM).

	avg run time (sec)		ratio bC over C			run time (sec)		ratio bS over S		
name	C	bC	avg	min	max	S	bS	avg	min	max
DC	0.26	0.06	0.33	0.00	1.00	0.12	0.03	0.06	0.00	0.15
RI	5.42	2.11	0.34	0.03	0.87	1.80	0.63	0.39	0.04	0.95
NJ	30.46	10.23	0.33	0.02	0.78	19.74	5.95	0.28	0.01	0.83
grid	2.74	3.79	1.13	0.82	1.78	19.47	21.40	1.01	0.77	1.19
NM	368.21	0.00	0.00	0.00	0.00	1205.15	0.00	0.00	0.00	0.00

refer to these instances as DC, RI, and NJ. Networks in the second group have a rectangular grid structure, and instances are denoted by grid. There are 33 different grid networks. The last group consists of random networks with a structure similar to the NetMaker networks described by Skriver and Andersen (2000). We have 60 different NM networks. Instances in this group are denoted by NM. All instances vary in problem size and number of efficient solutions.

In Table 1 the average run time of the different algorithms is listed in columns C, bC, S, and bS. For C and S we use the run times from Raith and Ehrgott (2009). The average, minimal and maximal ratio of run times of the bounded algorithms over the original ones is listed in columns bC *over* C and bS *over* S. If this ratio is equal to 1, run times of both approaches are identical. If the ratio is between 0 and 1, the bounded algorithms have a shorter run time than the original ones. Otherwise, if the ratio is larger than 1, the original algorithms perform better.

All algorithms were implemented in C and compiled with the gcc compiler (version 4.1.1). Numerical tests are performed on a Linux computer with 2.40GHz Intel®$Core^{\text{TM}}$2 Duo processor and 2GB RAM. Run time is measured in seconds with a precision of 0.01 seconds, a run time of 0.00 represents any run time < 0.01. Algorithms were stopped after 3600 seconds. When the run time of the label correcting or label setting algorithms is $\leqq 0.1$, we do not consider the corresponding ratio in the calculation of the average ratio, as the run time itself is too small to enable meaningful comparisons.

In the case of road networks (rows DC, RI, and NJ in Table 1), the bounded algorithms bC and bS both improve the run time significantly when compared with the original algorithms C and S. For the three network types we observe an average ratio of 0.33 for bounded label correcting and 0.28 for bounded label setting. These ratios indicate that the achieved run time improvements of the bounded labelling algorithms are significant.

Grid networks (row 'grid' in Table 1) on the other hand have a network structure that does not permit (average) improvement of run time through bounded label correcting, exhibited by the average ratio $1.13 > 1$. Here, the additional effort of checking for every newly created label whether it is dominated by any of the (often many!) labels at t seems much larger than what is saved by discarding labels occasionally. For the label setting algorithm, we observe similar run times of the bounded algorithm compared to the original one with an average ratio of 1.01.

The most striking results appear for the NetMaker instances (row 'NM' in Table 1). Here, run time is always reduced to $\leqq 0.01$ although the run time of C and S is very high in most cases leading to an average ratio of 0.00 in all four cases. It is interesting to note that even though the NM instances investigated have different sizes, reflected in increasing run times of C and S, as NM instances grow larger.

However, both bC and bS consistently finish in less than 0.01 seconds. This can be explained via the structure of the networks. The n nodes are numbered consecutively. From every node, arcs can only reach a certain number of nodes "forward" and "backward" leading to an elongate network structure with many possible paths from source node $s = 1$ to origin node $t = n$. The instances were constructed to wrap around, so that arcs from nodes with low numbers that reach backwards may connect to a node with very high number, i.e. close to the target node. Note that this may not be exactly the network structure the authors of Skriver and Andersen (2000) had in mind originally. In many instances, only few efficient paths exist which are also very short as they reach "backwards" from the nodes with low numbers to those with high numbers. For any labelling algorithm to finish, however, it is necessary to generate the paths from the source to all other nodes, whereby many long paths are enumerated that can never be efficient once they reach t. The bounds are very effective here, because efficient s-t paths are found quickly and have fairly low costs in both components, so that many labels can be discarded.

3 Data Structure in Biobjective Label Setting Algorithms

An another approach to improve the computational performance of biobjective labelling algorithms is to select the most appropriate data structures. An example of similar work for single objective label setting and correcting algorithms is Cherkassy, Goldberg, and Radzik (1996), where the computational performance based on different data structures was compared.

3.1 Data Structures

We focus here on biobjective label setting algorithms, which require, in each iteration, the selection of tentative labels that are guaranteed to remain non-dominated as briefly discussed at the end of Section 1. Criteria for selection are for example the selection of a tentative label which (Tung and Chew 1988; Paixão and Santos 2007) is lexicographically minimal, has minimal z_1 value (if all $c_{ij} > 0$), has minimal z_1 or minimal z_2 value (if all $c_{ij} > 0$), or has minimal $z_1 + z_2$ value. With each selection strategy it is necessary to extract a minimal label of some kind from the set of tentative labels. Hence, it is worth investigating how to most efficiently achieve this within the label setting algorithm.

We briefly describe four data structures we use below. Figure 1 shows the same set of tentative labels inserted into the different data structure explained below based on lexicographic ordering of z_1 and z_2. The set of labels is $(10, 10)$, $(10, 21)$, $(12, 28)$, $(13, 14)$, $(13, 15)$, $(13, 28)$, $(14, 28)$, $(16, 15)$, $(17, 28)$, $(19, 23)$, $(21, 23)$, $(24, 15)$.

A very simple data structure is to keep the tentative labels in an ordered linked list, which means the first element is always minimal, the second element is the second smallest, etc.

While it is important to have access to the minimal element, all remaining tentative labels do not need to be ordered. Thus a binary heap or Fibonacci heap, which keep the set of tentative labels only partially ordered, may increase the efficiency (over an ordered list) of inserting and extraction of tentative labels (Ahuja, Magnanti, and Orlin 1993). A binary heap is a tree, in which every node has a smaller key (value by which is is sorted) than its two child nodes, hence the root node is the

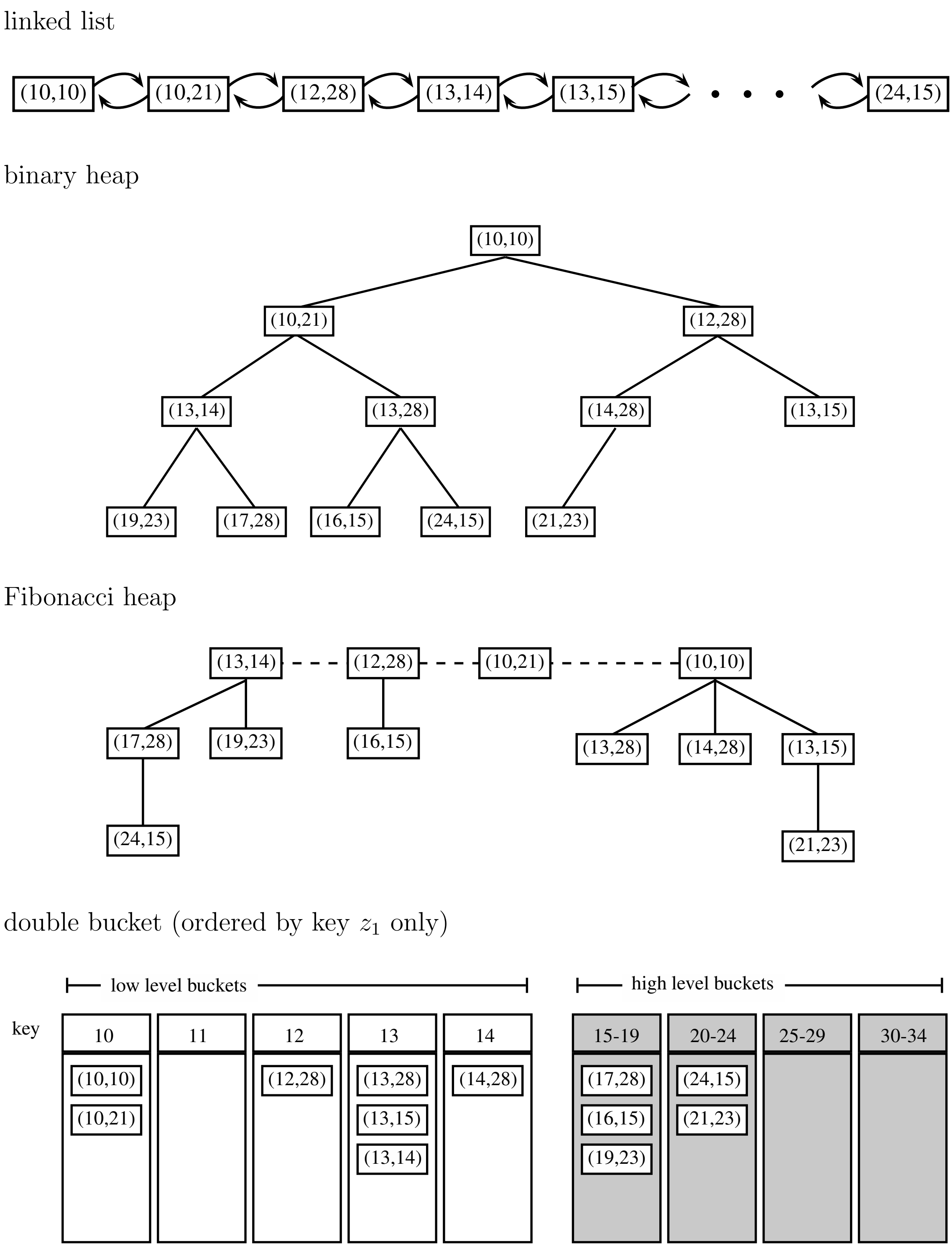


Figure 1: Same set of tentative labels in different data structures.

Table 2: Average run times for biobjective label setting algorithm with different data structures for different network types.

	avg run time (sec)				average ratio binary heap over ...		
name	list	Fibonacci Heap	binary heap	double bucket	list	Fibonacci heap	double bucket
DC	1.67	0.26	0.13	0.12	0.08	0.49	1.18
RI		3.62	2.28	2.08		0.57	1.14
NJ		32.41	22.92	20.72		0.66	1.12
grid	151.82	26.09	22.34	47.4	0.2	0.66	0.79
NM		548.51	511.53	632.86		0.9	0.62

Note: NM results are for 40 smallest NM networks only, which already have high run times.

Numerical tests performed in slightly different computer hence run times differ slightly from those in Table 1.

minimal element but other elements are only partially ordered. A Fibonacci heap contains a forest of trees of different sizes. Again, each parent node has smaller key than its child nodes, but the tree structure is more flexible as every node can have several child nodes. The minimal element is the root node of one of the trees.

It was found by Cherkassy, Goldberg, and Radzik (1996) that double-buckets are an efficient data structure for single objective label setting. A bucket contains labels with a certain key value. A double bucket data structure has two levels of buckets: Low level buckets collect labels that have the same key, whereas high level buckets each collect labels within a range of keys. The buckets for immediate consideration are low level buckets, labels can be selected from the non-empty bucket with lowest key until all low level buckets are empty. Then, elements of the first high level bucket are sorted into the low level buckets for easy access.

Results for single objective shortest path problems are not directly transferable to bi- or multiobjective shortest path problems as there are many more labels in the latter case, due to multiple tentative labels at each node. Hence, we study the effect of using different data structures for the biobjective case.

It should be noted that the first to investigate the effect of data structures on run time of multiobjective shortest path algorithms are Paixão and Santos (2007), compare binary heaps, linked lists, and a single bucket data structure.

3.2 Numerical Experiments

We investigate how the usage of the data structures discussed in Section 3.1 affect the run times for the problem instances discussed in Section 2.2 above. In all our experiments we select lexicographically minimal labels as this appears to be the most common selection criterion in the literature. A double bucket data structure considers all elements that have the same key as being equal, i.e. any minimal element can be selected. This would require many low level buckets for different combinations of the two components of tentative labels. To efficiently use the data structure, we choose to use minimal z_1 as key in this case, which is sufficient as we assume $c_{ij} > 0$.

Table 2 shows average run times of biobjective label setting with each of the data structures used to manage tentative labels. It should be noted that the list data structure mostly had excessive run times, hence results are only reported for DC networks and grid networks.

As the data structure used within the biobjective label setting algorithm in Section 2 is the binary heap, we compare all run times with those of the binary heap implementation. Hence ratios of the binary heap run time over all other run times are computed. Ratios > 1 indicate an approach is, on average, faster than binary heap, and ratios < 1 indicate an approach is slower.

As expected the list data structure consistently performs worst as the computational effort of keeping all labels ordered is large. All other data structures perform worse than the binary heap except for the double bucket data structure in case of the road network instances (DC, NJ, RI). Furthermore, the double bucket data structure clearly outperforms the list data structure but also the Fibonacci heap.

Conclusion and Future Research

The proposed bounded labelling techniques for BSP were shown to be able to improve the run time of biobjective labelling algorithms for two out of three network types. Unfortunately we did not observe any improvements in run time in case of grid networks. Performance of the label correcting algorithms for road networks was improved by 67% on average, that of the label setting algorithm even by an average 72%. The most impressive run time improvements were seen for NetMaker networks where the bounds were able to exploit the network structure very efficiently leading to run time improvements of 100%, i.e. decreasing run time to $\leqq 0.01$ for all instances.

Both for the biobjective and the multiobjective problem, a promising approach might be to compute a few single objective shortest s-t paths initially, such as the lexicographically minimal shortest paths or maybe some weighted sum solutions. These initial solutions can be exploited as bounds from the start of the biobjective labelling algorithm overcoming the problem that bounds can only be used in the bounded labelling algorithms once the first label of an s-t path is generated.

It was also shown how different data structures in biobjective label setting may affect computational performance. The adaptation of speed-up techniques known for single objective shortest path problems to biobjective and multiobjective problems would certainly be worthwhile investigating, for example data structures can be exploited as outlined above. Of course the effect on run time of many more data structure can be studied, also for multiobjective label correcting algorithms.

Acknowledgments

I thank Matthias Ehrgott for his comments and suggestions. Thank you also to Kuan-Min Lin who implemented different data structures in bi-objective label setting algorithms as part of his part IV project in Engineering Science in 2009.

References

Ahuja, R.K., T.L. Magnanti, and J.B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall.

Brumbaugh-Smith, J., and D. Shier. 1989. "An empirical investigation of some bicriterion shortest path algorithms." *European Journal of Operational Research* 43 (2): 216–224.

Cherkassy, B.V., A.V. Goldberg, and T. Radzik. 1996. "Shortest path algorithms: Theory and experimental evaluation." *Mathematical Programming* 73:129–174. Implementations of algorithms obtainable at `http://www.avglab.com/andrew/soft.html`.

Corley, H.W., and I.D. Moon. 1985. "Shortest Paths in Networks with Vector Weights." *Journal of Optimization Theory and Applications* 46 (1): 79–86.

Guerriero, F., and R. Musmanno. 2001. "Label Correcting Methods to Solve Multicriteria Shortest Path Problems." *Journal of Optimization Theory and Applications* 111 (3): 589–613.

Hansen, P. 1980. "Bicriterion Path Problems." Edited by G. Fandel and T. Gal, *Multiple Criteria Decision Making, Theory and Application. Proceedings of the 3rd International Conference, Hagen/Königswinter 1979*, Volume 177 of *Lecture Notes in Economics and Mathematical Systems.* Springer Verlag, Berlin, 109–127.

Hartley, R. 1985. "Vector Optimal Routing by Dynamic Programming." In *Mathematics of Multiobjective Optimization*, edited by P. Serafini, CISM International Centre for Mechanical Sciences – Courses and Lectures, 215–224. Wien: Springer Verlag.

Henig, M. 1985. "The Shortest Path Problem with Two Objective Functions." *European Journal of Operational Research* 25:281–291.

Mandow, L., and J.L. Pérez de la Cruz. 2003. "Multicriteria heuristic search." *European Journal of Operational Research* 150:253–280.

———. 2006. "Comparison of heuristics in multiobjective A* search." *Current Topics in Artificial Intelligence, Selected Papers from the 11th Conference of the Spanish Association for Artificial Intelligence (CAEPIA 2005)*, Lecture Notes in Artificial Intelligence.

Martins, E.Q.V. 1984. "On a multicriteria shortest path problem." *European Journal of Operational Research* 16:236–245.

Martins, E.Q.V, and J.L. Santos. 2000. "The labelling algorithm for the multi-objective shortest path problem." Technical Report, Universidade de Coimbra, Portugal, Departamento de Matématica. `http://www.mat.uc.pt/~eqvm/cientificos/investigacao/mo_papers.html`.

Paixão, J.M., and J.L. Santos. 2007. "Labelling methods for the general case of the multi-objective shortest path problem - a computational study." Technical Report 07–42, Universidade de Coimbra.

Raith, A., and M. Ehrgott. 2009. "A comparison of solution strategies for biobjective shortest path problems." *Computers & Operations Research* 36:1299–1331.

Sastry, V.N., T.N. Janakiraman, and S.I. Mohideen. 2003. "New Algorithms For Multi Objective Shortest Path Problem." *Opsearch* 40 (4): 278–298.

Skriver, A.J.V., and K.A. Andersen. 2000. "A label correcting approach for solving bicriterion shortest-path problems." *Computers & Operations Research* 27:507–524.

Stewart, B.S., and C.C. White. 1991. "Multiobjective A*." *Journal of the Association for Computing Machinery* 38:775 – 814.

Tung, C.T., and K.L. Chew. 1988. "A Bicriterion Pareto-optimal path algorithm." *Asia-Pacific Journal of Operational Research* 5:166–172.

———. 1992. "A multicriteria Pareto-optimal path algorithm." *European Journal of Operational Research* 62:203–209.

Worst-Case Analysis for the Split Delivery Vehicle Routing Problem with Minimum Delivery Amounts

Yupei Xiong
Mercury Systems Inc.
Princeton, NJ 08540

Damon Gulczynski
Department of Mathematics & Physical Sciences
University of Newcastle
Callaghan, NSW 2308
damon.gulczynski@newcastle.edu.au

Bruce Golden
Robert H. Smith School of Business
University of Maryland,
College Park, MD 20742

Daniel Kleitman
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, MA 02139

Edward Wasil
Kogod School of Business
American University,
Washington DC, 20016

Abstract

In the vehicle routing problem (VRP), a fleet of vehicles must service the demands of customers. A vehicle begins and ends its route at the same depot and the sum of the demands of the customers on a route cannot exceed a vehicle's capacity. A

customer must have all of its demand delivered at one time by a single vehicle. The objective is to minimize the total distance traveled by the fleet.

In the split delivery vehicle routing problem (SDVRP), more than one vehicle is allowed to service a customer, so that a customer's demand can be split among several vehicles on different routes. The objective in the SDVRP is to minimize the total distance traveled by the fleet, while satisfying the demand of each customer.

Archetti, Savelsbergh, and Speranza [1] provide a worst-case analysis of the SDVRP. They show that, by allowing split deliveries, travel distance can be reduced by at most 50%, and this is a tight bound. That is, they showed $\frac{Z(VRP)}{Z(SDVRP)} \leq 2$, where $Z(VRP)$ is the distance of an optimal VRP solution to an instance (no splits allowed), and $Z(SDVRP)$ is the distance of an optimal SDVRP solution to the same instance (splits allowed). Furthermore, there exist instances for which $\frac{Z(VRP)}{Z(SDVRP)}$ is arbitrarily close to 2.

Recently, Gulczynski, Golden and Wasil [2] consider the split delivery vehicle routing problem with minimum delivery amounts (SDVRP-MDA). In the SDVRP-MDA, split deliveries are allowed only if at least a minimum fraction of a customer's demand is delivered by each vehicle visiting the customer. For example, if $p = .2$ is the minimum delivery fraction, and $D_i = 10$ is the demand of customer i, then each vehicle visiting customer i must deliver at least $pD_i = 2$ units. A split delivery of, say, 1 unit from one vehicle and 9 units from another vehicle would not be allowed, since 1 is less than the minimum delivery amount. The objective of the SDVRP-MDA is the same as for the SDVRP: minimize the total distance traveled by the fleet, while satisfying the demand of each customer.

Notice, in the SDVRP-MDA the minimum delivery fraction p must be at least 0 and at most 1. When $p = 0$ the SDVRP-MDA reduces to the SDVRP, and when $p > .5$ the SDVRP-MDA reduces to the VRP. Thus, in this paper, we focus on the minimum delivery fractions p for which $0 < p \leq .5$.

Since the SDVRP-MDA is a generalization of the SDVRP, it is natural to wonder what properties of the SDVRP extend to the SDVRP-MDA. Gulczynski, Golden, and Wasil [2] briefly consider properties of the SDVRP-MDA in their paper. One of their results gives bounds for a worst-case SDVRP-MDA scenario. Let $Z(VRP)$ be the distance of an optimal VRP solution to an instance, and let $Z_p(MDA)$ be the distance of an optimal SDVRP-MDA solution to the same instance with minimum delivery fraction p. Further, let $M(p)$ be the least upper bound of $\frac{Z(VRP)}{Z_p(MDA)}$. They showed that $2 - p \leq M(p) \leq 2$.

Notice that the result of Gulczynski, Golden, and Wasil [2] leaves open the possibility that the bound of $\frac{Z(VRP)}{Z_p(MDA)}$ depends on p. Surprisingly, this turns out to *not* be the case for almost all p. In this paper, we prove that for p, $0 < p < .5$, $M(p) = 2$. That is, the worst-case bound for the SDVRP-MDA is independent of p, and it is the same as that for the SDVRP. When $p = .5$ this result does not hold. In the interesting special case of $p = .5$, we have that $M(p) = 1.5$.

References

[1] C. Archetti, M. Savelsbergh, and M. Speranza, *Worst-case analysis for split delivery vehicle routing problems*, Transportation Science, 40 (2006), pp 226-234.

[2] D. Gulczynski, B. Golden, and E. Wasil, *The split delivery vehicle routing problem with minimum delivery amounts*, Transportation Research Part E, 46 (2010), pp 612-626.

Performance of the Branch and Bound Algorithm on the Multistage Insertion Formulation of the Traveling Salesman Problem

Laleh Haerian Ardekani*, Tiru S. Arthanari*, Matthias Ehrgott†
*Department of Information Systems and Operations Management
†Department of Engineering Science
The University of Auckland
{l.haerian, t.arthanari, m.ehrgott}@auckland.ac.nz

Abstract

The traveling salesman problem (TSP) is probably the most studied difficult combinatorial optimization problem. The classical formulation for the TSP suggested by Dantzig, Fulkerson and Johnson (DFJ). The multistage insertion formulation (MI) is a compact formulation. Polytope given by the LP relaxation of the MI formulation for the TSP is proven to be as tight when projected into the DFJ variable space.

In this study we compare the performance of the branch and bound algorithm on the MI formulation for the TSP to other formulations. We define different branching rules using the MI formulation structure. We solve instances from the TSP library using the branch and bound algorithm on various TSP formulations. The MI formulation is found to perform better than other formulations in terms of solution time and the size of the branch and bound tree.

Key words: The Multistage Insertion Formulation, The Traveling Salesman Problem, Branch and Bound Method.

1 Introduction

The traveling salesman problem (TSP) was one of the first problems that was proven to be $\mathcal{NP}-$complete by Karp (1972) and has attracted a lot of attention from researchers. Given a complete graph $G = (V, E)$, where $V = \{1, .., n\}$ and $E = \{(i, j) \in V \times V | i \neq j\}$, the TSP is about finding the least cost Hamiltonian cycle in the graph.

Dantzig, Fulkerson and Johnson (1954) have suggested a formulation for the TSP (DFJ). Let the decision variables x_{ij}, for all nodes i and j in V be equal to one if the edge between i and j belongs to the solution, and be equal to zero otherwise. Let c_{ij} be the cost of edge $(i, j) \in E$. The DFJ formulation for the TSP is given by constraints (1) to (3).

$$\min \sum c_{ij} x_{ij}$$

subject to:

$$\sum_{j<i} x_{ji} + \sum_{j>i} x_{ij} = \quad 2, \qquad \forall i = 1,..,n, \tag{1}$$

$$\sum_{i,j \in S, i<j} x_{ij} \leq \quad |S|-1, \quad \forall S \subseteq V, 2 \leq |S| \leq n-1, \tag{2}$$

$$x_{ij} \in \quad \{0,1\}, \quad \forall i,j \in V. \tag{3}$$

Constraints (1) guarantee that each node is connected to other nodes by exactly two edges. Constraints (2), also known as the *subtour elimination constraints*, make sure that there are no subtours (non-Hamiltonian cycles) in the solution. There are $O(2^{n-1})$ subtour elimination constraints in the DFJ formulation. The polytope given by the LP relaxation of the DFJ formulation is called *the subtour elimination polytope (SEP)*. The SEP is known to be a tighter polytope compared to the polytopes given by other TSP formulations, in the sense that it closely wraps around the TSP polytope.

The branch and bound method is based on the work by Dantzig, Fulkerson and Johnson (1959) on the TSP. The term *branch and bound* originates from the algorithm suggested by Little et. al. (1963). In a generic branch and bound method feasible solutions of an optimization problem are enumerated in order to find the optimal integer solution. The problem is constantly split into two subproblems by adding two mutually exclusive and exhaustive constraints (branching), and lower bounds are used to construct a proof of optimality without exhaustive search (bounding) (Papadimitriou and Steiglitz 1982). More studies on branch and bound for the TSP can be found in the works by Lawler and Wood (1966), Bellmore and Nemhauser(1968), and Balas and Toth (1985).

The multistage insertion formulation (MI) (Arthanari 1983) for the TSP is a compact formulation. It was shown that the polytope given by the LP relaxation of the MI formulation (P_{MI}) when projected into the variable space of the DFJ formulation, is a subset of the SEP (Arthanari and Usha 2000). In previous studies we compared the performance of the MI formulation for the TSP with other formulations (Haerian Ardekani, Arthanari, and Ehrgott 2010). We solved some problem instances from the TSP library, and some instances designed by Papadimitriou and Steiglitz (1978) for this purpose. The MI formulation is shown to outperform other formulations in terms of solution quality, solution time, and number of iterations, for both STSP and ATSP.

In this paper we compare the performance of the branch and bound method on various TSP formulations. The remainder of this paper is structured as follows. The MI formulation is given in Section 2. In Sections 3 and 4 we give a summary of various relaxations and branching rules used in the branch and bound method for the TSP. In Section 5 we suggest some branching rules based on the MI formulation. We report the computational results on applying branch and bound for various TSP formulations on some TSP Library instances in Section 6. Section 7 includes the conclusions.

2 The Multistage Insertion (MI) Formulation for the STSP

The MI formulation for the STSP is based on constructing STSP tours by sequentially inserting nodes into the initial tour of three nodes 1, 2 and 3. Given a set of nodes $V = \{1, ..., n\}$, nodes from 4 to n are inserted sequentially between the nodes of this tour. Let x_{ijk} be equal to one if node k is inserted between nodes i and j, for $1 \leq i < j \leq k-1$ and $4 \leq k \leq n$, and be zero otherwise. Let c_{ij} be the cost of an edge $(i, j) \in E_n = \{(u, v)|1 \leq u < v \leq n\}$, and Let $C_{ijk} = c_{ik} + c_{jk} - c_{ij}$. The MI formulation (Arthanari 1983) is:

$$\min \sum_{k=4}^{n} \sum_{(1 \leq i < j \leq k-1)} C_{ijk} x_{ijk}$$

subject to:

$$\begin{aligned} \sum_{1 \leq i < j \leq k-1} x_{ijk} = &\quad 1, \quad 4 \leq k \leq n, &(4)\\ \sum_{k=4}^{n} x_{ijk} \leq &\quad 1, \quad 1 \leq i < j \leq 3, &(5)\\ -\sum_{r=1}^{i-1} x_{rij} - \sum_{s=i+1}^{j-1} x_{isj} + \sum_{k=j+1}^{n} x_{ijk} \leq &\quad 0, \quad 1 \leq i < j, 4 \leq j \leq n-1, &(6)\\ x_{ijk} \in &\quad \{0, 1\}, \quad 1 \leq i < j \leq k-1, 4 \leq k \leq n. &(7) \end{aligned}$$

Constraints (4) of the formulation guarantees that each node from 4 to n is inserted in an edge. Constraint (5) ensures that at most one node is inserted in each of the edges of T_3. Constraint (6) makes sure that a node is inserted into an edge of the subtour only if that edge has been generated by previous insertions and is available. The MI formulation has $O(n^3)$ variables and $O(n^2)$ constraints.

3 Relaxations for the TSP

In order for the branch and bound method to perform well, it is important to start with an LP relaxation of the IP problem with a small gap. Different relaxations of the TSP have been considered by researchers to apply in the branch and bound methods. We give some of these results below.

The assignment problem (AP) relaxation of the ATSP is used by Eastman (1958), Little et al. (1963), and Bellmore and Maloneyear(Bellmore and Malone 1971). The AP is given by the objective function of the DFJ formulation, and constraints (8)– (11).

$$\sum_{(i,j) \in A} x_{ij} = \quad 1, \quad \forall i \in V, \qquad (8)$$

$$\sum_{(i,j)\in A} x_{ij} = 1, \quad \forall j \in V, \tag{9}$$

$$x_{ij} \leq 1, \quad \forall i,j \in V, \tag{10}$$

$$x_{ij} \geq 0, \quad \forall i,j \in V. \tag{11}$$

The solution to the AP is either a directed tour or a set of directed subtours. Eastman used the network flow algorithm by Ford and Fulkerson (1956) to solve the AP. The AP can be solved using the Hungarian method in $O(n^3)$ time (Ahuja, Magnanti, and Orlin 1993).

The 2-matching relaxation is used by Bellmore and Malone (1971) for the STSP. The answer to the 2-matching problem is either a tour or a collection of subtours. The 2-matching problem has the same objective function as the DFJ formulation subject to constraints (1), (10), and (11).

The 1-tree relaxation of the STSP is first used by Held and Karp (1971) and Christofides (1970). Let V' be $V - \{1\}$. This relaxation is given by constraints (10), (11), and constraints (12) – (14).

$$\sum_{(i,j)\in S\times(V'-S),j>i} x_{ij} + \sum_{(i,j)\in(V'-S)\times S,j>i} x_{ij} \geq 1, \quad \forall S \subset V', \tag{12}$$

$$\sum_{i\in V}\sum_{j>i} x_{ij} = n, \tag{13}$$

$$\sum_{i\in V} x_{1j} = 2, \tag{14}$$

The n-path problem is about finding the shortest path in a graph that includes n nodes (n-path) starting and ending at some node $v \in V$. The LP relaxation of the $n-$path problem is first used by Houck et al. (1980) for the TSP. The $n-$path problem can be solved using dynamic programming in $O(n^3)$ steps (Balas and Toth 1985).

The LP with cutting planes was first suggested by Gomory (1958) for solving generic integer programming optimization problems. Crowder and Padberg (1980) used this solution method for the STSP. The main feature of their method is finding appropriate inequalities to use as cutting planes in each step (Balas and Toth 1985).

4 Branching Methods

We borrow the notations used by Balas and Toth (1985), for defining the subproblems in a branching tree. Starting with the root problem labeled as problem 1, we use string labels for the subproblems in a way that they show the hierarchy of the problem in the branch and bound tree. Given some problem m in the branching tree, let $\mathcal{E}_m$ indicate the set of edges (i,j) that are excluded from problem m, and let $\mathcal{I}_m$ indicate the set of edges included in the problem. Using variables x_{ij}, the sets $\mathcal{E}_m$ and $\mathcal{I}_m$ for some subproblem m can be defined by the following condition.

$$\begin{cases} (i,j) \in \mathcal{I}_m, & \text{if } x_{ij} = 1 \text{ in problem } m, \\ (i,j) \in \mathcal{E}_m, & \text{if } x_{ij} = 0 \text{ in problem } m. \end{cases} \tag{15}$$

The solution to problem m is given subject to the constraints of the predecessor problem of m and subject to condition (15). Given problem m, let the r^{th} successor of problem m be labeled as problem mr. We use the notations $\mathcal{I}_{mr}$ and $\mathcal{E}_{mr}$ to indicate the set of variables that are included in and excluded from mr problem respectively.

Many different branching rules have been suggested for the DFJ formulation. Little et al. (1963) designed a branching rule that creates two subproblems where some arc (i, j) is included in one branch and excluded from the other branch. This rule does not use the structure in the TSP and can be applied to any IP problem (Balas and Toth 1985). Little et al. do not solve the LP at each node but they calculate a lower bound for the LP using the dual solution. Their algorithm has more branching steps but requires less computation at each step compared to other methods (Lawler and Wood 1966). At each branching step, Little et al. (1963) select a pair (i, j) such that the subproblem corresponding to $x_{ij} = 0$ would give a bound as large as possible (Lawler and Wood 1966). The branching rule can be expressed as follows.

$$\begin{cases} \mathcal{E}_{m1} = \mathcal{E}_m \cup \{(i,j)\}, & \mathcal{I}_{m1} = \mathcal{I}_m, \\ \mathcal{E}_{m2} = \mathcal{E}_m, & \mathcal{I}_{m2} = \mathcal{I}_m \cup \{(i,j)\}. \end{cases} \tag{16}$$

For more branching rules and TSP relaxations we refer the reader to the works by Balas and Toth (1985), and Applegate et al. (2006).

5 Branch and Bound Method for the MI Formulation

Similar to the notations used in the previous section for x_{ij} variables, we use the following notations based on the MI formulation structure to define subproblems in a branch and bound tree. Let the sets $\mathcal{E}_m$ and $\mathcal{I}_m$ correspond to the sets of x_{ijk} variables that are excluded from or included in the solution of some subproblem m respectively. We define $\mathcal{E}_m$ and $\mathcal{I}_m$ by the following condition for the subproblem m.

$$\begin{cases} (i,j,k) \in \mathcal{I}_m, & \text{if} x_{ijk} = 1, \\ (i,j,k) \in \mathcal{E}_m, & \text{if} x_{ijk} = 0. \end{cases} \tag{17}$$

We can use sets $\mathcal{E}_m$ and $\mathcal{I}_m$ to define subproblems $m1$ and $m2$ that partition the solution space of problem m into two regions.

Example 1 Consider this solution to a 13-city MI-relaxation problem, $x_{2,3,4} = 1$, $x_{1,2,5} = 1$, $x_{1,3,6} = 1$, $x_{2,5,7} = 1/2$, $x_{2,3,4} = 1$, $x_{2,3,4} = 1$, $x_{2,3,4} = 1$, $x_{2,3,4} = 1$, $x_{2,3,4} = 1$, $x_{2,3,4} = 1$, $x_{2,3,4} = 1$, $x_{2,3,4} = 1$, $x_{3,6,7} = 1/2$, $x_{3,7,8} = 1/2$, $x_{6,7,8} = 1/2$, $x_{2,4,9} = 1$, $x_{3,6,10} = 1/2$, $x_{6,8,10} = 1/2$, $x_{7,8,11} = 1$, $x_{3,8,12} = 1/2$, $x_{3,10,12} = 1/2$, and $x_{6,10,13} = 1$. Having $x_{3,8,12} = 0.5$, and $x_{3,10,12} = 0.5$, suggests that edge $(3, 12)$ is created by having variables $x_{3,j,12}$ to take values greater than zero. We use the following branching rule that partitions the solution space into two regions $\sum_{i \neq 3, i<j<12} x_{i,j,12}$, and $\sum_{i=3,i<j<12} x_{i,j,12}$. Similarly for $k = 8$ we have $x_{3,7,8} = 0.5$, and $x_{6,7,8} = 0.5$, so we can have $\sum_{j \neq 7} x_{ij8} = 0$, and $\sum_{j=7} x_{ij8} = 0$. And finally based on having $x_{3,6,10} = 0.5$ and $x_{6,8,10} = 0.5$, we can use $\sum_{i \neq 6 \wedge j \neq 6} x_{i,j,10} = 0$, and $\sum_{i=6 \vee j=6} x_{i,j,10} = 0$, to partition the solution space into two regions.

The following branching rules generalize the branching formulations given in Example 1. These rules partition the solution space into at least two regions

MI Branching Rule 1 (MIR$_1$)

Given the solution to some problem m, if for some $\hat{k}$ and some $\hat{i}$ we have $0 < x_{\hat{i}j\hat{k}} < 1$, then the successors of m are partitioned into two groups based on the following rules.

$$\begin{cases} \mathcal{E}_{m1} = \mathcal{E}_m \cup \{(i,j,k)|i = \hat{i}, k = \hat{k}, i < j < k\}, \mathcal{I}_{m1} = \mathcal{I}_m, \\ \mathcal{E}_{m2} = \mathcal{E}_m, I_{m2} = \mathcal{I}_m \cup \{(i,j,k)|i = \hat{i}, k = \hat{k}, i < j < k\}, \end{cases} \tag{18}$$

Similarly if for some $\hat{k}$ and some $\hat{j}$ for some problem m we have $0 < x_{i\hat{j}\hat{k}} < 1$, then the successors of m are partitioned into two groups using the following rules.

$$\begin{cases} \mathcal{E}_{m1} = \mathcal{E}_m \cup \{(i,j,k)|j = \hat{j}, k = \hat{k}, 1 < i < j\}, \mathcal{I}_{m1} = \mathcal{I}_m, \\ \mathcal{E}_{m2} = \mathcal{E}_m, \mathcal{I}_{m2} = \mathcal{I}_m \cup \{(i,j,k)|j = \hat{j}, k = \hat{k}, 1 < i < j\}, \end{cases} \tag{19}$$

MI Branching Rule 2 (MIR$_2$)

Given some problem m, if for some $\hat{k}$ and some $\hat{i}$, we have $0 < x_{\hat{i}_1\hat{j}k}, x_{\hat{i}_2\hat{j}k} < 1$, the successors of m can be defined as follows.

$$\begin{cases} \mathcal{E}_{m1} = \mathcal{E}_m, \mathcal{I}_{m1} = \mathcal{I}_m \cup \{(\hat{i}_1\hat{j}\hat{k})\}. \\ \mathcal{E}_{m2} = \mathcal{E}_m, \mathcal{I}_{m2} = \mathcal{I}_m \cup \{(\hat{i}_2\hat{j}\hat{k})\}. \\ \mathcal{E}_{m3} = \mathcal{E}_m \cup \{(\hat{i}_1\hat{j}\hat{k}), (\hat{i}_2\hat{j}\hat{k})\}, \mathcal{I}_{m3} = \mathcal{I}_m. \end{cases} \tag{20}$$

Similarly, for some $\hat{k}$ and some $\hat{i}$, we have $0 < x_{\hat{i}\hat{j}_1k} < 1$, and $0 < x_{\hat{i}\hat{j}_2k} < 1$, the successors of m can be defined as follows.

$$\begin{cases} \mathcal{E}_{m1} = \mathcal{E}_m, \mathcal{I}_{m1} = \mathcal{I}_m \cup \{(\hat{i}\hat{j}_1\hat{k})\} \\ \mathcal{E}_{m2} = \mathcal{E}_m, \mathcal{I}_{m2} = \mathcal{I}_m \cup \{(\hat{i}\hat{j}_2\hat{k})\} \\ \mathcal{E}_{m3} = \mathcal{E}_m \cup \{(\hat{i}\hat{j}_1\hat{k}), (\hat{i}\hat{j}_2\hat{k})\}, \mathcal{I}_{m3} = \mathcal{I}_m. \end{cases} \tag{21}$$

MI Branching Rule 3 (MIR$_3$)

This branching rule is the same as the generic branching rule for the IP methods. Given an MI relaxation solution for some $0 < x_{\hat{i}\hat{j}\hat{k}} < 1$, we define the following branching rule.

$$\begin{cases} \mathcal{E}_{m1} = \mathcal{E}_m, \mathcal{I}_{m1} = \mathcal{I}_m \cup \{(\hat{i}, \hat{j}, \hat{k})\}, \\ \mathcal{E}_{m2} = \mathcal{E}_m \cup \{(\hat{i}, \hat{j}, \hat{k})\}, \mathcal{I}_{m2} = \mathcal{I}_m, \end{cases} \tag{22}$$

When choosing x_{ijk} variables to branch on, we have the option of choosing variables with k values as close to n or as close to 4 as possible. For example in the solution given in Example 1, we can choose between branching on x_{ijk} variables with $k = 7$ or $k = 12$. We represent the combination of a branching rule MIR$_i$ with branching on variables of either greatest value of k or smallest values of k, using MIR$_{i,1}$, and MIR$_{i,2}$, respectively.

In the next section we give some computational results on these different branching rules for some TSPLIB (Reinelt 1995) instances.

Table 1: CPU Second for Branch and Bound Method with MI Branching Rules

Problem	$\mathrm{MIR}_{1,1}$	$\mathrm{MIR}_{2,1}$	$\mathrm{MIR}_{3,1}$	$\mathrm{MIR}_{1,2}$	$\mathrm{MIR}_{2,2}$	$\mathrm{MIR}_{3,2}$
bayg29	1.13	1.41	1.04	**0.75**	1.97	1.97
bays29	1.47	1.43	2.23	**1.00**	3.23	3.29
dantzig42	5.44	3.27	6.07	**2.34**	3.38	5.22
swiss42	**1.42**	1.83	1.45	1.66	3.48	2.40
att48	**4.70**	8.69	8.46	16.89	24.52	17.18
hk48	2.93	3.64	**2.92**	9.31	12.70	12.53
brazil58	**6.23**	8.34	6.98	6.65	9.09	6.77
st70	408.91	588.33	629.31	98.46	**59.38**	206.07
eil76	187.67	504.81	**139.58**	355.28	408.42	851.87
rd100	**305.99**	441.49	361.57	394.76	652.46	444.35
eil101	**1209.30**	3342.42	3956.10	2254.70	3618.60	3693.50
lin105	**113.78**	148.27	119.61	187.84	538.37	384.00
gr120	$> 10^4$	$> 10^4$	$> 10^4$	-	-	-

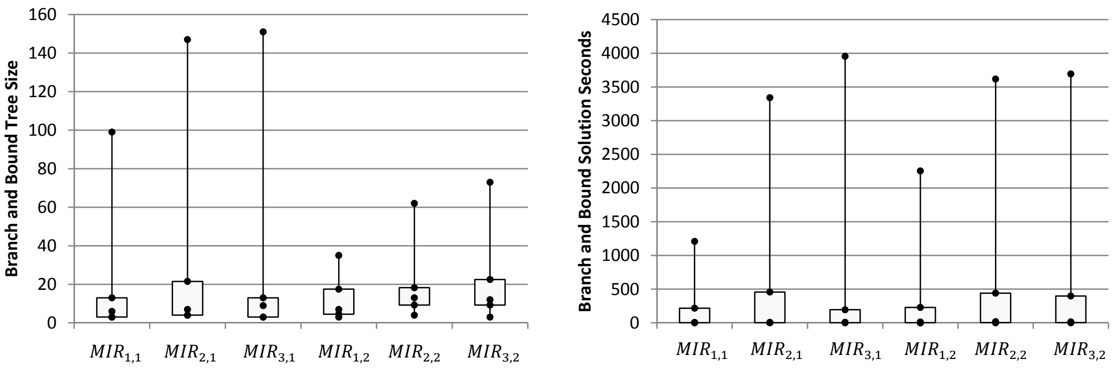


Figure 1: Branch and Bound Solution Seconds (left) and Tree Size (right) for MI Branching Rules

6 Computational Results on Branch and Bound for Various TSP Formulations

We applied the different MI branching rules, $\mathrm{MIR}_{i,j}$ for $i = 1, 2, 3$ and $j = 1, 2$, on some TSPLIB instances. We use Cplex 9.1 for solving the subproblems in each tree node. The solution times are compared in Table 1 and the sizes of the branch and bound trees are compared in Table 2. These results are also illustrated in Figure 1. The median, minimum and maximum values of the solution times and the sizes of the trees are shown in this figure. The first and third quartiles are illustrated using boxes. From Figure 1 we can observe that $\mathrm{MIR}_{1,1}$ and $\mathrm{MIR}_{1,2}$ provide smaller branch and bound trees compared to other methods. The solution times for $\mathrm{MIR}_{1,1}$ are less than those for other rules, except for problem st70, a Euclidean 70-city problem. For the MIR_1 rules, branching on the variables with smallest value of k seems to perform better than using larger values for k.

We applied branch and bound with the DFJ (1954), Wong (1980), Claus (1984), and Carr (1996) formulations and compared with branch and bound on the MI formulation. The results are given in Table 2 and 3, and illustrated in Figure 2. The number of violated subtour elimination constraints for the DFJ formulation that are found and added to the subproblems are shown in Table 3 in column *SEC*. Apart from the DFJ formulation, all the three MI branching rules provide the smallest size

Table 2: Size of the Branch and Bound Tree

Problem	Carr	Claus	DFJ	$MIR_{1,1}$	$MIR_{2,1}$	$MIR_{3,1}$	$MIR_{1,2}$	$MIR_{2,2}$	$MIR_{3,2}$	Wong
bayg29	7	19	5	5	7	5	5	**3**	10	19
bays29	11	33	7	7	7	11	11	**5**	16	33
dantzig42	11	55	5	11	7	13	13	5	7	49
swiss42	5	17	5	**3**	4	**3**	**3**	7	5	27
att48	9	25	7	**5**	10	9	17	25	17	27
hk48	7	7	7	**3**	4	**3**	9	13	13	13
brazil58	3	25	7	**3**	4	**3**	**3**	4	**3**	21
st70	-	-	117	99	147	151	19	**13**	39	-
eil76	-	-	11	19	53	**13**	33	41	73	-
rd100	-	-	7	**7**	11	9	9	13	9	-
eil101	-	-	47	**19**	70	81	35	62	63	-
lin105	-	-	**3**	**3**	4	**3**	5	13	9	-
gr120	-	-	2055	**71**	90	93	-	-	-	-

Table 3: Solution Seconds for Branch and Bound on Different Methods

Problem	Carr	Claus	DFJ	SEC	$MIR_{1,1}$	$MIR_{2,1}$	$MIR_{3,1}$	Wong
bayg29	29.87	136.07	3.97	65	1.15	1.41	**1.04**	558.0
bays29	45.88	218.88	2.26	34	1.47	**1.43**	2.23	934.7
dantzig42	547.43	4706.28	3.37	32	5.44	**3.27**	6.07	13777.5
swiss42	260.33	1418.20	2.85	44	**1.42**	1.83	1.45	4878.5
att48	3097.80	6765.88	9.26	285	**4.70**	8.69	8.46	15277.8
hk48	1964.80	2415.82	4.84	139	2.93	3.64	**2.92**	6764.8
brazil58	5075.50	18168.93	6.53	131	**6.23**	8.34	6.98	23619.5
st70	-	-	**125.21**	90757	408.91	588.33	629.31	-
eil76	-	-	**12.70**	106	187.67	504.81	139.58	-
rd100	-	-	**43.15**	456	305.99	441.49	361.57	-
eil101	-	-	**174.90**	14006	1209.30	3342.42	3956.10	-
lin105	-	-	**46.17**	138	113.78	148.27	119.61	-
gr120	-	-	**5411.50**	4930140200	$> 10^4$	$> 10^4$	$> 10^4$	-

for branching trees and require the least amount of computational time. For the DFJ formulation, the size of the branch and bound tree is greater than MI, except for problem eil76, but the computational time for the DFJ formulation is less than for the MI formulation. This is probably due to the small size of the LPs solved at each branch and bound node. The size of the branch and bound tree is significantly larger for the DFJ formulation for problem gr120 which is a 120-city problem with geographical distances compared to the MI formulation. The MI branching methods provide smaller branch and bound trees for gr120, although their solution time is greater than that of the DFJ formulation.

7 Conclusion

In this paper we suggested three branching rules for the MI formulation of the TSP, we refer to as MIR_1, MIR_2, and MIR_3. We applied these three branching rules in branch and bound algorithms and found rule MIR_1 to perform better than the other two rules. We also compared the performance of branch and bound method on various TSP formulations by solving some TSPLIB instances. We found that the branch and bound method on the MI formulation performed better than other TSP

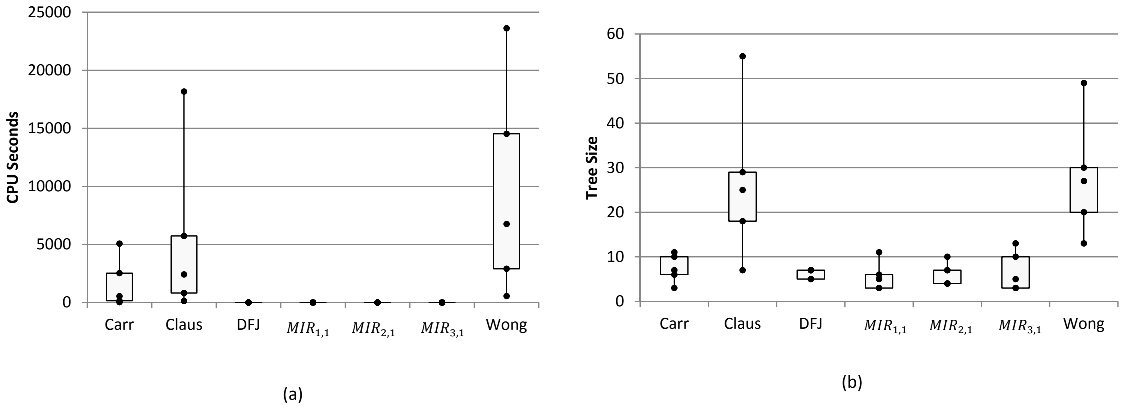


Figure 2: Branch and Bound Solution Seconds (a) and Tree Size (b) for TSP formulations

formulations in terms of solution time and the size of the branch and bound tree. Considering the small solution time for instances smaller than 58, it seems feasible to use the branch and bound method with the branching rule $\text{MIR}_{1,1}$ for solving such TSP instances.

References

Ahuja, R.K., T.L. Magnanti, and J.B. Orlin. 1993. *Network Flows.* Prentice-Hall, New Jersey.

Applegate, D.L., R.E. Bixby, V. Chvtal, and W.J. Cook. 2006. *The Traveling Salesman Problem: A Computational Study.* Princton University Press, New Jersey.

Arthanari, T.S. 1983. "On the traveling salesman problem." Edited by A. Bachem and et. al., *Mathematical Programming - The State of the Art.* Springer-Verlag.

Arthanari, T.S., and M. Usha. 2000. "An alternate formulation of the symmetric traveling salesman problem and its properties." *Discrete Applied Mathematics* 98 (3): 173–190.

Balas, E., and P. Toth. 1985. "Branch and Bound Methods." In *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, edited by E.L. Lawler, J.K. Lenstra, A.H.G. Rinooy Kan, and D.B. Shmoys, 361–402. Wiley, New York.

Bellmore, M., and J.C. Malone. 1971. "Pathology of traveling-salesman subtour-elimination algorithms." *Operations Research* 19 (2): 278–307.

Bellmore, M., and G.L. Nemhauser. 1968. "The traveling salesman problem: A survey." *Operations Research* 16 (3): 538–558.

Carr, R. 1996. "Separating over classes of TSP inequalities defined by 0 node-lifting in polynomial time." In *Integer Programming and Combinatorial Optimization*, edited by E. Lawler, J.K. Lenstra, A.H.G. Rinooy Kan, and D.B.S Shmoys, 460–474. Springer Berlin Heidelberg.

Christofides, N. 1970. "The shortest Hamiltonian chain of a graph." *SIAM Journal on Applied Mathematics* 19 (4): 689–696.

Claus, A. 1984. "A new formulation for the travelling salesman problem." *SIAM Journal on Algebraic and Discrete Methods* 5 (1): 21–25.

Crowder, H., and M.W. Padberg. 1980. "Solving large-scale symmetric travelling salesman problems to optimality." *Management Science* 26 (5): 495–509.

Dantzig, G.B., D.R. Fulkerson, and S.M. Johnson. 1954. "Solution of a large-scale traveling-salesman problem." *Operations Research* 2 (4): 393–410.

———. 1959. "On a linear-programming, combinatorial approach to the traveling-salesman problem." *Operations Research* 7 (1): 58–66.

Eastman, W.L. 1958. "Linear Programming with Pattern Constraints." Ph.D. diss., Harvard University, Boston, MA.

Ford, L.R., and D.R. Fulkerson. 1956. "Solving the transportation problem." *Management Science* 3 (1): 24–32.

Gomory, R.E. 1958. "Outline of an algorithm for integer solutions to linear programs." *Bulletin of the American Mathematical Society* 64 (5): 275–278.

Haerian Ardekani, L., T.S. Arthanari, and M. Ehrgott. 2010. "The multistage insertion formulation of the symmetric traveling salesman problem - An empirical study." Department of Information Systems and Operations Management, Business School, The University of Auckland/Department of Engineering Science, Faculty of Engineering, The University of Auckland. Working Paper No. 319, 18pp.

Held, M., and R.M. Karp. 1971. "The traveling-salesman problem and minimum spanning trees: Part II." *Mathematical Programming* 1 (1): 6–25.

Houck Jr., D.J., J.C. Picard, M. Queyranne, and R.R. Vemuganti. 1980. "The traveling salesman problem as a constrained shortest path problem: Theory and computational experience." *Opsearch* 17:93–109.

Karp, R.M. 1972. "Reducibility Among Combinatorial Problems." In *Complexity of Computer Computations*, edited by R.E. Miller and J.W. Thatcher, 85–103. Plenum New York.

Lawler, E.L., and D.E. Wood. 1966. "Branch-and-bound methods: A survey." *Operations Research* 14 (4): 699–719.

Little, J.D.C., K.G. Murty, D.W. Sweeney, and C. Karel. 1963. "An algorithm for the traveling salesman problem." *Operations Research* 11 (6): 972–989.

Papadimitriou, C.H., and K. Steiglitz. 1978. "Some examples of difficult traveling salesman problems." *Operations Research* 26 (3): 434–443.

———. 1982. *Combinatorial Optimization Algorithms and Complexity.* Printce-Hall.

Reinelt, G. 1995. TSPLIB. http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95.

Wong, R.T. 1980. "Integer programming formulations of the travelling salesman problem." *IEEE Conf. on Circuits and Computers.* 149–152.

A Subset-Selection Prize-Collecting TSP with Uncertain Speed

Dr Shane Dye, Dr John Giffin, Dr Nicola Petty
Department of Management
University of Canterbury
New Zealand
{shane.dye,john.giffin,nicola.petty }@canterbury.ac.nz

Abstract

Rogaine is a sport similar to orienteering but with the locations to visit and visit-order selected by each competitor during the event. Locations have an associated reward value and competitors have a time limit to collect the largest accumulated reward and return to the base location. Penalties apply if competitors return late. The competitor's problem can be modelled as a subset-selection prize-collecting TSP assuming deterministic travel times. In practice, travel times are estimated from the map provided at the start of the event and the route taken may be adjusted depending on conditions observed.

To better understand the effect of uncertain travel times, we model the problem as a two-stage stochastic program with uncertain speed. Speed is modelled as a finite number of constant-speed scenarios. In the first stage a single route is selected. In the second stage, the speed is revealed and the route may be shortened by heading directly back to base from any location on the route. The objective is to maximise expected accumulated reward.

We present an IP formulation and explore properties of optimal solutions. A branch and bound algorithm with knapsack based bounds is developed and initial computational results are provided.

Key words: Rogaine, TSP, prize-collecting, subset-selection, stochastic programming.

1 The Model and Rogaining

We examine a stochastic subset-selection prize-collecting TSP (travelling salesperson problem) model. For the deterministic version (DR), consider a set of locations $\{0, \ldots, n\}$ with given distances, d_{ij}, between locations, base location 0 and reward, $r_i > 0$, associated with each non-base location. Given time limit $W > 0$, linear time penalty per unit time, $c > 0$ and speed, $s > 0$, a solution is a route beginning at and returning to the base location, $A = \{a_0{=}0, a_1, \ldots, a_m, 0\}$, $m \leq n$, visiting locations at most once. The objective is to maximise the net reward, that is, the accumulated reward from locations visited less any penalty accrued from returning after the time limit. More formally, the objective is:

$$\max \sum_{k=1}^{m} r_{a_k} - c \max\left\{0, \frac{1}{s}\left(\sum_{k=1}^{m} d_{a_{k-1},a_k} + d_{a_m,0}\right) - W\right\}$$

We assume the distances satisfy the triangle inequality: $d_{ij} \leq d_{ik} + d_{kj}$ for all i, j, k, although this is non-restrictive since we may replace any distance matrix with the corresponding shortest path matrix. The data sign restrictions are used to eliminate uninteresting cases.

The stochastic version (SR) introduces a set of q speed scenarios, each described by a speed and corresponding probability, (s_t, p_t). A solution corresponds to a, first-stage, planned visit-order for all, or a subset, of the locations, $\{a_0=0, a_1, \ldots, a_m\}$, and, in the second stage, for each scenario an abandon-location, $a_{L(t)}$ where $0 \le L(t) \le m$, at which point in the visit-order the route heads directly back to the base location, $\{a_0=0, a_1, \ldots, a_{L(t)}, 0\}$, collecting reward only from those locations visited. The objective is to maximise expected net reward. The stochastic objective function can be written as:

$$\max \sum_{t=1}^{q} p_t \left(\sum_{k=1}^{L(t)} r_{a_k} - c \max \left\{ 0, \frac{1}{s_t} \left(\sum_{k=1}^{L(t)} d_{a_{k-1}, a_k} + d_{a_{L(t)}, 0} \right) - W \right\} \right)$$

We study this model to gain insight into the effects of uncertain travel times in the sport of rogaining and to determine whether it is feasible to develop optimisation methods to study these solutions.

Rogaine is a sport similar to orienteering but with the locations to visit and visit-order selected by each team, of two or more, during the event. Teams are supplied with a map showing the area of the event and locations with reward values. They have a limited time to plan before being allowed to start-off. All members of a team must remain together throughout the event. Locations have an associated reward value and teams have a time limit to collect the largest accumulated reward and return to the base location. Linear time penalties apply if a team returns late.

During the planning phase, teams must make judgements about the travel times between locations. These judgements are used in planning the route. Computers and other electronic devices are not allowed in the planning phase or throughout the event, and our plan is to investigate and evaluate planning strategies. As a first step this model allows us to compare optimal strategies using deterministic and stochastic travel times for a very simple model of uncertainty. The speeds can represent average or relative speeds and the distances could be adjusted to account of terrain.

2 Literature Review and related problems

Gordon (2006) develops heuristics to use when developing routes during rogaining events. The heuristics are compared to the optimal route found through an enumeration algorithm. The model and data sets used are deterministic with predefined paths between control points and a constant speed throughout assumed. Testing with different speeds for up-hill and down-hill legs was reported.

Related problems include TSPs with profits, classified by Feillet *et al* (2005) into three groups based on how two metrics (distance and profit) are combined. The profitable tour problem includes both in the objective function, the orienteering problem maximises profit under a hard distance limit, and the prize-collecting TSP minimises distance under a hard minimum profit limit. These variations occur in the literature under various names.

Model (DR) is a relaxation of the orienteering problem since it penalises violations of the distance constraint. Laporte and Martello (1990) develop an enumeration algorithm for the orienteering problem using a knapsack-based relaxation to provide upper bounds. Ramesh et al (1992) use Lagrangain relaxation and an improvement procedure to develop a branch and bound algorithm for the same problem.

Stochastic versions have also been studied. Tang and Miller-Hooks (2005) study a two-stage profitable tour problem with stochastic travel times, service times and travel costs. A chance constraint is added to enforce a given minimum probability that the travel time is within a given limit. The full tour is determined *a priori*. Andreatta and Lulli (2008) look at a multi-stage TSP with the problem cast as a Markov decision process in which the objective function is to minimise long-term travel costs. Nodes with urgent demands need to be visited in the period the demand occurs with non-urgent demands able to be delayed by a period.

3 IP formulations

When the speed scenarios are ordered by decreasing speed, the abandon-locations, $a_{L(t)}$, are in a (non-strict) reverse order to the first-stage visit-order, *i.e.*, $L(t+1) \le \mathrm{L}(t)$. This property can be used in IP formulations of the model.

Proposition 1: If the sequence of speeds $\{s_1, \ldots, s_q\}$ is decreasing, then there exist optimal abandon-location indices $\{L(1), \ldots, L(q)\}$ which are also decreasing.

Proof: It is sufficient to demonstrate that if $s_1 > s_2$ then $L(2) \le L(1)$ for a fixed visit-order. Define $Z(s,L)$ to be the net reward given speed s and abandon location a_L under the given visit-order. Assume the premise is false. This means that $L(2) > L(1)$, $Z(s_2,L(2)) > Z(s_2,L(1))$, and $Z(s_1,L(1)) > Z(s_1,L(2))$. For the last case, speed s_1, the accrued penalty using abandon-location $L(2)$ must be more than the accumulated reward from the additionally visited locations. In particular, it must be positive. From this it follows that $Z(s_1,L(2)) > Z(s_2,L(2))$ since $s_2 < s_1$ and more penalty will accrue under speed s_2.Then $Z(s_1,L(1)) > Z(s_1,L(2)) > Z(s_2,L(2)) > Z(s_2,L(1))$ and a penalty must accrue with speed s_2 and abandon-location $a_{L(1)}$ since otherwise $Z(s_1,L(1)) = Z(s_2,L(1))$. Put:

$$R_1 = \sum_{k=1}^{L(1)} r_{a_k}, \qquad R_2 = \sum_{k=L(1)+1}^{L(2)} r_{a_k},$$

$$D_1 = \sum_{k=1}^{L(1)} d_{a_{k-1},a_k} + d_{a_{L(1)},0}, \quad D_2 = \sum_{k=L(1)+1}^{L(2)} d_{a_{k-1},a_k} + d_{a_{L(2)},0} - d_{a_{L(1)},0}.$$

It follows that $R_2 > 0$, $D_2 > 0$ and:

$$\begin{aligned} Z(s_1,L(1)) - Z(s_1,L(2)) &= R_1 - c\max\left\{0, \frac{1}{s_1}D_1 - W\right\} - (R_1+R_2) + c\left(\frac{1}{s_1}\left(D_1+D_2\right) - W\right) \\ &= \frac{cD_2}{s_1} - R_2 - c\max\left\{W - \frac{1}{s_1}D_1, 0\right\} \\ &\le \frac{cD_2}{s_1} - R_2 \end{aligned}$$

As a consequence:

$$Z(s_2,L(1)) - Z(s_2,L(2)) = \frac{cD_2}{s_2} - R_2 > \frac{cD_2}{s_1} - R_2 \ge Z(s_1,L(1)) - Z(s_1,L(2)) \ge 0$$

This contradicts $Z(s_2,L(2)) > Z(s_2,L(1))$.□

This property is used extensively throughout this paper. For the remainder of the paper we assume $s_1 > s_2 > \cdots > s_q$.

Model SR can be formulated as an integer program with the first-stage representing the visit-order of locations, and the second-stage determining the order of locations visited and the arcs taken in each speed scenario. The abandon-location ordering property means the first-stage can be subsumed into Scenario 1, the fastest speed scenario.

Decision Variables

x_{ikt} – Indicates whether location $i \in \{0, \ldots, n\}$ is the kth location visited, $k = 0, \ldots, n + 1$, in scenario $t \in \{1, \ldots, q\}$,.

y_{ijt} – Indicates whether the route travels directly from location i to j in scenario $t \in \{1, \ldots, q\}$.

v_t – Time limit violation for scenario $t \in \{1, \ldots, q\}$.

Formulation (SR-1)

$$\max \sum_{t=1}^{q} p_t \left(\sum_{i=1}^{n} \sum_{k=1}^{n} r_i x_{ikt} - c v_t \right) \tag{1}$$

$$\sum_{k=1}^{n} x_{ik1} \le 1 \qquad i = 1, \ldots, n. \tag{2}$$

$$\sum_{i=1}^{n} x_{ik1} \le 1 \qquad k = 1, \ldots, n \tag{3}$$

$$x_{00t} = 1,\ x_{01t} = 0,\ x_{i0t} = 0 \qquad i = 1, \ldots, n,\ t = 1, \ldots, q. \tag{4}$$

$$\sum_{k=2}^{n+1} x_{0kt} = 1 \qquad t = 1, \ldots, q. \tag{5}$$

$$\sum_{i=0}^{n} x_{ikt} \le \sum_{i=1}^{n} x_{i,k-1,t} \qquad k = 2, \ldots, n + 1,\ t = 1, \ldots, q. \tag{6}$$

$$x_{ik,t+1} \le x_{ikt} \qquad i \in \{1, \ldots, n\},\ k = 1, \ldots, n,\ t = 1, \ldots, q - 1. \tag{7}$$

$$y_{ijt} \ge x_{i,k-1,t} + x_{jkt} - 1 \qquad i \ne j \in \{0, \ldots, n\},\ k = 1, \ldots, n + 1,\ t = 1, \ldots, q. \tag{8}$$

$$\sum_{\substack{j=0 \\ j \ne i}}^{n} (y_{ijt} + y_{jit}) \le 2 \qquad i \in \{0, \ldots, n\},\ t = 1, \ldots, q. \tag{9}$$

$$\sum_{i=0}^{n} \sum_{\substack{j=0 \\ j \ne i}}^{n} d_{ij} y_{ijt} - s_t v_t \le s_t W \qquad t = 1, \ldots, q. \tag{10}$$

$$x_{ikt} \in \{0,1\}, y_{ijt} \in \{0,1\}, v_t \ge 0 \qquad i \ne j \in \{0, \ldots, n\},\ k = 0, \ldots, n + 1,\ t = 1, \ldots, q.$$

The objective (1) calculates the expected net reward over all speed scenarios. Constraints (2)–(6) ensure loops are correctly defined, in sequential order starting from the base location, 0, with trivial loops excluded. Constraint (7) ensures speed ordering of scenarios according to Proposition 1. Constraint (8) forces arcs to be correctly indicated between consecutive locations and Constraint (10) determines the time limit penalty. The formulation allows irrelevant arcs to be indicated if they do not violate the time limit when this is slack. Constraint (9) avoids these irrelevant arcs being connected to the optimal loop—strictly it is not necessary.

Formulation SR-1 can be strengthened by adding additional scenario ordering constraints based on the abandon-location ordering property.

$$y_{ij,t+1} \le y_{ijt} \qquad i \ne j \in \{1, \ldots, n\},\ t = 1, \ldots, q - 1.$$

An alternative model can be formulated using only the arc and node indicator variables and subtour elimination constraints. Due to the implicit ordering of routes by speed, the subtour elimination constraints are only needed for the Scenario 1.

Decision Variables

u_{0j} – Indicates whether all routes travel directly from location 0 to j as the first arc, $j \in \{1, \ldots, n\}$.

y_{ijt} – Indicates whether the route travels directly from location i to j, $0 \le i < j \le n$, in scenario $t \in \{1, \ldots, q\}$.

z_{it} – Indicates whether location $i \in \{1, \ldots, n\}$ is visited in scenario $t \in \{1, \ldots, q\}$.

v_t – Time limit violation for scenario $t \in \{1, \ldots, q\}$.

Formulation (SR-2)

$$\max \sum_{t=1}^{q} p_t \left(\sum_{i=1}^{n} r_i z_{it} - c v_t \right) \tag{11}$$

$$\sum_{j=1}^{n} u_{0j} = 1 \tag{12}$$

$$\sum_{i=1}^{n} y_{0it} = 1 \qquad t = 1, \ldots, q. \tag{13}$$

$$u_{0k} + \sum_{i=0}^{k-1} y_{ikt} + \sum_{i=k+1}^{n} y_{ikt} = 2 z_{kt} \qquad k = 1, \ldots, n,\ t = 1, \ldots, q. \tag{14}$$

$$\sum_{j=1}^{n} d_{0j} u_{0j} + \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} d_{ij} y_{ijt} - s_t v_t \le s_t W \qquad t = 1, \ldots, q. \tag{15}$$

$$y_{ij,t+1} \le y_{ijt} \qquad 1 \le i < j \le n,\ t = 1, \ldots, q-1. \tag{16}$$

$$z_{i,t+1} \le z_{it} \qquad i = 1, \ldots, n,\ t = 1, \ldots, q-1. \tag{17}$$

$$2 \sum_{k \in S} z_{k1} \le |S| \left(\sum_{\substack{i \in S, j \notin S \\ i < j}} y_{ij1} + \sum_{\substack{i \notin S, j \in S \\ i < j}} y_{ij1} + \sum_{j \in S} u_{0j} \right) \qquad S \subset \{1, \ldots, n\},\ |S| \ge 3. \tag{18}$$

$$u_{0j} \in \{0,1\}, y_{ijt} \in \{0,1\}, z_{jt} \in \{0,1\}, v_t \ge 0 \qquad 0 \le i < j \le n,\ t = 1, \ldots, q.$$

The objective (11) calculates the expected net reward over all speed scenarios. Constraints (12) and (13) force exactly one leaving and returning arc from base location 0. Constraint (14) ensures the degree of location nodes is two on the loop and zero otherwise. Constraint (15) determines the time limit penalty. Constraints (16) and (17) ensure speed ordering of scenarios according to Proposition 1, for both locations and arcs not connecting to the base location, 0. Constraint (18) eliminates subtours in the fastest speed scenario (scenario 1). The scenario ordering constraints ensure no subtours for the other scenarios.

Optimising SR-2 could be done by relaxing the subtour elimination constraints, and then adding back violated constraints during a branch-and-cut type algorithm.

4 Enumerative algorithm

Notice that if the first-stage visit order is set, the second-stage optimal solutions can be found by a linear-time search for each scenario. This suggests a branch-and-bound or enumerative scheme branching on the visit-order rather than the arcs to use in each scenario. We follow a scheme based on that proposed by Laporte and Martello (1990) for a deterministic version of the problem. The algorithm framework consists of incrementally extending a simple path from location 0, representing a partial visit-order, using branch-and-bound. Each branch-and-bound node, h, records a partial visit-order $A(h) = \{a_0{=}0, a_1, \ldots, a_{m(h)}\}$. From $A(h)$ local upper and lower bounds are determined using solutions; global bounds are also kept. These bounds are used, as usual, to fathom nodes, maintain an incumbent solution and indicate termination of the algorithm. Descendent nodes are generated by branching over all possible choices for $a_{m(h)+1}$.

4.1 Properties and Bounds

Various upper and lower bounds are examined, many based on a partial or full visit-order A. In the following Z^{SR} refers to the optimal solution value of an instance SR. The lower bound fixes the visit-order in SR. It is tight if the visit-order is optimal.

Proposition 2: Given any instance of SR, and a visit-order $A = \{a_0{=}0, a_1, \ldots, a_m\}$, define

$$F(t) = \max_{0 \le \ell \le m} \sum_{k=1}^{\ell} r_{a_k} - \frac{c}{s_t} \max\left\{0, \sum_{k=1}^{\ell} d_{a_{k-1}a_k} + d_{a_\ell 0} - s_t W\right\} \tag{19}$$

Then $Z^{\mathrm{F}} = \sum_{t=1}^{q} p_t F(t) \le Z^{\mathrm{SR}}$. Furthermore, the bound is tight if A is the optimal visit-order for SR.

The following upper bounds follow knapsack bounds derived in Laporte and Martello (1990) for a deterministic version of the problem. They are based upon generalised knapsack problems with vertex weights

$$w_i = \alpha \min_{i \ne j}\{d_{ji}\} + (1-\alpha) \min_{i \ne j}\{d_{ij}\} \qquad i = 0, \ldots, n. \tag{20}$$

for some real value α ($0 \le \alpha \le 1$).

Proposition 3: Given any instance of SR, real value α ($0 \le \alpha \le 1$), and vertex weights given by (20), let Z^{KL} be the optimal solution value of the following problem, KL, consisting of q linked 0-1 knapsack problems with linear over-capacity penalties:

$$\max \sum_{t=1}^{q} p_t \left(\sum_{i=1}^{n} r_i z_{it} - c v_t \right) \tag{21}$$

$$\sum_{i=1}^{n} w_i z_{it} - s_t v_t \le s_t W - w_0 \qquad t = 1, \ldots, q \tag{22}$$

$$z_{i,t+1} \le z_{it} \qquad i = 1, \ldots, n,\ t = 1, \ldots, q-1. \tag{23}$$

$$z_{it} \in \{0,1\}, v_t \ge 0 \qquad i = 1, \ldots, n,\ t = 1, \ldots, q.$$

Then, $Z^{\mathrm{KL}} \ge Z^{\mathrm{SR}}$. Furthermore, let $Z^{\mathrm{K}}(t)$ be the optimal solution value of individual 0-1 knapsack problem (with linear over-capacity penalty):

$$\max \sum_{i=1}^{n} r_i z_{it} - c v_t \tag{24}$$

$$\sum_{i=1}^{n} w_i z_{it} - s_t v_t \le s_t W - w_0 \tag{25}$$

$$z_{it} \in \{0,1\}, v_t \ge 0 \qquad i = 1, \ldots, n.$$

Then:

$$\sum_{t=1}^{q} p_t Z^{\mathrm{K}}(t) \geq Z^{\mathrm{SR}}$$

Proof: It is sufficient to prove the first part of the theorem as the second part is a consequence of relaxing constraints (23). Let $A^* = \{a_0=0, a_1, \ldots, a_m\}$ be the optimal visit-order, $L(t)$, $t = 1, \ldots, q$, the optimal abandon-location indices, and v_t^* the optimal time-limit violations for SR. We show there is a feasible solution to KL with the same objective function value. Define $z_{a_i t} = 1$ for $i = 1, \ldots, L(t)$, $t = 1, \ldots, q$, and zero otherwise. From (10), for $t = 1, \ldots, q$, we must have

$$\sum_{k=1}^{L(t)} d_{a_{k-1}a_k} - s_t v_t^* \leq s_t W,$$

so,

$$(1-\alpha) d_{0a_1} + \sum_{k=1}^{L(t)-1} \left(\alpha d_{a_{k-1}a_k} + (1-\alpha) d_{a_k a_{k+1}} \right) + \alpha d_{a_{L(t)}0} - s_t v_t^* \leq s_t W.$$

From (22) $\alpha d_{a_{k-1}a_k} + (1-\alpha) d_{a_k a_{k+1}} \geq w_{a_k}$ and the conclusion follows.□

Proposition 4: Given any instance of SR with an initial fixed partial vertex-order $A(h) = \{a_0=0, a_1, \ldots, a_{m(h)}\}$, real value α ($0 \leq \alpha \leq 1$), and vertex weights given by (20), let $F(t)$ be given by (19) using $A(h)$ as the visit-order and $Z^{\mathrm{KF}}(A(h), t)$ be the optimal solution value of problem (24)–(25) with items in $A(h)$ fixed in the knapsack. Then:

$$\sum_{t=1}^{q} p_t \max\{F(t), Z^{\mathrm{KF}}(A(h),t)\} \geq Z^{\mathrm{SR}}.$$

Proof: Let $A^* = \{a_0=0, \ldots, a_{m(h)}, \ldots, a_m\}$ be the optimal visit-order, $L(t)$, $t = 1, \ldots, q$, the optimal abandon-location indices, and v_t^* the optimal time-limit violations for SR. If $L(t) \leq m(h)$ then $F(t)$ equals the optimal solution value contribution from scenario t, otherwise, following the proof of Proposition 3, $Z^{\mathrm{KF}}(t)$provides an upper bound on that value. The conclusion follows.□

5 Computational results

Formulation SR-1 was solved in CPLEX 10, but was only able to optimise problems with fewer than 5 locations in a reasonable time. Solution using formulation SR-2 was not tried as experience from Laporte and Martello (1990) suggests the enumeration algorithm is more likely to perform better.

The enumeration algorithm was coded in Visual Basic and tested on randomly generated instances. The programming environment use allowed for a fast development, but not necessarily the most efficient code. The instances used 8, 12, 16, 24 or 30 reward locations, 1, 2, 4, or 8 speed scenarios and three different time limits which were dependent on the instance. The time limits were set proportional to an estimate of the optimal TSP time at the average speed over all scenarios. The TSP time was estimated by summing, for each location, the average time to the 5 nearest locations. The time limits were set to be close to 0.1, 0.2 or 0.4 of this value. One would expect tour to reach 10%, 20% and 40%, respectively, of the locations in optimal Rogaine routes, for each of these time limits.

Locations were generated randomly over the unit square, with the base location constrained to be in a 0.4 by 0.4 square centred in the middle. Instances with locations closer than 0.03 were not used. Only one location-layout was generated for each number of rewards.

The speed probability distributions were symmetric with mean speed 1, and a higher probability of being closer to the mean.

Instances were run for a maximum of 10 minutes. Timing results are shown in Table 1.

Time limit proportion	Scena-rios	Locations 8	12	16	24	30
0.1	**1**	0	0	0.02	0.39	0.70
0.1	**2**	0	0	0.02	0.24	0.48
0.1	**4**	0	0	0.02	0.39	0.80
0.1	**8**	0	0	0.02	0.59	1.3
0.2	**1**	0	0.05	0.63	28	>600
0.2	**2**	0	0.11	0.47	31	>600
0.2	**4**	0.02	0.05	0.55	65	>600
0.2	**8**	0	0.05	0.89	>600	>600
0.4	**1**	0.03	1.8	>600	>600	>600
0.4	**2**	0.02	2.2	>600	>600	>600
0.4	**4**	0.02	2.3	>600	>600	>600
0.4	**8**	0.03	3.2	>600	>600	>600

Table 1: Solution times (in seconds) for 80 instances.

In Table 1 times are shown in seconds. Those showing 0 are less than 0.01 seconds, while those showing >600 were stopped after 10 minutes. Interestingly, for many of the instances, two scenario instances solved faster than the corresponding deterministic instances. The times only appear to increase linearly with the number of scenarios, although the number of test instances is too small to see this clearly. As expected solution times increased exponentially as the number of locations and the time limit increase.

Different branch and bound node selection polices were tried. Depth-first and breadth-first policies out performed policies which searched through bundles of nodes to find the largest or smallest upper bound. Depth-first search was marginally faster.

Examining the solution shows that often similar solutions appear for the deterministic and stochastic but the order of locations becomes very important. Further analysis is needed to determine how well the various solutions perform under out-of-sample scenarios.

6 Conclusions and Future Work

A stochastic version of the prize-collecting, subset-selection TSP was studied to give some insights into solutions and the feasibility of an optimisation algorithm. It does appear that the algorithm is viable, although one could expect the exponential nature of the algorithm to provide a limit on the size or relative time limits which can be feasibly solved by this method. The algorithm could be improved by the use of stronger lower and upper bounds. The stochastic nature of the problem did not appear to have as much effect on solution times as the number of locations or the relative time limit.

Future work will involve refining the algorithm, strengthening bounds and improving the time to calculate these. Alternative bounds also need to be tested. The test problems used do not allow control to find for which features of the problem the algorithm performs best and worst, and some form of instance-generation approach is needed to more fully test the algorithm.

7 References

Andreatta, G., and G. Lulli. 2008. "A multi-period TSP with stochastic regular and urgent demands." *European Journal of Operational Research* **185**: 122–132.

Feillet, Dominique, Pierre Dejax and Michel Gendreau. 2005. "Travelling salesman problems with profits." *Transportation Science* **39** 2 (May): 188–205.

Gordon, Samuel P. 2006 "Rogaining: a prize-collecting orienteering problem." *Proceedings of the 41st Annual Conference of the Operational Research Society of New Zealand: ORSNZ'06*: 173–182.

Laporte, Gilbert, and Silvano Martello. 1990. "The selective travelling salesman problem." *Discrete Applied Mathematics* **26**: 193–207.

Ramesh, R., Y.S. Yoon and M.H. Karwen. 1992. "An optimal algorithm for the orienteering tour problem." *INFORMS Journal on Computing* **4** (2): 155–165.

Tang, H., and E Miller-Hooks. 2005. "Algorithms for a stochastic selective travelling salesperson problem." *Journal of the Operational Research Society* **56**: 439–452.

Determining Degree Of Difficulty In Rogo, A TSP-based Paper Puzzle

Dr Nicola Petty, Dr Shane Dye
Department of Management
University of Canterbury
New Zealand
{shane.dye,nicola.petty}@canterbury.ac.nz

Abstract

Rogo®, a pencil and paper puzzle, is based on a subset-selection travelling salesperson problem with a known optimal score. There is an infinite number of possible Rogo puzzles, with at least twelve aspects which may be varied. In order for puzzles to be appealing, they should be difficult enough to be challenging and interesting, but not intractable or tedious. Through examination, mathematical modelling and experimentation on human subjects, we begin research into what elements affect the degree of difficulty of Rogo puzzles. Comparisons are made with other puzzles. Some preliminary results are given.

Key words: Travelling salesperson problem, puzzles.

1 Introduction

Rogos were invented in August 2009 and were developed during 2010, being released as an iPhone app in December 2010. In order to provide a graded level of difficulty which helps people learn and engage with the puzzle, we needed to explore what elements make a Rogo puzzle difficult to solve. This has strong parallels with the problem of instance generation for testing and developing heuristic solution methods.

2 What is Rogo

Rogo is a puzzle based on a prize-collecting, subset selection, Travelling Salesperson Problem set on a rectilinear grid. It was developed in 2009 by Petty and Dye, who have developed an algorithm to solve Rogo puzzles to optimality. Solving a Rogo puzzle involves finding the complete tour of a specified length, avoiding forbidden squares, to maximise the score. In the paper version of the puzzle, the "Best" score is given, along with a "Good" score. These provide targets and stopping criteria, without which the puzzle has limited appeal. All puzzles have a unique solution with regard to the prizes selected, though there may be slightly different loops possible. In Figure 1 a small Rogo is given for illustration, with the solution showing. Rogos can be any size, though the smallest sensible format is 6 by 4 squares, with a loop of length 10, and there seems to be little need to go beyond a total of 150 squares (16 by 9, 12 by 12, 15 by 10) with a loop length of 20.

Rogo has elements in common with Sudoku, mazes and word-search puzzles. Superficially it looks like Sudoku as it is played on a grid of squares, involves numbers, and requires logic to solve. However the similarity ends there. In Sudoku the numbers are merely representative objects, and could as easily be letters or symbols, whereas in

Rogo the values of the numbers are essential to the puzzle. In contrast with Sudoku, the paper version of Rogo requires the player to add, subtract and count. Like a maze, Rogo involves a certain degree of trial and error, and route-finding. Also like a maze, Rogo is completed as soon as the optimal route has been identified and the "Best" score obtained. Rogo is similar to a word-search puzzle in that it requires scanning of the game board, and pattern recognition.

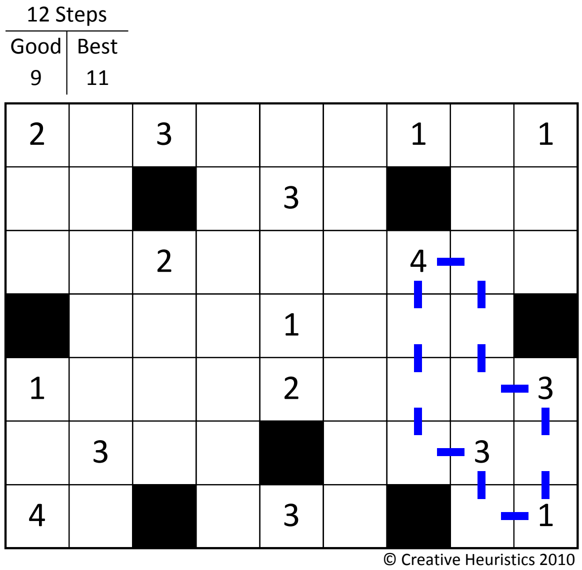


Figure 1: A Rogo puzzle, solved

3 What does a puzzle need to be appealing?

For a puzzle to be appealing and enjoyable, rather than a task, is somewhat personal, but there are some aspects which potentially affect the level of enjoyment. We suggest some of them are degree of difficulty, reward, and aesthetics.

If a puzzle is too easy, it is trivial and solved too quickly and if it is too hard, it can be frustrating and off-putting. This is different for different individuals.

Puzzles need to have an inherent reward. For many puzzles the reward is completion. Satisfaction comes from filling in all the squares in a Sudoku or crossword, or making the completed picture in a jigsaw. Rogo is different from these examples as the goal is not completion; there is no tidy up stage at the end. The sense of accomplishment or discovery comes when the route adds up to the target best score, and you have found the "best" route.

We also suggest that level of difficulty does not map directly onto solution time. Crosswords give a good example of this – some puzzles take a long time because they are large, with many squares. Others may be small and cryptic, and take a long time because of the thinking involved.

It can be suggested that enjoyment is a function of level of difficulty and solution time, and each person will have his or her person "enjoyment function" for a specific type of puzzle. This research focusses on the level of difficulty rather than on the level of enjoyment. Elapsed time to solve is used as a proxy for level of difficulty.

4 Elements of Rogo – what affects level of difficulty?

Twelve different dimensions or aspects of Rogo puzzles have been identified as potentially affecting the degree of difficulty. These are listed below, with a description and explanation as to how each may affect the level of difficulty. It is expected that the different elements would also interact in their effect on solution time.

4.1 Size of grid

The grids vary in size from a total of 24 to 150 squares. It is expected that larger puzzles take longer to solve as there is more area to scan for potential solutions.

4.2 Shape of the grid

The grids may be square or rectangular, landscape orientation or portrait. A narrow puzzle may speed up the search process.

4.3 Placement of black squares

Black (forbidden) squares form obstacles in the Rogo board. Some restrict the number of possible routes. This may make the puzzle instance easier by reducing the number of possible routes, or it can make it more difficult because numbers that look as if they could go together, can't.

4.4 Symmetry

Rotational and reflectional symmetry is often used in Rogo making in order to make the puzzles more pleasing to the eye. The symmetry relates to the placement of the forbidden squares and sometimes the placement of the numerals. This can speed up the search for an optimal solution as route shapes that are identified in one part of the board can be easily replicated in similar parts.

4.5 Number of prizes/density

The more dense the numerals are, the more computation is needed and too many numerals makes the game less interesting, as it is more difficult to get an idea by "eye" of where the high scores are. Blank squares are equivalent to a zero score, but because they are blank it is easier to spot patterns and potential routes in the puzzle.

4.6 Number of steps in the solution

Longer loops require more counting and adding, and have a larger number of possible shapes, thus increasing the solution time.

4.7 Shape of the solution

Shapes that have 'double-back's in them may be harder to find than, for instance, plain rectangles.

4.8 Location of the solution

It could be that people scan from left to right and top to bottom. Thus a solution in the top left would be more quickly found than one in the bottom right.

4.9 Variety in numbers used

If only a small range of numbers is used, this makes the logic and the computation easier. However, the effect of adding one number to a route is fairly consistent, compared with having a range of numbers.

4.10 Magnitude of numbers

Larger numbers make for a greater computational load, and make the marginal effect of adding another single number (reward) to a route greater.

4.11 Number of near misses

As we generate Rogo problem instances, we aim to make sure that there are sufficient routes, distributed throughout the grid, that provide scores close to the optimal. This stops the solution from being trivial.

4.12 Location of largest value rewards

A number noticeably greater in magnitude can draw the eye, and possibly anchor the players thinking. “Surely the 9 must be in the optimal route!” This could affect the solution time either way, depending on whether the high number really is in the optimal route or not.

5 Other work on puzzle difficulty and human performance

Previous research has looked at puzzle difficulty and human performance on puzzle solving. We examined work on Sudoku, TSP as a puzzle and the 15-puzzle.

Research on Sudoku has generally centred on developing a method for classifying puzzle difficulty. One approach for Sudoku is to use algorithms designed to mimic human puzzle solvers. As a logic-based puzzle there are essentially two main processes to use in solving Sudoku puzzles for both humans and computer algorithms. The first is to use logical reasoning to reduce the possible values that can be placed in a cell, isolating those for which only one value is feasible. The second process is trial-and-error, effectively branching over possible cell values. For computer algorithms differences in performance come from how the Sudoku problem is formulated.

For example, Henz and Troung (2009) formulate Sudoku puzzles as satisfiability (SAT) problems and present a tool, called SudokuSat. They classify Sudoku puzzles based on solving times. Chen (2009) formulates Sudoku as a graph colouring problem with edges connecting cells in the same row, column or block. The algorithm proposed uses limited logic levels and branching, where necessary. An index is defined based on the number of feasible values for each cell after limited preprocessing. This index is used to classify puzzle difficulty.

Human performance on the TSP problem has been studied. MacGregor and Ormerod (1996), MacGregor *et al* (2000), and Dry *et al* (2006), among others, tested human performance on visually presented TSP instances. They draw conclusions about the heuristics that might be employed by the problem-solvers and what aspects of the TSP made the problems difficult (in terms of solution time). Two aspects which appeared to impact on difficulty were the number of nodes and the number of nodes on the convex hull. To judge the quality of solutions produced these were compared to solutions found by a (computer) heuristic and to the average length of randomly sampled solutions.

Visually presented TSP instances have a number of features which can complicate conclusions about puzzle difficulty. The subjects generally produced a single solution (without improvement) which was often not the optimal solution. One effect of this is that solution times correspond to different quality solutions. Visual discrimination is also a factor in performance. It was noted by van Rooij *et al* (2003) that studies in which tours were produced by pencil produced proportionally more tours with crossings than those using a computer interface.

Pizlio and Li (2005) studied human performance on the 15-puzzle. They compared the number of solution steps taken with the minimum number required. They formulated a computational model to mimic the apparent human behaviour evident from the results.

6 The Pilot Study

As we are interested in the time that people take to solve Rogo puzzles, we did some preliminary research to see if there were consistencies between people, asking whether there are some puzzles that are more difficult than others for all or nearly all people.

We developed a set of Rogo puzzles which had a range of difficulties, we believed, based on a small sample. Testing sessions were set up, with up to six subjects at a time. The subjects were taught how to solve Rogos and timed as they completed each of the 12 puzzles. The puzzles were checked later to make sure they were correctly completed.

7 Preliminary results

There were 71 subjects in the sample for the pilot study. Summary data is given in Table 1. Scores over 600 seconds were removed. Similarly, puzzles 3, 7 and 12 had a large number of incorrect results or incompletes, so were removed before further analysis.

Puzzle Instance	1	2	3	4	5	6	7	8	9	10	11	12
Solved to Best	50	57	20	51	61	38	7	28	35	22	20	12
Solved to Good	6	6	7	10	5	17	23	14	12	11	12	14
Incorrect	9	6	30	3	1	6	16	11	4	8	3	2
Other	6	2	14	7	3	8	20	10	7	12	8	10
Attempted	71	71	71	71	70	69	66	63	58	53	43	38
Percentage Best	70%	80%	28%	72%	87%	55%	11%	44%	60%	42%	47%	32%
Mean (Best)	140	106	254	173	95	176	327	246	149	176	149	210
Standard deviation	89	82	183	110	89	148	135	173	90	113	107	160

Table 1: Summary data for the 12 puzzle instances.

Puzzle 8 was an interesting in that it had more outliers than the other puzzles. Even some of the “good” puzzlers (ones who solved 8 or more of the 12 puzzles) recorded very long times for puzzle 8. (The puzzle instances are given in the Appendix.)

The nature of the puzzles were examined to see if any could explain some of the variation in time taken to solve. With a sample of only 9 puzzles, this could only give an indication. The number of near misses (routes that scored 1 or 2 less than the ‘best’ score) was a significant predictor, with an R-sq value of 6% and a p-value of 0.000. This would suggest that the higher the number of near misses, the harder the puzzle is, which aligns with aspects of the method used for generating puzzle instances.

The differences in means between the 9 puzzles was examined. The ANOVA resulted in a p value of 0.000. This indicates that there is greater variation in the scores between the different puzzles than there is within the puzzles. Thus we can conclude that there is an element of universality in difficulty, which promises well for further research in this area. Graphs of the times taken for individuals for the puzzles showed little similarity, which indicates that though some puzzles are more difficult than others in general, it is by no means universal to all the puzzle-solvers.

The subjects were classified according to the number of puzzles they were able to solve successfully, with 0 to 4 being “weak”, 5 to 7, “medium” and 8 to 11 “good”. There was a significant difference in the average time taken for solving between the three groups. Some background information on the individuals, such as their puzzle-solving behaviour and ability at mathematics had no predictive ability on the time taken.

8 What next?

Many of the problems encountered in the paper-based experiment could be reduced or eliminated by the use of the electronic form of Rogo. This would ensure that the puzzles were solved correctly and would require the subject to deliberately choose to give up on a puzzle if they wanted to move on. An electronic form of Rogo would provide a quick and less labour intensive way to collect considerable data on solution times. In addition, the routes created in the course of solving the puzzle could be analysed to explore the kind of thinking used. This, in turn could be possibly be used to inform computer-based heuristic solution methods. The computational aspect of the puzzle is reduced with the electronic medium, as the adding and counting are done by the program. This could be turned off if required to evaluate the impact of computation on solution times.

9 Conclusions

The Rogo puzzle format has a number of aspects that can be controlled to potentially affect degree of difficulty of solving. As a pilot, this study showed that there are many aspects of puzzle-solving related to the nature of the puzzle that can be explored, and there appear to be some general effects, though there are still marked individual differences between people solving the puzzles. This research has the potential to provide interesting insights into both human behaviour, and the nature of puzzles.

References

Chen, Zhe. 2009. “Heuristic reasoning on a graph and game complexity of Sudoku.” *Computer Research Repository (CoRR)* arXiv:0903.1659 (http://arxiv.org/abs/0903.1659)

Dry, Matthew, Michael D. Lee, Douglas Vicker and Peter Hughes. 2006. “Human performance on visually presented traveling salesperson problems with varying numbers of nodes.” *The Journal of Problem Solving* **1** (1):20–32.

Henz, Martin, and Hoang-Minh Truong. 2009. “SudokuSat—a tool for analyzing difficult Sudoku puzzles.” In *Tools and Applications with Artificial Intelligence* – C. Koutsojannis and S. Sirmakessis (eds) SCI **166**: 25–35.

MacGregor, J.N. and T. Ormerod. 1996. “Human performance on the traveling salesman problem.” *Perception and Psychophysics* **58** (4):527–539.

MacGregor, J.N., T.C. Ormerod and E.P. Chronicle. 2000. "A model of human performance on the traveling salesperson problem." *Memory and Cognition* **28** (7): 1183–1190.
Pizlio, Zygmunt, and Zheng Li. 2005. "Solving combinatorial problems: the 15-puzzle" *Memory and Cognition* **33** (6): 1069–1084.
van Rooij, Iris, Ulrike Stege and Alissa Schactman. 2003. "Convex hull and tour crossings in the Euclidean traveling salesperson problem: implications for human performance studies." *Memory and Cognition* **31** (2):215–220.

Appendix

The 12 Rogo puzzles used in the experiment.

Trial 1 — 12 Steps — Good 19 | Best 21

2					1	
	2		2		■	5
		■	5			
	2	1		1	1	
			5	■		
1	■		2		5	
	5					2

Trial 2 — 12 Steps — Good 22 | Best 25

3							2
	3	■	2	3	■	3	
5		5			2		3
3		3			3		3
	3	■	2	2	■	2	
2							5

Trial 3 — 12 Steps — Good 22 | Best 23

■	4	2		2	5	■
5			■			2
5			4			5
	■	4		5	■	
2			5			2
4			■			5
■	2	4		4	2	■

Trial 4 — 12 Steps — Good 16 | Best 19

4		4		3	
		2		■	4
4		■			
■			2		3
3		4			■
			■		4
2	■		4		
	4		2		4

Trial 5 — 16 Steps — Good 24 | Best 28

5			5			3
	■				■	
		3		4		
5			■			4
		5		4		
	■				■	
3			3			5

Trial 6 — 16 Steps — Good 21 | Best 23

■	■	3		4	■
3		3			■
		■			1
1			4		3
1		3			1
4			■		
■			3		1
■	3		4	■	■

Trial 7

16 Steps

Good	Best
28	30

	2	4			2	
5				■		4
	■		5			5
		4	2	2		
2			4		■	
4		■				2
	4			2	4	

Trial 8

16 Steps

Good	Best
21	24

■	■		4	4		■	■
■	3					3	■
1							4
	3	■	1	3	■	3	
3			4	3			1

Trial 9

16 Steps

Good	Best
25	29

3			3			5
	■			2	■	
	5		3		2	
3	■	2	■	3	■	3
	3		5		2	
	■	2			■	
2			5			5

Trial 10

12 Steps

Good	Best
17	18

	4		2	4		4	
4							3
	3	■		2	■		
		■	3		■	3	
2							4
	3		3	4		3	

Trial 11

12 Steps

Good	Best
13	15

3			1			3
	3	■		■	2	
		1	3	2		
■	1				3	■
		2	1	1		
	3	■		■	3	
3			3			3

Trial 12

16 Steps

Good	Best
22	24

5	1			5	1
2					2
		■	■		
1		5		1	
	2		2		5
		■	■		
2					5
2	2			1	2

Public-Transit Frequency Setting Using Minimum-Cost Approach with Stochastic Demand

Yuval Hadas
Department of Civil and Environmental Engineering
University of Auckland, New Zealand
and
Department of management, Faculty of Social Sciences
Bar-Ilan University, Israel
y.hadas@auckland.ac.nz, yuval.hadas@biu.ac.il

Matan Shnaiderman
Department of management, Faculty of Social Sciences
Bar-Ilan University, Israel
matan.shnaiderman@biu.ac.il

Avishai (Avi) Ceder
Department of Civil and Environmental Engineering
University of Auckland, New Zealand
a.ceder@auckland.ac.nz

Abstract

Common practice in Public-Transit planning is to determine the frequency of service based on accumulated hourly passenger counts, average travel time, given vehicle capacity, and the standard of minimum frequency by time of day. With the increased usage of automatic vehicle location (AVL) and automatic passenger counting (APC) systems, it is possible to construct the statistical distributions of passenger demand and travel time by time of day. This can give rise to improve the accuracy of the determination of frequencies. This study presents a new approach enabling the use of stochastic properties of the collected data and its associated costs. An optimization framework is constructed based on two main cost elements: (a) empty-seat driven (unproductive cost), and (b) overload and un-served demand (increased user cost). The objective function is to minimize the total cost incurred with decision variables of either frequency or vehicle capacity (vehicle size). That is, from the operator perspective it is desirable to utilize efficiently the fleet of vehicles (decisions of the vehicle size). From the authority perspective, the concern is to provide an adequate level of service in terms of frequency. The study contains sensitivity analysis of the cost elements for economic evaluation.

Key words: Public-Transit, Frequency, Optimization.

1 Introduction

Public-Transit planning consists of five major components: network design, setting frequencies, timetable development, vehicle scheduling, and crew scheduling (Ceder and Wilson, 1986). Common practice in Public-Transit planning is to determine the frequency of service based on accumulated hourly passenger counts, average travel time, given vehicle capacity, desired occupancy (load standard) and the minimum frequency permitted by time of day (Ceder, 2007). Other models for frequency settings exists, such an economic costs model which is based on a simple route (Jara-Díaz et al., 2008). Another model introduces performance measures for crowdedness level (COD) and probability of not allowed to alight (POF) (Grosfeld-Nir and Bookbinder, 1995). This model assumes simple stochastic attributes of arrival and departure rates, and treats vehicle capacity as unlimited. An optimization model for headways was presented as part of a more general cost model for public transit route design (Kocur and Hendrickson, 1982), with the assumption of evenly spaced stops and demand.

Automatic Vehicle Location (AVL) and Automatic Passenger counting (APC) are two technologies enabling tracking the location of vehicles en route (AVL), and collecting the number of passengers alighted and boarded at each stop (APC). The data acquired can be used for analysis, as well as to enhance the performance of public transit systems, and the introduction of advanced models. It was shown that the availability of bus locations, estimated arrival times, number of passengers and their destinations, can open the door to the implementation of bus-dispatching at timed transfer transit stations algorithm (Dessouky et al., 1999). Such an algorithm can intelligently decide whether to hold a bus in order to achieve a transfer with a late bus or not. Based on AVL technology it is possible to forecast accurately the buses estimated arrival times and to use bus holding strategies to coordinate transfers (Dessouky et al., 2003). The use of advanced public transit systems in fixed-route and paratransit operations was found important for improvements in departure times and transfers (Levine et al., 2000). Travel time estimation is also possible (Tétreault and El-Geneidy, 2010), as well as the evaluation of transit operations based on both AVL and APC data (Strathman et al., 2002).

Advanced supply chain models, specifically inventory models (Lee et al., 2000, Zipkin, 2000), are used to optimize the total costs occurs within a time frame, by setting an inventory replenishment strategies (Kogan and Shnaiderman, 2010). The costs are associated with shortage and overage (or surplus). The former relevant when demand is not met, while the later when inventory is higher than the demand. Such models can be used to formulate an optimal frequency setting model, in which shortage and overage costs are transformed to overload and empty-seats costs respectively, and inventory strategy transforms to bus capacity.

This paper presents a new concept for optimal frequency setting, which based on supply chain models that integrates costs, stochastic demand and travel time; it is organized as follows. Section 2.1 introduces a general formulation of the model, followed by deterministic and stochastic variations of the model (Sections 2.2, 2.3, 2.4). Section 3 provides a comparison of the different models. Section 4 presents the conclusions and recommendations of study.

2 Cost-based approach of frequency setting

Consider a single route in which several vehicles of the same capacity, are serving the route at a fixed frequency (or headway). Let *i*, *i*=1..*I*, be the vehicle index, and let *k*, *k*=1..*K*, be the stop index. *CP* will denote the capacity of each vehicle. Let μ_k be the average travel time between stops *k* and *k*+1. Also, the passenger demand (the unconstrained load) from stop k will be denoted by d_{ik}. For simplicity, the model uses headways instead of frequencies, thus for a frequency *f* (vehicles per hour), the equivalent headway *H* (minutes between consecutive departures) is $H = 60/f$.

The study begins with a general formulation of the model including stochastic demand and distributed travel time. The study continuous with a deterministic model, stochastic demand model, and stochastic demand and distributed travel time.

2.1 Optimal cost-based frequency setting model

The study uses two common frequency setting methods. The first is based on a point check and called hourly max-load-point method and the second is based on ride check and called weighted average load profile method (Ceder, 2007); the latter method is introduced for comparison.

Assume that for a time period T (which is measured in minutes, and usually referred to one hour), the average demand at stop *k*, for every $1 \le k \le K-1$ is $\overline{D}_k$, and L_0 is the desired occupancy.

According to the hourly max load point method, the headway is determined by the highest loaded stop. Generally, the headway is equal to $L_0 T \Big/ \max_k \overline{D}_k$. However, there are two constraints which refer to the lower and upper bounds of possible headways. The lower bound is equal to T/N, where N denotes the number of vehicles available to serve during time period T. The upper bound is determined by the service level, and is denoted by H_0.

Taking into account the constraint

(1) $$\frac{T}{N} \le H \le H_0$$

the following headway is obtained

(2) $$H = \min\left\{ \max\left(\frac{L_0 T}{\max_k \overline{D}_k}, \frac{T}{N} \right),\ H_0 \right\}.$$

For the weighted average load profile method, we've substituted the distance travelled by the travel time, so a comparison with the general model is possible.

(3) $$L_0 T \sum_{k=1}^{K-1} \mu_k \Bigg/ \sum_{k=1}^{K-1} \left(\mu_k \sum_{i=1}^{I} d_{ik} \right).$$

with the constraint

(4) $$H \le \frac{CP \cdot T}{\max_k \overline{D}_k}.$$

Taking into account (3), (1) and (4), the following headway is attained:

(5) $$H = \min\left\{ \max\left(L_0 T \sum_{k=1}^{K-1} \mu_k \Big/ \sum_{k=0}^{K-1}\left(\mu_k \sum_{i=1}^{I} d_{ik} \right),\ \frac{T}{N} \right),\ H_0,\ \frac{CP \cdot T}{\max\limits_k \bar{D}_k} \right\}.$$

Formulas (2) and (5) are deterministic in nature; moreover, overload and empty-seats costs are not included. Hence, a new approach, based on supply chain principles (Kogan and Shnaiderman, 2010) is introduced.

Let c^+ be the empty seat overage cost per time unit, let c^- as the unserved passenger shortage cost per time unit, and let R_{ik} be the running time between stop $k-1$ to stop *k*, for vehicle *i*.

If the load is smaller than the capacity CP, then overage cost of vehicle i at stop k is equal to

(6) $$R_{ik} c^+ \max(CP - d_{ik}, 0).$$

On the other hand, if the load is higher than the capacity, then the shortage cost of vehicle i at stop k is equal to

(7) $$C_{ik}^- = R_{ik} c^- \max(d_{ik} - CP, 0).$$

The total cost of vehicle i at stop k is therefore the sum of (6) and (7):

(8) $$C_{ik} = R_{ik} c^+ \max(CP - d_{ik}, 0) + R_{ik} c^- \max(d_{ik} - CP, 0).$$

Proposition 1: Assuming that either a) R_{ik} are deterministic or b) The probabilities $\{\Pr(L_{ik} = n)\}_{n=0}^{\infty}$ do not depend on the exact distributions of R_{ik}, but only on their expected values, then the expected cost C_{ik} depends on the distribution of L_{ik} as follows:

(9) $$\mathrm{E}[C_{ik}] = \mu_k \sum_{n=0}^{CP} c^+ (CP - n) \Pr(L_{ik} = n \mid H) + \mu_k \sum_{n=CP+1}^{\infty} c^- (n - CP) \Pr(L_{ik} = n \mid H).$$

The objective function is the expected total cost for all vehicles and stops:

(10) $$\text{Minimize } \mathrm{ETC}(\mathrm{H,CP}) = \sum_{i=1}^{I} \sum_{k=1}^{K-1} \mathrm{E}[C_{ik}].$$

with decision variables H and CP.

Let L_{ik} be the load in vehicle i after departing from stop k:

(11) $$L_{ik} = \min(d_{ik}, CP).$$

Let A_{ik} denotes the number of passengers alighting from vehicle i at stop k, and B_{ik} the number of passengers arriving at stop k *after* the previous vehicle ($i-1$) has left. Also, let r_{ik} be the number of un-served passengers by vehicle i at stop k, that is

(12) $$r_{ik} = \max(d_{ik} - CP, 0).$$

Thus, the following formula is derived for the demand d_{ik}:

(13) $$d_{ik} = L_{i,k-1} - A_{ik} + r_{ik} + B_{ik} = \min(d_{i,k-1}, CP) - A_{ik} + \max(d_{i-1,k} - CP, 0) + B_{ik}$$

$2 \le i \le I$ and $2 \le k \le K-1$

with the initial conditions:

$$(14) \qquad \begin{cases} d_{1k} = L_{1,k-1} - A_{ik} + B_{ik} = \min(d_{i,k-1}, CP) - A_{ik} + B_{ik}, \quad k \ge 2 \\ d_{i1} = r_{i1} + B_{i1} = \max(d_{i-1,1} - CP, 0) + B_{i1}, \quad i \ge 2 \\ d_{11} = B_{11}. \end{cases}$$

Proposition 2: The probability of d_{ik} to be n, $n \ge 0, n \in \mathbb{Z}$ is:

$$(15) \quad \begin{cases} \Pr(d_{ik} = n) = \\ \displaystyle\sum_{\ell=0}^{n-1} \Pr(d_{i,k-1} = \ell) \left[\sum_{m=0}^{\ell} \Pr(A_{ik} = m \,|\, L_{i,k-1} = \ell) \begin{pmatrix} \displaystyle\sum_{j=0}^{CP-1} \Pr(d_{i-1,k} = j) \Pr(B_{ik} = n - \ell + m) + \\ \displaystyle + \sum_{j=CP}^{CP+n-\ell+m} \Pr(d_{i-1,k} = j) \Pr(B_{ik} = CP + n - \ell + m - j) \end{pmatrix} \right] + \\ \displaystyle + \sum_{\ell=n}^{CP} \Pr(d_{i,k-1} = \ell) \left[\sum_{m=\ell-n}^{\ell} \Pr(A_{ik} = m \,|\, L_{i,k-1} = \ell) \begin{pmatrix} \displaystyle\sum_{j=0}^{CP-1} \Pr(d_{i-1,k} = j) \Pr(B_{ik} = n - \ell + m) + \\ \displaystyle + \sum_{j=CP}^{CP+n-\ell+m} \Pr(d_{i-1,k} = j) \Pr(B_{ik} = CP + n - \ell + m - j) \end{pmatrix} \right] + \\ \displaystyle + \sum_{\ell=CP+1}^{\infty} \Pr(d_{i,k-1} = \ell) \left[\sum_{m=CP-n}^{CP} \Pr(A_{ik} = m \,|\, L_{i,k-1} = CP) \begin{pmatrix} \displaystyle\sum_{j=0}^{CP-1} \Pr(d_{i-1,k} = j) \Pr(B_{ik} = n - CP + m) + \\ \displaystyle + \sum_{j=CP}^{n+m} \Pr(d_{i-1,k} = j) \Pr(B_{ik} = n + m - j) \end{pmatrix} \right], \\ \hfill \text{if} \quad n \le CP \\ \\ \displaystyle\sum_{\ell=0}^{CP} \Pr(d_{i,k-1} = \ell) \left[\sum_{m=0}^{\ell} \Pr(A_{ik} = m \,|\, L_{i,k-1} = \ell) \begin{pmatrix} \displaystyle\sum_{j=0}^{CP-1} \Pr(d_{i-1,k} = j) \Pr(B_{ik} = n - \ell + m) + \\ \displaystyle + \sum_{j=CP}^{CP+n-\ell+m} \Pr(d_{i-1,k} = j) \Pr(B_{ik} = CP + n - \ell + m - j) \end{pmatrix} \right] + \\ \displaystyle + \sum_{\ell=CP+1}^{\infty} \Pr(d_{i,k-1} = \ell) \left[\sum_{m=0}^{CP} \Pr(A_{ik} = m \,|\, L_{i,k-1} = CP) \begin{pmatrix} \displaystyle\sum_{j=0}^{CP-1} \Pr(d_{i-1,k} = j) \Pr(B_{ik} = n - CP + m) + \\ \displaystyle + \sum_{j=CP}^{n+m} \Pr(d_{i-1,k} = j) \Pr(B_{ik} = n + m - j) \end{pmatrix} \right], \\ \hfill \text{if} \quad n \ge CP + 1 \end{cases}$$

$$\forall\ 2 \le i \le I,\ 2 \le k \le K-1$$

(where the probabilities $\{\Pr(d_{i-1,k} = j)\}$ are under the condition $d_{i,k-1} = \ell$, and the probabilities $\{\Pr(B_{ik} = J)\}$ are under the conditions $d_{i,k-1} = \ell$ and $d_{i-1,k} = j$).
The initial conditions are:

$$\Pr(d_{1k}=n)=$$

$$(16)\quad \begin{cases} \sum_{\ell=0}^{n-1}\Pr(d_{1,k-1}=\ell)\left[\sum_{m=0}^{\ell}\Pr(A_{1k}=m\,|L_{1,k-1}=\ell)\Pr(B_{1k}=n-\ell+m|d_{i,k-1}=\ell)\right]+ \\ \sum_{\ell=n}^{CP}\Pr(d_{1,k-1}=\ell)\left[\sum_{m=\ell-n}^{\ell}\Pr(A_{1k}=m|L_{1,k-1}=\ell)\Pr(B_{1k}=n-\ell+m|d_{i,k-1}=\ell)\right]+ \\ \sum_{\ell=CP+1}^{\infty}\Pr(d_{1,k-1}=\ell)\left[\sum_{m=CP-n}^{CP}\Pr(A_{1k}=m|L_{1,k-1}=CP)\Pr(B_{1k}=n-CP+m|d_{i,k-1}=\ell)\right], & \text{if} \quad n\le CP \\ \\ \sum_{\ell=0}^{CP}\Pr(d_{1,k-1}=\ell)\left[\sum_{m=0}^{\ell}\Pr(A_{1k}=m|L_{1,k-1}=\ell)\Pr(B_{1k}=n-\ell+m|d_{i,k-1}=\ell)\right]+ \\ \sum_{\ell=CP+1}^{\infty}\Pr(d_{1,k-1}=\ell)\left[\sum_{m=0}^{CP}\Pr(A_{1k}=m|L_{1,k-1}=CP)\Pr(B_{1k}=n-CP+m|d_{i,k-1}=\ell)\right], & \text{if} \quad n\ge CP+1 \end{cases}$$

$$\forall\ 2\le k\le K-1$$

$$\Pr(d_{i1}=n)=\sum_{j=0}^{CP}\Pr(d_{i-1,1}=j)\Pr(B_{i1}=n|d_{i-1,1}=j)+$$

$$(17)\quad \sum_{j=1}^{n}\Pr(d_{i-1,1}=CP+j)\Pr(B_{i1}=n-jd_{i-1,1}=CP+j)$$

$$\forall\ 2\le i\le I$$

$$(18)\qquad \Pr(d_{11}=n)=\Pr(B_{11}=n).$$

2.2 Optimal frequency setting with deterministic data

In this case we assume that the travel and dwell times are all deterministic. Assume that the rate in which passengers arrive to stop k (denoted by λ_k) is known, then the variable B_{ik}, which is approximately equal to $\lambda_k H$, is necessarily equal to some $n_k^{(B)}$ which satisfies $\left|n_k^{(B)}-\lambda_k H\right|=\min_{n\in\mathbb{N}\cup\{0\}}\left|n-\lambda_k H\right|$, to obtain

$$(19)\qquad \Pr(B_{ik}=n)=\begin{cases}1, & n=n_k^{(B)}\\ 0, & n\neq n_k.\end{cases}$$

Also, we assume that at stop k it is known that p_k of the passengers alight ($0\le p_k\le 1$), then A_{ik} is necessarily equal to a value $n_k^{(A)}$ such that $\left|n_k^{(A)}-p_k L_{i,k-1}\right|=\min_{n\in\mathbb{N}\cup\{0\}}\left|n-p_k L_{i,k-1}\right|$, that is

$$(20)\qquad \Pr(A_{ik}=n)=\begin{cases}1, & n=n_k^{(A)}\\ 0, & n\neq n_k.\end{cases}$$

Therefore, the value of d_{ik} is well known for every $1\le i\le I$ and $1\le k\le K-1$, and it is calculated from (13)-(14). The expected cost is then found by (8).

2.3 Optimal frequency setting with stochastic demand

For this model we assume that demand is stochastic, the variables $\{\lambda_k, p_k\}$ are stochastic and finite-valued. That is, for every $1 \le k \le K-1$, $\lambda_k \in \{\lambda_{k_1}, \ldots, \lambda_{k_{j_1(k)}}\}$, such that for every $1 \le \ell \le j_1(k)$ $\Pr(\lambda_k = \lambda_{k_\ell}) = q_{k_\ell}^{(B)}$, and for every $2 \le k \le K-1$, $p_k \in \{p_{k_1}, \ldots, p_{k_{j_2(k)}}\}$, such that for every $1 \le \ell \le j_2(k)$ $\Pr(p_k = p_{k_\ell}) = q_{k_\ell}^{(A)}$.

In order to calculate the probabilities (15)-(18), we define for every $1 \le \ell \le j_1(k)$ the index $n_{k_\ell}^{(B)}$ which satisfies $\left| n_{k_\ell}^{(B)} - \lambda_{k_\ell} H \right| = \min_{n \in \mathbb{N} \cup \{0\}} \left| n - \lambda_{k_\ell} H \right|$.

In addition, given a value of $L_{i,k-1}$, we define the index $n_{k_\ell}^{(A)}$ which satisfies $\left| n_{k_\ell}^{(A)} - p_{k_\ell} L_{i,k-1} \right| = \min_{n \in \mathbb{N} \cup \{0\}} \left| n - p_{k_\ell} L_{i,k-1} \right|$ for $1 \le \ell \le j_2(k)$.

Therefore, the probabilities (19) and (20) become now

(21) $$\Pr(B_{ik} = n) = \begin{cases} q_{k_1}^{(B)}, & n = n_{k_1}^{(B)} \\ \vdots & \\ q_{k_{j_1(k)}}^{(B)}, & n = n_{k_{j_1(k)}}^{(B)} \\ 0, & \text{else} \end{cases}$$

and

(22) $$\Pr(B_{ik} = n) = \begin{cases} q_{k_1}^{(A)}, & n = n_{k_1}^{(A)} \\ \vdots & \\ q_{k_{j_2(k)}}^{(A)}, & n = n_{k_{j_2(k)}}^{(A)} \\ 0, & \text{else} \end{cases}$$

respectively.

2.4 Optimal frequency setting with stochastic demand and travel time

This section incorporates the general cost-based model with properties from a previous work that modelled stochastic demand and dwell time (Bellei and Gkoumas, 2010, Hickman, 2001).

Let D_{ik} be the dwell time of vehicle i at stop k. Also, let H_{ik} be the headway between the departure times of vehicles $i-1$ and i (for $2 \le i \le I$) at stop k, then:

(23) $$\begin{cases} H_{i1} = H + D_{i1} - D_{i-1,1} \\ H_{ik} = H_{i,k-1} + R_{ik} - R_{i-1,k} + D_{ik} - D_{i-1,k}, \quad k \ge 2 \end{cases}$$

(Hickman, 2001).

Assuming that the headways $\{H_{ik}\}$ for $2 \le i \le I$ and $1 \le k \le K-1$ are stochastic, and that boarding is performed *after* alighting (there are researches which assume that these events are performed in parallel, for example Sun and Hickman, 2008), the stochastic dwell time of vehicle i at stop k is equal to

(24) $$D_{ik} = a + b_A A_{ik} + b_B (r_{i-1,k} + B_{ik}),$$

where a denotes the lost time due to accelerating and decelerating of the vehicles at each stop, b_A denotes the incremental time for a single passenger to alight from the vehicle, and b_B denotes the incremental time for a single passenger to board the vehicle. The travel times between the stops are stochastic as well, satisfying an autoregressive process AR(1)

$$(25)\qquad \begin{cases} R_{1k} - \mu_k = \varepsilon_{1k} \\ R_{ik} - \mu_k = \alpha_k (R_{i-1,k} - \mu_k) + \varepsilon_{ik}, \quad 2 \le i \le I \end{cases}$$

for $2 \le k \le K$ (Mishalani et al., 2008). The parameter α_k in (25) denotes the correlation between the travel times of two consecutive vehicles, and $0 \le \alpha_k < 1$. In the special case of $\alpha_k = 0$, the travel times (from stop $k-1$ to stop k) of all the vehicles are identically distributed and independent. Also, the means of the noises $\{\varepsilon_{ik}\}$ are all zero. From (25) we have $R_{ik} = \mu_k + \sum_{j=1}^{i} \alpha^{i-j} \varepsilon_{jk}$, and therefore, while there are no real-time updates, the expected value of R_{ik} is μ_k, namely

$$(26)\qquad \mathrm{E}[R_{ik}] - \mathrm{E}[R_{i-1,k}] = 0.$$

The variable B_{ik} is distributed according to a Poisson process with mean $\lambda_k \mathrm{E}[H_{ik}]$ (Hickman, 2001). Therefore, the probability (21) becomes

$$(27)\qquad \Pr(B_{ik} = n) = \frac{\left(\lambda_k \mathrm{E}[H_{ik}]\right)^n}{n!} e^{-\lambda_k \mathrm{E}[H_{ik}]}.$$

Given the value of the load $L_{i,k-1}$, the variable A_{ik} is binomially distributed such that $A_{ik} \sim \mathrm{B}(L_{i,k-1}, p_k)$ for some $0 \le p_k \le 1$. Thus, (22) is now equal to

$$(28)\qquad \Pr(A_{ik} = n) = \binom{L_{i,k-1}}{n} p_k^n (1 - p_k)^{L_{i,k-1} - n}.$$

In order to calculate the probabilities (15)-(18) the calculation of the expected headway H_{ik} ought to take place (see (27)) for $2 \le i \le I$ and $1 \le k \le K-1$. The following proposition is then used:

Proposition 3: The expected headways at stop 1 are

$$(29)\qquad \begin{cases} \mathrm{E}[H_{21}] = \dfrac{H + b_B\left(\mathrm{E}[r_{11}] - \lambda_1 H\right)}{1 - \lambda_1 b_B} \\ \mathrm{E}[H_{i1}] = \dfrac{H + b_B\left(\mathrm{E}[r_{i-1,1}] - \mathrm{E}[r_{i-2,1}] - \lambda_1 \mathrm{E}[H_{i-1,1}]\right)}{1 - \lambda_1 b_B}, \quad i > 2. \end{cases}$$

The expected headways at stop $k \ge 2$ are

$$(30)\qquad \begin{cases} \mathrm{E}[\mathrm{H}_{2k}] = \dfrac{\mathrm{E}[H_{i,k-1}] + b_A p_k \left(\mathrm{E}[L_{2,k-1}] - \mathrm{E}[L_{1,k-1}]\right) + b_B\left(\mathrm{E}[r_{1k}] - \lambda_k H\right)}{1 - \lambda_k b_B} \\ \mathrm{E}[\mathrm{H}_{ik}] = \dfrac{\mathrm{E}[H_{i,k-1}] + b_A p_k \left(\mathrm{E}[L_{i,k-1}] - \mathrm{E}[L_{i-1,k-1}]\right) + b_B\left(\mathrm{E}[r_{i-1,k}] - \mathrm{E}[r_{i-2,k}] - \lambda_k \mathrm{E}[H_{i-1,k}]\right)}{1 - \lambda_k b_B}, \quad i > 2. \end{cases}$$

Furthermore, it is required that

(31) $\lambda_k b_B < 1$.

Note that from (11) and (12) we respectively obtain:

$\mathrm{E}[L_{ik}] = \sum_{n=0}^{CP} n \Pr(d_{ik} = n) + CP \sum_{n=CP+1}^{\infty} \Pr(d_{ik} = n)$ and $\mathrm{E}[r_{ik}] = \sum_{n=CP+1}^{\infty} (n - CP) \Pr(d_{ik} = n)$.

3 Evaluation of Models

Evaluating the performance and the benefits of the models was carried out with the following simple example. Consider a route of five stops, with vehicle capacity $CP = 80$, the arrival rates (per one minute) to the stops are $\lambda_1 = 3.1, \lambda_2 = 1.4, \lambda_3 = 0.65, \lambda_4 = 0.35$, the alighting passengers fractions are $p_2 = 0.09, p_3 = 0.19, p_4 = 0.42$, and the average travel times are $\mu_2 = 10, \mu_3 = 6, \mu_4 = 20, \mu_5 = 3$. According to the Ceder's model (2007), the average loads at stops 1,2,3 and 4 (during 3 hours) are 558, 760, 733 and 488. The suggested headway, according to the weighted average load profile(5), is 17 minutes. A comparison between the four models is summarized in Table 1. Also, a comparison of the expected costs obtained from each model to those obtained from the weighted average load profile is presented.

Table 1 - Frequency setting models' comparison

Model	Variance	$C^+=C^-$	$C^+<C^-$	$C^+>C^-$
Weighted ave. load profile	-	17	17	17
Deterministic	-	19 (26%)	18 (9%)	26 (79%)
Stochastic demand	Low	20 (36%)	18 (9%)	27 (77%)
	High	21 (11%)	15 (20%)	30* (60%)
Stochastic demand and travel time	Low	20 (33%)	17 (0%)	27 (75%)
	High**	7 (68%)	6 (60%)	7 (78%)

* The optimal headway is higher, but due to service level we assumed an upper bound of 30.

** The use of Poisson and Binomial distributions affected the result of the weighted average load profile, thus the stochastic demand and travel time model were compared with 3 minutes.

The results show that both the ratio between the costs and the variability of demand and travel time affect the optimal headway. High variability cause extreme headway, while a high empty-seats cost tends to increase the headway, in contrast to high overload cost which obviously decrease headway (and increase the level of service).

4 Conclusions

An optimal model was developed taking into account the costs associated with running empty-seats and passenger overload, and considering stochastic demand and distributed travel time. The model provides optimal frequency, based on quantitative costs. Moreover, the model reflects the necessary adjustments of the frequency because of the fluctuation of travel time and demand.

The availability of AVL and APC data across transit agencies and operators, justifies the development of models that estimate the demand and travel time (ride time and dwell time) statistical distributions. These data are key input to the optimal frequency setting model.

The sensitivity analysis of the costs introduced portrays the avenues for the authorities and operators to attain a better decision-making process to reach an improved service and more efficient use of resources.

As both frequency and vehicle capacity are decision variables, an optimal capacity variation of the model was developed.

References

G. Bellei, Gkoumas, K. 2010. Transit vehicles' headway distribution and service irregularity. *Public Transport* 1-21.

A. Ceder. 2007. *Public Transit Planning and Operation: Theory, Modeling and Practice*. Butterworth-Heinemann, Oxford, UK.

A. Ceder, Wilson, N.H.M. 1986. Bus network design. *Transportation Research B*. **20**(4) 331-344.

M. Dessouky, Hall, R., Nowroozi, A., Mourikas, K. 1999. Bus dispatching at timed transfer transit stations using bus tracking technology. *Transportation Research Part C: Emerging Technologies*. **7**(4) 187-208.

M. Dessouky, Hall, R., Zhang, L., Singh, A. 2003. Real-time control of buses for schedule coordination at a terminal. *Transportation Research Part A: Policy and Practice*. **37**(2) 145-164.

A. Grosfeld-Nir, Bookbinder, J.H. 1995. The planning of headways in urban public transit. *Annals of Operations Research*. **60**(1-4) 145-160.

M.D. Hickman. 2001. An Analytic Stochastic Model for the Transit Vehicle Holding Problem. *Transportation Science*. **35**(3) 215.

S. Jara-Díaz, Tirachini, A., Cortés, C.E. 2008. Modeling public transport corridors with aggregate and disaggregate demand. *Journal of Transport Geography*. **16**(6) 430-435.

G. Kocur, Hendrickson, C. 1982. Design of Local Bus Service with Demand Equilibration. *Transportation Science*. **16**(2) 149.

K. Kogan, Shnaiderman, M. 2010. Continuous-Time Replenishment Under Intermittent Observability. *Automatic Control, IEEE Transactions on*. **55**(6) 1460-1465.

H.L. Lee, So, K.C., Tang, C.S. 2000. The Value of Information Sharing in a Two-Level Supply Chain. *Management Science*. **46**(5) 626.

J. Levine, Hong, Q., Hug, G., Jr, Rodriguez, D. 2000. Impacts of an Advanced Public Transportation System Demonstration Project. *Transportation Research Record*(1735) 169-177.

R.G. Mishalani, McCord, M.R., Forman, S. 2008. *Schedule-based and autoregressive bus running time modeling in the presence of driver-bus heterogeneity*. City.

J.G. Strathman, Kimpel, T.J., Dueker, K.J., Gerhart, R.L., Callas, S. 2002. Evaluation of transit operations: data applications of Tri-Met's automated Bus Dispatching System. *Transportation*. **29**(3) 321-345.

A. Sun, Hickman, M. 2008. *The Holding Problem at Multiple Holding Stations*. Springer Berlin Heidelberg, City.

P.R. Tétreault, El-Geneidy, A.M. 2010. Estimating bus run times for new limited-stop service using archived AVL and APC data. *Transportation Research Part A: Policy and Practice*. **44**(6) 390-402.

P.H. Zipkin. 2000. *Foundations of inventory management*. McGraw-Hill, Boston.

A Decision Support Tool for Equipment Replacement in Forestry Harvesting Operations

Matthew Baxter
Department of Mathematics and Statistics
University of Melbourne
Australia
m.baxter3@pgrad.unimelb.edu.au

Mark Brown
Department of Forestry and Ecosystem Science
University of Melbourne
Australia
mwbrown@unimelb.edu.au

Heng-Soon Gan
Department of Mathematics and Statistics
University of Melbourne
Australia
hsg@unimelb.edu.au

Abstract

While managing forests for diverse and conflicting outcomes requires great care and patience over many years, the largest expense associated with producing timber products is the cost of harvesting. The contractors responsible for this task must invest in several types of expensive harvesting machinery with limited lifespans, with the goal of minimising their operating cost per tonne including machine ownership costs. Much work has been done on calculating the approximate total cost of owning a machine, given its expected lifespan and other parameters. While extremely useful, these calculations neglect several effects that can only be seen by considering a machine as a part of the larger operation. In this paper, we describe several mixed-integer linear programming models, with varied levels of complexity, developed to make decisions regarding equipment replacement. We further describe a prototype decision support tool to be used by machine owners.

Key words: forestry operations, equipment replacement, mixed integer linear programming, decision support tool

1 Introduction

Much work has been done on calculating the approximate total cost of owning a machine, given it's expected lifespan and other parameters. While extremely useful, these calculations neglect several effects that can only be seen by considering a machine as a part of the larger operation. The operation can only be as productive as its least productive class of machines, and this should be considering during purchasing decisions. The lifespan of a machine impacts its total cost of ownership considerably. Currently, the manufacturer's recommended lifespan is used in calculations, but it may be more cost effective in the long term to replace the machine earlier or to keep it running well past its recommended lifespan. This project will consider the cost of machines as a long-term planning problem.

Contractors are engaged by forest owners/managers, typically on a five-year contract, to harvest products from specified strands of trees. These contactors operate several classes of machines, each class having its own role, with the set-up of the operation determining which classes are required. For example, an operation producing woodchips may use Feller-Bunchers to fell and bunch trees, a skidder to move them to roadside then a Chipper to create woodchips ready to be trucked to the client.

With the exception of trucks, a contractor is likely to have no more than a few of each class of machine. Additionally, the cost of purchasing and maintaining these machines is a large proportion of a contractor's costs. For these reasons, deciding when to replace a machine and selecting a model to replace it with is a highly important problem.

The paper will consider the Equipment Replacement Problem in forestry harvesting operations. Typically, we aim to answer questions including "Which setup to use?", "What models to buy? (size, power, manufacturer...)", "When to buy and salvage machines?", "How to distribute the workload when the machines have greater capacity than required?" and "How much should the Forest Owner be charged for harvesting?".

If the equipment is always replaced with identical equipment (with identical costs over its age) and the time horizon is infinite (or only the first replacement is considered), then the optimal solution is to always replace the equipment at the end of its economic lifetime. This is the lifetime which minimises its average cost per unit of time (including purchase cost). This approach can be adjusted to account for technological improvement (Rogers and Hartman 2005) and/or inflation (a linear relationship between cost and time).

With a finite time horizon and discrete time periods, a dynamic programming algorithm exists which finds optimal solutions for the single machine case. (Waddell 1983) describes such an approach. This algorithm runs quickly enough to be run for large time horizons, and does not require the assumption that costs are linear with regard to time. However the algorithm is still limited in that it is designed for the single machine problem, and expanding it to multiple machines requires the assumption that there is no interdependence between the decisions.

The common approaches to equipment replacement used in Forestry ignore any and all interdependency between replacement decisions and do not attempt to calculate the economic lifetime of equipment. See for example (Brinker et al. 2002; Murphy and Associates 2008; Bilek 2007; Bilek 2009).

(Burt et al. 2010; Burt 2008)'s work, which is applied to the mining industry, is by far the closest analogue to the problem faced by the forestry industry. It involves a Multi–Period Equipment Selection problem (with heterogeneous fleets), which is largely equivalent to the type of Equipment Replacement problem which is described here. A brief section of this paper will be dedicated to compare and contrast (Burt et al. 2010; Burt 2008)'s model with our model.

The plan of this paper is as follows. We will start by describing various mathematical formulations in §2. The decision support tool is described in §3, and we conclude in §4.

2 Mathematical Formulations

The purpose of this model is to create a strategic plan for buying machines and distributing the work between them in order to maximise profits. The model is able to select between different possible contracts and configurations in order to meet these goals. However it is expected that in most applications the user will already have a contract in mind (and can make an educated guess regarding future contracts) and will be locked into a particular configuration. The contract income can be ignored by the model in those cases.

Each year is modelled as a discrete period, and the model has a finite time horizon of T years. The following concepts will be used throughout the model:

Contract: A contract specifies an amount of work to be done for a given price. For each possible contract this model requires knowledge of the amount of work required to fulfill the contract and the expected profit before machine costs. Details given in the section on parameters.

Time block: This is a set of consecutive time periods, blocked to represent the effective duration of a contract. This will typically be a five-time-period block.

Machine group: This is also known as a machine class or machine type. For example; feller-buncher, processor, forwarder or loader. There exist different makes of machine in each machine group.

Harvest system: This is the set of machine groups required to work together to convert "raw materials" into "finished products". Only one harvest system is chosen for a given contract over the effective time block.

Key to the understanding of the problem is the knowledge that forestry contractors cannot simply produce as much product as they wish in order to increase profits. They have a maximum level of production specified by the contract. Most forestry contracts allow for less product to be produced, so it could on occasion be more profitable not to reach the contract figure in a given year. However this is considered poor practice and may jeopardise the contractors reputation and future contracts. Therefore this model considers the contracted amount of work to be a hard constraint, and when used to choose between models considers their net profit before machine costs (ignoring the price per tonne).

The contract will generally give this constraint in tonnes per year whereas this model needs to know how much work is required from each machine group (given a particular harvest system) for every year. The relationship between the two is not straightforward; the amount of work per tonne varies immensely, and depends on different factors for different machine groups. Tools such as ALPACA(Murphy and Associates 2008) could be useful in calculating one from the other and experienced contractors are likely to be able to give good estimates.

Machines also vary in performance, will break down more often as they get older and may suffer additional performance penalties. This is accounted for in our model using piecewise linear functions to model the actual productive hours.

The index sets used by the model are:

$\mathbf{T}$	:	the set of time periods $\{1, \ldots, T\}$ (one time period equals one year)
$\mathbf{B}$	:	set of time blocks (one time block equals five consecutive time periods)
$\mathbf{T}_b \subset \mathbf{T}$	:	set of time periods in block $b \in \mathbf{B}$ (mutually exclusive)
$\mathbf{G}$	:	set of machine groups (classes or types)
$\mathbf{M}$	:	set of machines
$\mathbf{M}_g \subset \mathbf{M}$	:	set of machines in the machine group $g \in \mathbf{G}$
$\mathbf{C}$	:	set of contracts
$\mathbf{C}_b \subset \mathbf{C}$	:	set of contracts in time block $b \in \mathbf{B}$
$\mathbf{O}_c$	:	set of harvest systems for contract $c \in \mathbf{C}$

Parameters:

T	:	planning horizon
H	:	number of working hours in a year.
C_{mij}	:	unit cost of purchasing machine $m \in \mathbf{M}$ at the START of, period $i \in \mathbf{T}$ and salvaging it at the END of period $j \in \mathbf{T}$ (include interest, inflation, depreciation rates and insurance costs)
R_c	:	the net profit expected on contract $c \in \mathbf{C}$ before machine costs
W_{gkt}	:	the amount of productivity required from machine group $g \in \mathbf{G}$ in time period $t \in \mathbf{T}$ if the harvest system $k \in \mathbf{O}_c$ is chosen for contract $c \in \mathbf{C}$

Decision variables:

x_{kb}	=	1 if harvest system $k \in \mathbf{O}_c$ is chosen for contract $c \in \mathbf{C}_b$ in time block $b \in \mathbf{B}$, 0 otherwise
n_{mij}	=	the number of machines of type $m \in \mathbf{M}$ are purchased at the START of period $i \in \mathbf{T}$, and salvaged at the END of period $j \in \mathbf{T}$ (note: the (i, j) combinations may be restricted by the machine replacement policy;)
w_{mijt}	=	the amount of scheduled hours done by one unit of machine of type m, which were bought at time i and sold at j, during time period t. (We assume here that all machines of type m bought at i and salvaged at the end of j is given the same workload, thus giving a total workload of $n_{mij}w_{mijt}$ for this set of machines for time period t.)

Let $f_m(\theta)$ be a piecewise linear function that gives the total amount of work (in idealised productive hours) that has been done by *a machine* of age θ (in scheduled hours) and $g_{mt}(\theta)$ be a piecewise linear function that gives the maintenance costs incurred by that machine during time period t. The function $g_{mt}(\theta)$ is in dollars, defined for each time period t to support adjustments for the time value of money. We assume, without loss of generality, that the linear intervals of f_m and g_{mt} (for all t) are the same.

We designate the following mathematical program as model FHER. Expressing in general terms for functions f_m and g_{mt}, model FHER aims to maximise

$$\sum_{b\in\mathbf{B}}\sum_{c\in\mathbf{C}_b}\sum_{k\in\mathbf{O_c}} R_c x_{kb} - \sum_{m\in\mathbf{M}}\sum_{i,j\in\mathbf{T}} C_{mij} n_{mij}$$
$$- \sum_{m\in\mathbf{M}}\sum_{i,j\in\mathbf{T}}\sum_{i\le t\le j} n_{mij}\left[g_{mt}\left(\sum_{\tau=i}^{t} w_{mij\tau}\right) - g_{mt}\left(\sum_{\tau=i}^{t-1} w_{mij\tau}\right)\right] \tag{1}$$

such that:

- At most one harvest system in each time block:

$$\sum_{c\in\mathbf{C_b}}\sum_{k\in\mathbf{O}_c} x_{kb} \le 1, \qquad \forall b\in\mathbf{B} \tag{2}$$

- Minimum productivity:

$$\sum_{m\in\mathbf{M}_g}\sum_{i\in\mathbf{T}:i\le t}\sum_{j\in\mathbf{T}:j\ge t} n_{mij}\left[f_m\left(\sum_{\tau=i}^{t} w_{mij\tau}\right) - f_m\left(\sum_{\tau=i}^{t-1} w_{mij\tau}\right)\right]$$
$$\ge \sum_{k\in\mathbf{O}_c} W_{gkt}x_{kb}, \qquad \forall b\in\mathbf{B}, t\in\mathbf{T}_b, c\in\mathbf{C}_b, g\in\mathbf{G} \tag{3}$$

- There is a limit on scheduled machine hours.

$$w_{mijt} \le H, \qquad \forall m\in\mathbf{M},\ i,j,t\in\mathbf{T}: i\le t, j\ge t \tag{4}$$

- Variable signs:

$$\mathbf{x}\in\{0,1\},\ \mathbf{n}\ \mathit{integer},\ \mathbf{w}\ge 0 \tag{5}$$

Piecewise linear functions f_m and g_{mt} will be linearised so that FHER is a mixed-integer linear program. We describe the linearisation below.

2.1 Modelling functions f_m and g_{mt}

Recall that we give all machines bought and salvaged together the same workloads, thus we use the notation (m, i, j) to refer to machines of type m which are bought in time period i and salvaged in time period j. Also we assume that the linear intervals of functions f_m and $g_{mt}, \forall t\in\mathbf{T}$ are the same. We define the parameters variables below to capture the piecewise linear aspect of functions f_m and g_{mt}:

Z_m	=	the number of intervals for the functions for machine $m\in\mathbf{M}$
ζ_{mz}	=	the right endpoint of interval $z\in\{0,\ldots,Z_m\}$ for machine type $m\in\mathbf{M}$, so that f_m and g_{mt} have intervals

$[\zeta_{m0}, \zeta_{m1}]$, $[\zeta_{m1}, \zeta_{m2}]$,. . . , $[\zeta_{m(Z_m-1)}, \zeta_{mZ_m}]$ (let $\zeta_{m0} = 0$)

Γ_{mz} = the uptime and efficiency of machine type $m \in \mathbf{M}$ when its age is in interval $z \in \{1, \ldots, Z_m\}$

Ω_{mtz} = the age-dependent cost per scheduled hour of machine type $m \in \mathbf{M}$ when its age is in interval $z \in \{1, \ldots, Z_m\}$ (adjusted for inflation at time t)

Function f_m can be expressed as follows, in terms of the parameters described above:

$$f_m(\sigma) = \begin{cases} \Gamma_{m1}\sigma, & 0 \leq \sigma \leq \zeta_{m1}; \\ \Gamma_{mk}\left(\sigma - \sum_{l=1}^{k-1}\sigma_{ml}\right) + \sum_{l=1}^{k-1}[\Gamma_{ml}(\zeta_{ml} - \zeta_{m(l-1)})], & \zeta_{m(k-1)} \leq \sigma \leq \zeta_{mk} \\ & for\ k = 2, \ldots, Z_m; \end{cases} \tag{6}$$

Function g_{mt} can be expressed similarly by writing Ω_{mtz} in place of Γ_{mz}. We define the following decision variables to describe the piecewise linear functions:

α_{mijtz} = 1 if the machines (m, i, j) are in the interval $z \in \{1, \ldots, Z_m\}$ at time t, 0 otherwise;

y_{mijtz} = the coefficient of point $(\zeta_{mz}, f_m(\zeta_{mz}))$ or $(\zeta_{mz}, g_{mt}(\zeta_{mz}))$ at time t, for machines (m, i, j), where $z \in \{0, \ldots, Z_m\}$

We can replace terms involving f_m, g_{mt} and w_{mijt} in the objective function (1) and constraints (3), (4), (5) as follows:

$$\begin{aligned} &n_{mij}\left[f_m\left(\sum_{\tau=i}^{t} w_{mij\tau}\right) - f_m\left(\sum_{\tau=i}^{t-1} w_{mij\tau}\right)\right] \\ &= \sum_{z=1}^{Z_m} y_{mijtz} f_m(\zeta_{mz}) - \sum_{z=1}^{Z_m} y_{mij(t-1)z} f_m(\zeta_{mz}) \end{aligned} \tag{7}$$

$$\begin{aligned} &n_{mij}\left[g_{mt}\left(\sum_{\tau=i}^{t} w_{mij\tau}\right) - g_{mt}\left(\sum_{\tau=i}^{t-1} w_{mij\tau}\right)\right] \\ &= \sum_{z=1}^{Z_m} y_{mijtz} g_{mt}(\zeta_{mz}) - \sum_{z=1}^{Z_m} y_{mij(t-1)z} g_{mt}(\zeta_{mz}) \end{aligned} \tag{8}$$

$$0 \leq w_{mijt} \leq H \Rightarrow 0 \leq \sum_{z=1}^{Z_m} y_{mijtz}\zeta_{mz} - \sum_{z=1}^{Z_m} y_{mij(t-1)z}\zeta_{mz} \leq Hn_{mij} \tag{9}$$

and append the following constraints to model FHER:

- Only one interval is selected for machine (m, i, j) in time period t:

$$\sum_{z=1}^{Z_m} \alpha_{mijtz} = 1, \qquad \forall m \in \mathbf{M},\ i, j, t \in \mathbf{T} : i \leq t, j \geq t \tag{10}$$

- The coefficients sum to the number of machines (m, i, j) in time period t:

$$\sum_{z=0}^{Z_m} y_{mijtz} = n_{mij}, \qquad \forall m \in \mathbf{M},\ i, j, t \in \mathbf{T} : i \leq t, j \geq t \tag{11}$$

- The α_{mijtz} variables ensure that if interval z is selected, the $y_{mijt(z-1)}$ and y_{mijtz} are the only variables that are non-zero:

$$y_{mijt0} \leq M\alpha_{mijt1}, \qquad \forall m \in \mathbf{M},\ i,j,t \in \mathbf{T} : i \leq t, j \geq t \tag{12}$$

$$\begin{aligned} y_{mijtz} &\leq M\left(\alpha_{mijtz} + \alpha_{mijt(z+1)}\right), \\ &\forall z \in \{1,\ldots,Z_m - 1\}, m \in \mathbf{M},\ i,j,t \in \mathbf{T} : i \leq t, j \geq t \end{aligned} \tag{13}$$

$$y_{mijtZ_m} \leq M\alpha_{mijtZ_m}, \qquad \forall m \in \mathbf{M},\ i,j,t \in \mathbf{T} : i \leq t, j \geq t \tag{14}$$

where M is a sufficiently large number.

2.2 An alternative MILP formulation

Both Forestry and Mining have common operational features and equipment selection criteria. (Burt 2008; Burt et al. 2010)'s work on equipment selection for the mining industry also uses MILP to optimally select equipment over multiple time periods to satisfy production requirements. Model FHER considers purchase and salvage pairs of years, whereas (Burt 2008; Burt et al. 2010)'s model uses a binary matrix to record when a machine was owned. One key difference is that (Burt 2008; Burt et al. 2010)'s work assumes that the intervals of functions f_m and g_m are longer than the number of hours in the working year. This means that at most two intervals can be active in any given year, which simplifies the model considerably. Even so, their approach was only tractable for a horizon of 4 or 5 time periods with a commercial solver.

2.3 A simplified model, FHER-S

We can, alternatively, formulate a problem where machine running costs are approximated by assuming that the machines are run at full capacity. To do this, let

P_{ma}	:	the productivity of machine $m \in \mathbf{M}$ at age a
C^M_{mij}	:	the unit cost of purchasing machine $m \in \mathbf{M}$ at the start of period $i \in \mathbf{T}$, maintaining and operating it between periods i to j (inclusive) and salvaging it at the end of period $j \in \mathbf{T}$
s_{mat}	:	the number of machines $m \in \mathbf{M}$ of age a available for use in time period $t \in \mathbf{T}$.

Here, we wish to maximise

$$\sum_{c\in\mathbf{C}} \sum_{k\in\mathbf{O_c}} R_c x_{kb} - \sum_{m\in\mathbf{M}} \sum_{i,j\in\mathbf{T}} C^M_{mij} n_{mij} \tag{15}$$

such that:

- At most one harvest system for each contract:

$$\sum_{k\in\mathbf{O}_c} y_{kb} \leq 1, \qquad \forall c \in \mathbf{C} \tag{16}$$

- Meet the production targets (for setups in use):

$$\sum_{m \in \mathbf{M}_g} \sum_{0 \le a \le t-1} P_{ma} s_{mat} \ge \sum_{c \in \mathbf{C}} \sum_{k \in \mathbf{O_c}} W_{gkt} y_{kb}, \quad \forall t \in \mathbf{T}, g \in \mathbf{G} \tag{17}$$

- Number of machines:

$$s_{mat} = \sum_{i \in \mathbf{T}: i \le t, t-i=a} \sum_{j \in \mathbf{T}: j \ge t} n_{mij}, \quad \forall t \in \mathbf{T}, m \in \mathbf{M}, a \in \mathbf{N}, a \le T \tag{18}$$

  where $\mathbf{N}$ is the set of machine ages.

Let this formulation be FHER-S. One problem with this model relates to the fact that we include the full costs of running machines even when they are not required to run at capacity.

3 A Prototype Decision Support Tool

The decision support tool is a progression from the widely known *Machine Rate*. A Machine Rate is the cost per hour of owning and operating a piece of equipment, averaged across its life and including the purchase cost minus the salvage value. (Brinker et al. 2002; Murphy and Associates 2008) illustrate such an approach. They are relatively simple, and even though the amount of data required to calculate a Machine Rate is small, it is still the case that many of the inputs are typically rules of thumb. Machine Rate calculations commonly in use can be adapted to generate input for FHER-S with little added effort, with the disclaimer that more empirical data will need to be given for more accurate modelling to take place.

The FHER-S model can be solved quickly for realistically sized problems. As such it is currently the underlying model used by the prototype tool. The FHER model remains computationally challenging, but works as a basis for heuristics to be developed in the future. At least until these heuristics are implemented, the decision support tool has a relatively simple design. It displays an interface to the user, manages the data that is input and calculates the input that the solver requires. The finished prototype will further be required to invoke the solver directly and present the generated schedule back to the interface. The tool is important because these models require a significant amount of data in order to run and managing that data well is essential to these models being usable.

One strength of the tool is that it has its own internal representation of the model that is in use, including the data. This allows it to use different solvers – each solver only requires an interfacing layer to be written rather than a rewrite of the tool. For the purposes of testing it uses the FICO Xpress-Optimizer solver, but has a layer that will allow it to use the open source solvers GLPK or SYMPHONY when it is released to the industry. The tool is written in F# (mostly) and C# (GUI) and and uses the .NET Framework. Figure 1 shows screenshots from the tool's GUI whereby the financial scenario information and the attributes of a Feller Buncher is entered.

The tool is scheduled to be evaluated by contractors by the end of year 2010. Therefore we are not able to report on numerical results or feedback from contractors at this stage.

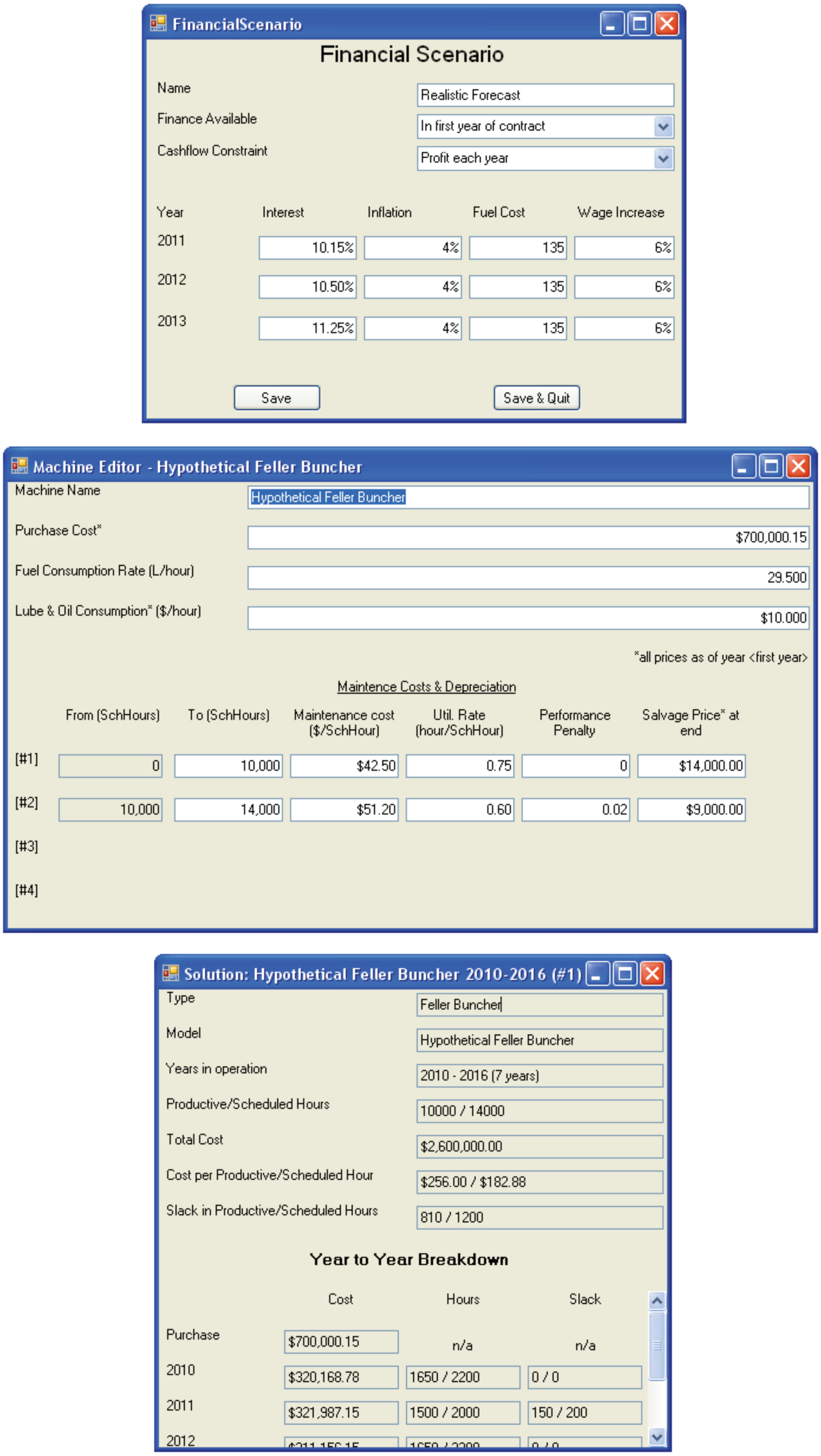


Figure 1: The Prototype Decision Support Tool.

4 Conclusions

We presented a prototype decision support tool for the equipment replacement problem in forestry, together with its underlying optimisation models. Much work remains to be done for effective solutions to this problem to be mature and to be adopted by the industry, but this paper has laid a modelling framework for further work. Particular challenges to be met include:

- Further development and refinement of the decision support tool.
- Productivity estimates and better costing parameters for harvesting machinery.
- An accurate but fast heuristic for the FHER model.
- Communication of results with the industry and researchers in forestry.

Acknowledgments

This project was supported by the CRC for Forestry. We would like to thank David Drane of the Australian Forest Contractors Association (AFCA), for sharing his thoughts on the project and for facilitating meetings with AFCA member contractors who have given their valuable time and input to this project. Thanks also to the CRC for Forestry researchers Dr. Mauricio Acuna and Dr. Mohammad Ghaffariyan.

References

Bilek, E. M. 2007. "ChargeOut! Determining Machine and Capital Equipment Charge-Out Rates Using Discounted Cash-Flow Analysis." Technical Report, United States Department of Agriculture: Forest Productions Laboratory.

———. 2009. "ChargeOut! Discounted Cash Flow Compared with Traditional Machine-Rate Analysis." Technical Report, United States Department of Agriculture: Forest Productions Laboratory.

Brinker, R. W., J. Kinard, B. Rummer, and B. Landford. 2002. "Machine Rates for Selected Harvesting Machines." Technical Report, Auburn University.

Burt, C., L. Caccetta, P. Welfama, and L. Fouche. 2010. "Equipment selection with heterogeneous fleets for multiple-period schedules." *Journal of the Operational Research Society*, pp. 1–12.

Burt, Christina. 2008. "An optimisation approach to materials handling in surface mines." Ph. D. thesis, Department of Mathematics and Statistics, Curtin University of Technology, Perth, Australia.

Murphy, G. E., and Associates. 2008. Australian Logging Productivity and Cost Appraisal Model. Cooperative Research Centre for Forestry, Hobart, Tasmania.

Rogers, J. L., and J. C. Hartman. 2005. "Equipment replacement under continuous and discontinuous technological change." *IMA Journal of Management Mathematics* 16 (1): 23–36.

Waddell, R. 1983. "A Model for Equipment Replacement Decisions and Policies." *Interfaces* 13 (4): 1 – 7.

Maintenance Scheduling for the Hunter Valley Coal Chain

N. Boland, T. Kalinowski, H. Waterer, L. Zheng
School of Mathematical & Physical Sciences
University of Newcastle
Australia
{natashia.boland,thomas.kalinowski,hamish.waterer,lanbo.zheng}@newcastle.edu.au

Abstract

The Hunter Valley Coal Chain (HVCC) comprises mining companies, rail operators, railway track owners and terminal operators, together forming one of the worlds largest coal exporters. Increasing demand calls for highly efficient use of available infrastructure and resources, as well as for well-informed decisions on potential capacity expansions. In particular, the planning processes for different parts of the network have to be integrated, and this leads to complex and very large scale optimization problems.

In the talk we discuss the annual planning of preventive maintenance for railway track and terminal equipment, where the main objective is to maximize the total system capacity. The problem is naturally modeled as a dynamic network flow where the capacities of arcs are temporarily reduced due to maintenance events. We present a MIP formulation for the problem as well as heuristic approaches, and we report on computational experiments using real world data.

Key words: Maintenance scheduling, dynamic flow

The Hunter Valley Coal Chain (HVCC) constituted by 30 mines owned by 15 companies, rail and road providers, as well as port coal services operates the world's largest coal export facility. In 2008, the throughput of HVCC was about 92 million tonnes, or more than 10 per cent of the world's total trade in coal for the year. The coal export operation is responsible for around 15 billion in annual export income for Australia (Boland and Savelsbergh). A main goal of the coal chain management is to align the operations on various parts of the chain efficiently to maximize the daily capacity and to enable a long term capacity expansion.

To optimize the coal chain as a single complex system is a complicated task where a large range of aspects from coal train transportation to terminal machine operation have to be considered and coordinated. Even optimizing a single aspect of the whole system is already a hard problem. In this paper, a scheduling problem for the maintenance of different facilities of the coal chain network is investigated. This study is motivated by the observation that every year, there are series of preventive and corrective maintenance jobs carried on trains, track sections, coal reclaiming or loading machines etc. to ensure that the whole network functions properly. The system capacity is significantly restricted by maintenance that there is about 30% coal reduction due to more than 2,000 maintenance jobs per year. Good alignment of maintenance on different parts of the infrastructure is essential for an efficient use of the system.

We model the coal chain as a network where the arcs represent the assets to be maintained. In a natural way, this leads to the study of a network flow problem where the capacity on an arc drops to zero once it's under maintenance. The objective is to schedule the maintenance jobs for a given time horizon such that the total flow is maximized.

Throughout we use the notation $[k,l] = \{k, k+1, \dots, l\}$ and $[k] = \{1, 2, \dots, k\}$ for $k, l \in \mathbb{Z}$. Let (N, A, s, s') be a network with node set N, arc set A, source s and sink s', and let in addition $u_a \in \mathbb{N}$ for $a \in A$ be capacities. We consider the network over a time horizon $[T] = \{1, 2, \dots, T\}$. A *maintenance job* j is specified by its associated arc $a_j \in A$, its execution time $\tau_j \in \mathbb{N}$, its release date $r_j \in [T]$, and its deadline $d_j \in [T]$. In our model, the execution of a maintenance job starting at time $t \in [r_j, d_j - \tau_j + 1]$ implies that the arc a_j is not available at time t, $t+1$,$\dots$, $t+\tau_j-1]$. We consider the problem to align a given set $\mathcal{J}$ of maintenance jobs in such a way that the total throughput over the interval $[T]$ is maximized. In order to formalize the problem we introduce the following notation.

- For $a \in A$ and $t \in [T]$
  - $\phi_{at} \in \mathbb{R}_+$ is the flow on arc a over time interval t,
  - $x_{at} \in \{0,1\}$ indicates the availability of arc a at time t.
- For $j \in \mathcal{J}$ and $t \in [r_j, d_j - \tau_j + 1]$, $y_{at} \in \{0,1\}$ indicates if job j starts at time t.

Our objective is to maximize the total throughput, i.e.

$$\max\ f(\boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{y}) = \sum_{t=1}^{T} \sum_{v\,:\,a=sv\in A} \phi_{at} \tag{1}$$

subject to the following constraints.

Flow conservation constraints.

$$\sum_{u\,:\,uv\in A} \phi_{at} - \sum_{w\,:\,vw\in A} \phi_{at} = 0 \qquad (v \in N \setminus \{s, s'\},\ t \in [T]), \tag{2}$$

Capacity constraints.

$$\phi_{at} \leqslant u_a x_{at} \qquad (a \in A,\ t \in [T]), \tag{3}$$

Execution constraints.

$$\sum_{t=r_j}^{d_j-\tau_j+1} y_{jt} = 1 \qquad (j \in \mathcal{J}), \tag{4}$$

Outage constraints.

$$x_{at} + \sum_{t'=t-\tau_j+1}^{t} y_{jt'} \leqslant 1 \qquad (a \in A,\ j \in \mathcal{J}_a), \tag{5}$$

where $\mathcal{J}_a = \{j \in \mathcal{J} \ : \ a_j = a\}$ is the set of all jobs on arc a.

We call the problem (1)–(5) *maximum total flow with flexible arc outages* (**MaxTFFAO**). By reduction from 3-partition it is easily seen that the problem **MaxTFFAO** is strongly NP-hard, suggesting that in order to tackle instances of practical relevance efficient heuristics might be needed.

We propose a simple local search heuristics using single job movements. The approach is based on the following observations.

- For a fixed maintenance schedule the evaluation of the maximum total flow is reduced to a sequence of max flow problems in very similar networks.
- Reduced costs can be used to detect arcs on which maintenance causes bottlenecks.
- The available solutions for the maximum flow problems can be used to efficiently evaluate a large number of job movements.

We also present some computational results, indicating that our local search heuristics can perform better than just putting the MIP formulation into a general purpose solver, especially for generating quickly high quality solutions.

References

Boland, N., and M. Savelsbergh. "Optimizing the Hunter Valley coal chain." In *Managing Supply Disruptions*, edited by H. Gurnani, A. Mehrotra, and S. Ray. Springer-Verlag London Ltd. to appear.

Mathematical Programming-based Heuristics for Production Planning

Ross J.W. James
Department of Management
University of Canterbury, New Zealand
ross.james@canterbury.ac.nz

Bernardo Almada-Lobo
Faculdade de Engenharia
da Universidade do Porto, Portugal
almada.lobo@fe.up.pt

Abstract

We develop construction and improvement heuristics to solve the single and parallel-machine capacitated-lotsizing and scheduling problem with sequence dependent setup times and costs. We compare the performance of these heuristics to metaheuristics and other MIP-based heuristics that have been proposed in the literature as well as a state-of-the-art commercial solver. Computational experiments show the effectiveness and efficiency of the proposed approach in solving medium size problems.

Key words: CLSD, Mixed Integer Programming, relax-and-fix, heuristics

1 Introduction

The aim of this paper is to develop a range of mathematical programming-based heuristics to solve complex lotsizing and scheduling problems. The techniques developed here could be easily generalised to other variants of lotsizing and scheduling problems and even other problems outside this domain.

Our motivation, ultimately, is to solve parallel-machine capacitated lotsizing and scheduling problems with sequence dependent setup times and costs (CLSD-PM). This single-stage problem is an extension of the pure capacitated lotsizing problem (CLSP – Bitran and Yanasse (1982)) and of the CLSD proposed by Haase (1996), which only addresses one machine and no setup times, as we also need to consider an allocation dimension with multiple parallel machines and a sequencing dimension with sequence-dependent setups.

CLSD-PM is considered to be a big-bucket problem, as several products/setups may be produced/performed per period. Despite its applicability, there is a lack of research on these problems and correlated variants, mainly due to its inherent complexity (Clark, Almada-Lobo, and Almeder (2010)).

We consider N products to be manufactured on M different capacitated machines over a discrete planning horizon of T periods. Due to the sequence-dependency of setups in a product changeover, lot sizing and sequencing are simultaneously tackled. The objective is to find a strategy that satisfies demands without backlogging and minimizes both setup and holding costs. The computational complexity of the resulting mixed integer programming (MIP) model makes the use of efficient heuristic strategies mandatory for solving large real-world instances. State-of-the-art optimisation engines either fail to generate feasible solutions to this problem or take a prohibitively large amount of computation time, even for the single-machine setting (see Almada-Lobo and James (2010)).

We focus on new mathematical programming-based heuristics that are flexible and could easily be adapted to cope with model extensions or to address different optimisation problems that arise in practice, as they are problem and constraint independent.

2 CLSD-PM Model Formulation

In this paper we study the CLSD-PM of which the single machine problem (CLSD) is a special case. We consider a planning interval with $t = 1, \ldots, T$ periods and $i, j = 1, \ldots, N$ products processed on $m = 1, \ldots, M$ machines. We use the notation $[K]$ to refer to the set $\{1, 2, \ldots, K\}$. As usual, d_{it} denotes the demand of product i in period t, s_{mij} and c_{mij} the time and cost incurred when a setup occurs from product i to product j on machine m respectively, h_i the cost of carrying one unit of inventory of product i from one period to the next, p_{mi} the processing time of one unit of product i on m and C_{mt} the capacity of machine m available in period t. In addition, G_{mit} is an upper bound on the production quantity of product i in period t on machine m. A_{mi} indicates which machines are able to produce which products. A_{mi} is set to one if machine m is able to produce product i, or zero otherwise. The decision variables used are: X_{mit} is the quantity of product i produced in period t on machine m, I_{it} the inventory level of product i at the end of period t and V_{mit} an auxiliary variable that assigns product i on machine m in period t. The larger V_{mit} is, the later the product i is scheduled in period t on machine m, assuring that each machine is only set up for one product at any given time. In addition, the following 0/1 decision variables are defined: T_{mijt} equals one if a setup occurs from product i to product j on machine m in period t, and Y_{mit} equals one if the machine m is set up for product i at the beginning of period t. The CLSD-PM model below is a generalization of the CLSP (single-machine) introduced in Almada-Lobo et al. (2007):

$$\min \sum_m \sum_i \sum_j \sum_t c_{mij} \cdot T_{mijt} + \sum_i \sum_t h_i \cdot I_{it} \tag{1}$$

$$I_{i(t-1)} + \sum_m X_{mit} - d_{it} = I_{it} \qquad \forall i, \forall t \tag{2}$$

$$I_{i0} = 0 \qquad \forall i \tag{3}$$

$$\sum_i p_{mi} \cdot X_{mit} + \sum_i \sum_j s_{mij} \cdot T_{mijt} \leq C_{mt} \qquad \forall m, \forall t \tag{4}$$

$$G_{mit} \cdot (\sum_j T_{mjit} + Y_{mit}) \geq X_{mit} \qquad \forall m, \forall i, \forall t \tag{5}$$

$$Y_{mi(t+1)} + \sum_j T_{mijt} = Y_{mit} + \sum_j T_{mjit} \forall m, \forall i, \forall t \tag{6}$$

$$\sum_i Y_{mit} = 1 \qquad \forall m, \forall t \tag{7}$$

$$V_{mit} + N \cdot T_{mijt} - (N-1) - N \cdot Y_{mit} \leq V_{mjt} \qquad \begin{array}{l}\forall m, \forall i, \forall t \\ \forall j \in [N] \setminus \{i\}\end{array} \tag{8}$$

$$\sum_t X_{mit} \leq G_{mit} A_{mi} \qquad \forall m, \forall i \tag{9}$$

$$(X_{mit}, I_{it}) \geq 0, (T_{mijt}, Y_{mit}) \in \{0,1\}, X_{mit} \in \mathbb{Z}, V_{mit} \in \mathbb{R} \tag{10}$$

Objective function (1) minimizes the sum of setup and inventory holding costs. Constraints (2) balance production and inventories with demand. Constraints (3) set the initial inventory levels. Constraints (4) ensure that production and setups do not exceed the available capacity. The setup forcing constraints are provided by (5). Constraints (6) keep track of the setup carryover information, whilst constraints (7) ensure that each machine is set up for one product at the beginning of each time period. Note that requirements (6) link two consecutive periods. Disconnected subtours are eliminated by constraints (8). These constraints apply whenever one subtour occurs in a period, forcing the respective machine to be set up at the beginning of that period to one of the products that are part of the subtour. Constraints (9) ensure production of a product can only occur on machines that are able to produce that product. Finally, there are the non-negativity and integrality constraints (10).

3 Decomposition Scheme

The techniques we use to solve the CLSD-PM problem are based on decomposing the original problem into subsets that can be solved more easily by a MIP in an iterative fashion. As the number of integer variables in each subproblem are significant smaller than those of the original problem, the solution times to solve each one to optimality is very small. In order to do this we assume we already have a partial or feasible solution, whose decision variables T, X, Y, and I have the values T', X', Y' and I', respectively, obtained in previous iterations. In each iteration, the set of decision variables to be released and the set of those to be frozen (fixed) need to be given. We formulate a subMIP that will solve specific combinations of periods, products and machines, while also bringing into the solution elements of the problem that have been determined at earlier stages. Let β_t denote a parameter that is set to 1 if the period t is to be optimized (i.e. binary variables related to period t are released), $\psi_i = 1$ to represent the products to be optimized and $\delta_m = 1$ to represent machines that are to be optimized. The periods, products and machines that are to remain the same as our incumbent solution have $\beta_t = 0$, $\psi_i = 0$ and $\delta_m = 0$, respectively. We refer to this as the $subMIP\,(T', X', Y', \beta_t, \psi_i, \delta_m)$ model which is formulated as follows:

$$\min \sum_{m}\sum_{i}\sum_{j}\sum_{t} c_{ij}\cdot T_{ijt} + \sum_{i}\sum_{t} h_i \cdot I_{it}$$
$$(2)-(9)$$

$$T_{mijt}(1-\beta_t\psi_i\psi_j\delta_m) = T'_{mijt}(1-\beta_t\psi_i\psi_j\delta_m) \forall m, \forall i, \forall j, \forall t \tag{11}$$
$$X_{mit}(1-\beta_t\psi_i\delta_m) = X'_{mit}(1-\beta_t\psi_i\delta_m) \quad \forall m, \forall i, \forall t \tag{12}$$
$$Y_{mit}(1-\beta_{t-1}\psi_i\delta_m) = Y'_{mit}(1-\beta_{t-1}\psi_i\delta_m) \ \forall m, \forall i, \forall t \tag{13}$$
$$I_{it}(1-\beta_{t-1}\psi_i) = I'_{it}(1-\beta_{t-1}\psi_i) \qquad \forall i, \forall t \tag{14}$$

$$(X_{mit}, I_{it}) \geq 0, (T_{mijt}, Y_{mit}) \in \{0,1\}, X_{mit} \in \mathbb{Z}, V_{mit} \in \mathbb{R}$$

Constraints (11)-(13) define the binary and integer decision variables for periods that have been solved previously, while constraints (14) the inventory which will be naturally integer due to production and demand being integer.

4 MIP-Based Construction Heuristics

4.1 Myopic Fix: Period construction heuristic

The principle of Myopic Fix period construction heuristic is to construct a solution to the problem instance by breaking it down into a given number of period blocks and solving them separately using a MIP solver. The larger the number of periods, the more CPU time it will take to create a solution and also the higher the quality of the solution. The blocks of periods are solved starting from the earliest time period and working through to the latest time period. We assume that the periods where $\beta_t = 1$ are contiguous and that all periods before the first period where $\beta_t = 1$ have been solved. β_t equals one for every t in $[k..min(T, k+w-1)]$. We set $\psi_i = 1 \forall i$ and $\delta_m = 1 \forall m$.

4.2 Relax-and-Fix

The relax-and-fix (RF) framework decomposes a large-scale MIP into a number of smaller partially relaxed MIP subproblems, that are solved in sequence. Due to the structure of the original MIP, we rely on a time-stage partition, which is a rolling horizon approach. The planning horizon is typically partitioned into non-overlapping intervals. A shift forward strategy solves a sequence of sub-MIPs (one per time interval), each dealing with a sub-set of the integer variable set of the overall problem. The other sub-sets are either relaxed or removed depending on the simplification strategy. As this heuristic progresses, the integer variables (and, depending on the freezing strategy, also the continuous variables) are permanently fixed to their current values. The schedule is completed at the last iteration.

The RF can be parameterized in many ways. Our design is based on Merce and Fontan (2003) and Absi and Kedad-Sidhoum (2007). Let γ be the number of overlapping periods and w the width of each interval. It is implied here that both parameters are constant throughout the algorithm. Note that γ allows us to smooth the heuristic solution by creating some overlap between successive planning intervals (Pochet and Wolsey (2006)). In addition, t_k^i and t_k^f denote the initial and final periods of the interval at step k. At each iteration, k, of the heuristic, the sub-set of variables T and Y up to period t_{k-1}^i are fixed, the sub-set from period t_k^i up to t_k^f are restricted to be integer, and other sub-sets being relaxed. Note that $w - \gamma$ defines the length of the anticipation horizon at each iteration and, consequently, the number of iterations of the heuristic. Recall that our formulation keeps track of the setup carryover information, knowing the product that the machine is ready to process at the beginning of each time period. The motivation for using $w > 1$ and $\gamma > 0$ (with $\gamma < w$) comes from the setup carryover information. With a $\gamma = 0$, blocks do not overlap and therefore the product the machine is set up for at the beginning of the block is fixed at the previous iteration. This therefore affects the production sequence of the current block to be optimised.

5 MIP-Based Improvement Heuristic

The principle behind the improvement heuristic is to reoptimise iteratively a contiguous set of periods and reoptimise the jobs in these periods, fixing all the variables of other periods. The reoptimisation is again done via a subMIP similar to that outine previously except that constraints (14) is removed. Each subproblem is derived from the best solution found so far, fixing the binary variables of the periods where $\beta_t = 0$ at the optimal values from the incumbent solution, thus dealing with a reduced number of “free binary” variables (related to periods where $\beta_t = 1$).

We do not relax the integrality restriction for any binary variable. In order to improve the speed of the heuristic, the improvement heuristic is designed to intelligently select the periods. Initially the probability of selecting the set of periods starting at any given periods will be the same. However as periods are selected, the probability of that period being selected is reduced. Two factors determine a period’s probability of selection, the frequency of the period being selected and the recency of the period being selected. The more frequently a period has been used (given by the number of runs the respective period was selected), and the more recently it has been used (given by the number of iterations passed since the last selection of that period), the lower the probability of selection. The probability of selection of a period is proportional to a weighing average of these two factors. The search also keeps track of which period combinations have been solved without any improvement in the solution quality. A period will not be selected if it already has been used and the current objective value was obtained (providing a form of short-memory which speeds up the solution procedure). If all period combinations have been tried and there has been no improvement in the solution, then we are in a local minima and the heuristic will terminate. The heuristic will also terminate if the maximum CPU time is exceeded. We refer to the heuristic using the above formulation as period heuristic (PH).

One of the main causes of local minima in the above formulation is that the quantity produced in any one period is only optimised to the periods under consideration. The formulation can quite easily be generalized by removing constraints (12) to allow for the quantities produced in each period (X_{mit}), and subsequently the inventory held in each period (I_{it}), to be optimised for all periods, while the sequence is optimised only for selected periods, hence overcoming this source of local minima. The disadvantage of this approach will be that the solution time to solve an iteration will increase. Nevertheless, it is well known that given a fixed set of the setup variables, the remaining linear subproblem can be solved optimally through a network flow reformulation that can be solved in polynomial time, thus each node of the branch-and-cut tree can be evaluated quickly. The cost of this extra computational time compared to the extra quality achieved will be assessed in Section 6. We refer to the heuristic using this formulation as X (production quantity) and P (period) heuristic (XPH). The pseudo code for PH and XPH is identical, except for the MIP model used.

6 Computational Experiments

6.1 Single Machine Experiments

For the single-machine problem we test different algorithms on random data sets generated using the approach of Almada-Lobo et al. (2007). Elements of the problem were generated from a uniform distribution and then rounded to the nearest integer, or calculated from elements that were generated this way. The ranges used for the elements are: Setup Times between 5 and 10 time units; Setup Costs are proportional to the setup time by a specified parameter (Cost of Setup per unit of time, θ); Holding costs between 2 and 9 penalty units per period; Demand between 40 and 59 per period; Period Capacity is proportional to the total demand in that period as defined by a parameter (Capacity Utilization per period, Cut): $C_t = \sum_i d_{it}/Cut$; Processing time for one unit = one unit of time.

Twenty four different problems types were created from the combinations of the following problem parameters: Number of Products $N \in 15, 25$; Number of Periods $T \in 5, 10, 15$; Capacity Utilization per period $Cut \in 0.6, 0.8$; Cost of Setup per unit of time $\theta \in 50, 100$

In each case 10 different instances were generated, creating a total of 240 problem instances to be solved. Each type of instance can be characterized by the quadruple N, T, Cut and θ.

The number of periods to solve at a time to solve for the Myopic-Fix MIP construction heuristic was tested with 2 and 3 period solutions. The Relax-and-Fix heuristic requires two parameters, namely the width of the interval w and the number of overlapping periods γ. Preliminary empirical analysis pointed out the following choice of values: $w = 2$ and $\gamma = 1$. For each subMIP$_k\left(t_k^i, t_k^f\right)$, we have set a time limit for each of $3600/K$ seconds, where $K = \lceil (T-w)/(w-\gamma) \rceil + 1$ denotes the total number of iterations, and a relative MIP gap tolerance of 0.5%. Besides these two MIP approaches, we implement the constructive heuristic of Almada-Lobo et al. (2007) to generate an initial solution. It contains several forward and backward steps that place feasibility over optimality to find feasible solutions efficiently. The aim is to compare afterwards the effectiveness of the improvement heuristic given different initial solutions.

To evaluate the quality of these heuristics, we use the lower bound generation procedure described in Almada-Lobo et al. (2007), which essentially strengthens the formulation with the (l, S) cuts that are introduced with the separation algorithm of Barany, Vanroy, and Wolsey (1984). These computational experiments were performed on a computer with a Pentium T7700 CPU running at 2.4 GHz with 2GB of random access memory. On this system, Parallel CPLEX 11.1 was used as the mixed integer programming solver, while the algorithms were coded in Visual C++ .NET 2005.

The problems solved here are a subset of the problems that were solved in Almada-Lobo and James (2010) using Tabu Search and Variable Neighbourhood Search. Almada-Lobo and James (2010) used similar hardware to what we used here, and therefore we can compare the quality of the solutions between the different techniques and the CPU times required to obtain these solutions.

Tables 1 and 2 present the computational results of the different algorithms proposed in this paper. The algorithms presented in these tables are as follows:

- PHOC/XPHOC: five-step construction heuristic of Almada-Lobo et al. (2007) then a two period improvement heuristic or two period and quantity improvement heuristic respectively;
- PHRF/XPHRF: relax-and-fix construction heuristic then a two period improvement heuristic or a two period and quantity improvement heuristic respectively;
- PH2/XPH2: two period construction heuristic then a two period improvement heuristic or a two period and quantity improvement heuristic respectively;
- PH3/XPH3: three period construction heuristic then a two period improvement heuristic or a two period and quantity improvement heuristic respectively;
- TS/VNS : the Tabu search or Variable Neighbourhood Search of Almada-Lobo and James (2010) respectively;
- FOHRF: relax-and-fix construction heuristic then a fix-and-optimise improvement heuristic with a MIP solution tolerance of 0.005;
- CPLEX: branch-and-cut performed by Parallel CPLEX 11.1 on the original model;

Of the two different types of LP heuristics, not surprisingly XPHOC, XPHRF, XPH2 and XPH3 dominate PHOC, PHRF, PH2 and PH3 respectively. XPHRF is clearly the best heuristic in terms of solution quality, being on average 1.40% off the lower bound, compared to 1.50%, 1.66%, 3.55%, 3.59% for XPH2, XPH3, PH2 and PH3 respectively. More surprisingly are the solution times for XPH2, which average 88.4 seconds for the test data. This was considerably faster than XPH3 and PH3, at 151.1 and 107.1 seconds respectively, but slower than PH2 at 57.8 seconds, as expected.

It appears that regardless of the initial solution, the extra quality of the final solutions obtained from the XPH heuristic are worth the extra computational time involved. Furthermore, the results show that the XPH improvement heuristic is not too-dependent upon the starting solution, as proven by the small differences on the gaps reported by XPHOC, XPHRF, XPH2 and XPH3.

The results clearly show that all LP-based heuristics outperformed the TS and VNS metaheuristics in almost every problem type. The only exception being where there are fifteen product types, five periods, $Cut = 0.6$ and $\theta = 100$, where VNS outperforms the PH2 and PH3 heuristics at the expense of significantly larger computational time. In this case, XPHRF, XPH2 and XPH3 performed better than VNS. In all other cases all the LP heuristics not only obtain higher quality solutions but also do so in less time. Given the same initial solution, it is clear that XPH (XPHOC) produces significantly better results with shorter CPU times than VNS or TS, being 1.59% off the lower bound (needing only 126.4 seconds), compared to 4.75% (2310.3 seconds) and 5.98% (727.7 seconds) for the other two, respectively. There are no single test instances where FOHRF dominates XPHRF. If we look at Table 1 we see that there are no problem classes where FOHRF has a better average than XPHRF. In terms of solution quality CPLEX, not surprisingly, provided the best solution when it could find them. Unfortunately CPLEX could only find solutions to the smaller problem instances and could not find even a feasible solution to the larger problems within the one hour time limit.

Table 1: Average % Deviation from the Lower Bound

Problem Type	Average % Deviation from the Lower Bound											
N-T-Cut-θ	PHOC	XPHOC	PH2	XPH2	PH3	XPH3	PHRF	XPHRF	TS	VNS	FOHRF	CPLEX
15-5-0.6-50	1.13%	0.57%	1.43%	0.35%	0.64%	0.44%	0.47%	0.46%	2.48%	1.05%	0.53%	**0.22%**
15-5-0.6-100	4.57%	2.39%	7.25%	1.97%	5.28%	2.80%	1.86%	1.82%	6.40%	3.85%	1.93%	**1.37%**
15-5-0.8-50	1.31%	0.73%	1.47%	0.66%	1.04%	0.76%	0.72%	0.64%	3.11%	1.42%	0.83%	**0.40%**
15-5-0.8-100	4.27%	2.33%	4.59%	2.45%	3.75%	2.50%	2.31%	2.08%	6.56%	3.83%	2.36%	**1.73%**
15-10-0.6-50	1.92%	0.67%	0.56%	0.46%	1.92%	0.76%	0.86%	0.64%	3.27%	2.33%	1.11%	**0.40%**
15-10-0.6-100	6.09%	3.24%	5.65%	3.47%	9.01%	3.65%	4.24%	2.95%	7.52%	5.85%	4.16%	bf2.70%
15-10-0.8-50	1.71%	0.84%	1.24%	0.92%	2.07%	0.90%	1.11%	0.89%	3.75%	2.39%	1.59%	bf0.60%
15-10-0.8-100	5.93%	3.42%	6.40%	3.44%	6.36%	3.31%	3.52%	**2.83%**	9.03%	6.53%	3.53%	3.40%
15-15-0.6-50	1.85%	0.77%	1.20%	0.73%	1.53%	1.02%	0.90%	**0.58%**	3.82%	3.26%	1.46%	0.64%
15-15-0.6-100	6.78%	3.69%	7.59%	**3.52%**	7.11%	3.76%	5.45%	3.54%	9.05%	7.78%	5.04%	-
15-15-0.8-50	1.65%	0.90%	1.61%	0.94%	1.85%	1.01%	1.09%	**0.84%**	4.10%	3.34%	1.81%	-
15-15-0.8-100	6.64%	3.80%	6.94%	3.60%	6.27%	3.73%	4.68%	**3.46%**	10.17%	8.16%	5.12%	-
25-5-0.6-50	1.43%	0.28%	1.59%	0.22%	0.55%	0.27%	0.24%	0.17%	3.38%	2.24%	0.43%	**0.09%**
25-5-0.6-100	4.78%	1.23%	6.97%	1.31%	4.50%	1.43%	0.94%	0.85%	5.82%	4.66%	1.03%	**0.66%**
25-5-0.8-50	1.24%	0.52%	1.65%	0.54%	0.67%	0.52%	0.47%	0.43%	3.81%	2.51%	0.68%	**0.27%**
25-5-0.8-100	3.06%	1.16%	3.23%	1.08%	2.46%	1.11%	1.08%	0.95%	6.56%	4.02%	1.32%	**0.85%**
25-10-0.6-50	1.89%	0.65%	0.52%	**0.38%**	2.01%	0.73%	0.64%	0.50%	4.22%	4.28%	1.13%	-
25-10-0.6-100	5.53%	2.23%	4.63%	**1.90%**	8.46%	2.04%	3.00%	2.03%	7.47%	7.10%	3.34%	-
25-10-0.8-50	1.56%	0.62%	1.34%	**0.49%**	1.67%	0.68%	0.66%	0.53%	4.74%	4.15%	1.15%	-
25-10-0.8-100	4.69%	2.06%	4.99%	1.93%	4.82%	2.10%	2.58%	**1.78%**	8.82%	6.72%	2.93%	-
25-15-0.6-50	2.09%	0.60%	1.18%	0.55%	1.50%	0.95%	0.81%	**0.54%**	4.74%	4.88%	1.59%	-
25-15-0.6-100	5.98%	2.46%	6.38%	2.19%	6.46%	2.50%	3.68%	**2.17%**	9.01%	9.62%	4.25%	-
25-15-0.8-50	1.69%	0.64%	1.54%	0.64%	1.53%	0.67%	0.94%	**0.62%**	5.28%	5.11%	1.56%	-
25-15-0.8-100	5.18%	2.38%	5.33%	2.26%	4.70%	2.30%	3.18%	**2.22%**	10.48%	8.96%	3.68%	-
Average	3.46%	1.59%	3.55%	1.50%	3.59%	1.66%	1.89%	1.40%	5.98%	4.75%	2.19%	-

Note: A '-' indicates no solution was found in the one hour time limit

6.2 Multiple Machines

The data for multi-machine problems are generated in a similar manner to the problems for single machines, however two extra parameters are required to generate the assignment matrix data A_{mi}, that is a 0/1 parameter that allows the allocation of products to machines. These parameters are the probability of the extra machines being able to produce the same product, $MProb$, and the maximum percentage difference in the number of jobs between machines, $MBal$. When $MProb$ equals to one, each machine can process every product.

As a base model we used a mid-sized problem with 15 products, 10 periods and 80% capacity utilization. We then tried different perturbations of M, N, T, Cut, θ, $MProb$ and $MBal$. For each combination 10 instances were generated. The average results are presented in Table 3. Note that all problems have a 3600 second time limit. The Fixed and Optimise Heuristics have a $3600/K$ second sub-MIP time limit, while the INSRF heuristic has a 600 second XPH sub-MIP time limit.

Table 2: Average Computational Times

Problem Type	Time in Seconds											
N-T-Cut-θ	PHOC	XPHOC	PH2	XPH2	PH3	XPH3	PHRF	XPHRF	TS	VNS	FOHRF	CPLEX
15-5-0.6-50	2.7	2.9	1.3	3.0	3.1	4.1	6.7	5.3	39.0	600.2	5.0	23.1
15-5-0.6-100	4.9	8.7	2.9	6.9	5.8	12.1	19.6	14.9	30.7	493.3	14.8	56.9
15-5-0.8-50	6.6	4.8	3.8	3.9	6.2	7.3	10.0	8.3	13.3	716.5	7.5	20.3
15-5-0.8-100	8.2	22.4	8.0	10.8	36.9	45.4	36.5	28.7	9.7	589.2	27.3	851.8
15-10-0.6-50	8.0	10.7	3.3	4.5	9.9	20.7	20.0	18.5	767.8	2755.3	14.9	2539.2
15-10-0.6-100	28.4	19.2	9.5	21.4	13.4	37.5	54.2	52.1	84.0	2730.7	42.9	3600.0
15-10-0.8-50	15.0	16.0	10.1	14.5	14.4	29.2	28.6	24.8	398.2	2694.7	20.3	2444.7
15-10-0.8-100	27.6	33.0	25.8	37.0	81.3	119.0	122.1	94.2	53.4	2736.7	87.6	3600.0
15-15-0.6-50	16.4	19.5	5.4	16.0	15.2	19.0	35.6	34.3	1344.6	2822.1	25.4	3275.9
15-15-0.6-100	16.8	31.7	17.2	35.5	22.3	48.7	105.7	102.1	598.9	2810.0	86.1	3600.5
15-15-0.8-50	24.2	30.5	17.8	24.9	30.7	48.9	48.9	44.8	428.4	2664.0	35.5	3300.7
15-15-0.8-100	47.1	57.7	44.4	64.8	148.9	208.7	222.9	189.0	239.5	2791.8	170.9	3600.1
25-5-0.6-50	29.9	36.2	7.7	20.9	15.7	24.7	41.9	34.8	566.7	1894.4	30.2	94.0
25-5-0.6-100	55.2	79.2	18.9	43.9	37.1	91.0	118.8	95.7	127.6	1646.8	79.4	1737.9
25-5-0.8-50	36.5	54.8	27.0	38.7	47.7	68.3	102.4	79.3	101.3	1996.7	75.0	224.6
25-5-0.8-100	108.5	176.3	90.6	208.0	130.8	229.3	280.4	283.5	193.2	1755.7	209.3	3369.6
25-10-0.6-50	34.3	103.0	16.5	41.1	45.1	72.2	153.0	137.3	1887.2	3099.7	101.3	3600.1
25-10-0.6-100	74.3	242.4	51.4	113.9	58.2	169.6	275.6	274.1	789.2	2908.1	210.8	3600.9
25-10-0.8-50	148.7	176.5	95.6	126.5	131.6	197.7	225.3	195.0	1377.1	3086.9	152.0	3602.1
25-10-0.8-100	608.7	475.1	233.2	260.9	453.0	668.0	496.6	455.6	895.8	2943.1	358.7	3600.0
25-15-0.6-50	59.7	161.0	33.4	109.8	76.6	124.8	385.4	347.0	2785.8	3053.0	192.1	3600.8
25-15-0.6-100	125.0	337.6	92.1	240.6	110.6	273.8	734.6	822.9	1188.3	2870.3	480.6	3601.1
25-15-0.8-50	179.8	336.8	185.1	218.5	269.4	284.7	407.4	359.6	2381.5	2822.9	261.4	3607.4
25-15-0.8-100	518.8	597.3	385.6	456.6	806.8	821.8	891.2	967.4	1163.3	2964.0	712.7	3600.3
Average	91.1	126.4	57.8	88.4	107.1	151.1	201.0	194.5	727.7	2310.3	141.7	2548.0

CPLEX could not find solutions to these problems within the one hour time limit, however it was used to calculate a lower bound that was based on a plant-location reformulation of our model which is known to produce tighter bounds.

From Table 3 we can see that XPHRF on average is marginally better than the FOHRF. Adding an extra machine clearly makes the problem much more difficult to solve as can be seen by the increasing average deviations between those problems with 2 machines and those with 3 machines. As the sub-MIPs for these techniques have been time limited, the difficulty of the problem will be reflected in a loss of quality rather than an increase in computational time. In several cases we can see this occurring with the problems with 3 machines actually taking less computational times than those with 2 machines. So although we can solve larger problems with this technique, the importance of providing an appropriate amount of time for both the overall search and the sub-MIPs is clearly a vital aspect which dictates the performance of these heuristics. The larger problems, in general, will require much more sub-MIP and overall computational time to obtain similar levels of performance to problems with fewer machines. The parameter $MProb$ (related to the flexibility of the machinery) has an impact on the performance of the algorithms. As $MProb$ increases, the problem gets more difficult as many alternative optimal solutions can be obtained. $MBal$ seems to have no impact on the behaviour of the solution procedures.

7 Conclusions

In this paper we have presented heuristics based on MIP to solve the CLSD and CLSD-PM. The first two heuristics are construction heuristics which solve a MIP optimally scheduling a specified number of periods at a time. The third is a stochastic improvement heuristic which reoptimises the sequence and production quantities

Table 3: Multi-Machine Results - Summary

Problem Type	Ave. % Dev. from Lower Bound		Ave. Time in Seconds	
M-N-T-*Cut*-θ-*MProb*-*MBal*	XPHRF	FOHRF	XPHRF	FOHRF
2-15-5-0.8-50-80-20	1.49%	2.05%	323.7	50.6
2-15-10-0.8-50-80-20	2.60%	3.60%	527.1	110.3
2-15-10-0.8-100-80-20	7.10%	8.31%	1371.2	345.0
2-20-10-0.8-100-80-20	5.90%	6.17%	3756.1	994.8
2-15-10-0.8-100-60-20	6.02%	7.19%	322.2	139.5
2-15-10-0.8-100-80-10	6.98%	8.46%	1254.5	370.9
3-15-10-0.8-50-80-20	10.78%	7.30%	7623.0	682.8
3-15-5-0.8-50-80-20	4.37%	4.89%	3023.8	173.8
3-15-10-0.8-100-60-20	12.86%	14.73%	2910.9	711.2
3-15-10-0.6-100-60-20	8.00%	10.86%	960.3	186.5
Average	6.61%	7.36%	2207.3	376.5

for given periods, and the fourth is a randomized improvement heuristic that reoptimises the production quantities for all periods, while reoptimising the sequence for only a given set of periods. The computational experiments found that the improvement heuristics proposed here, preceded by the construction heuristics, outperform other meta-heuristic approaches proposed in the literature in terms of quality and often in time, as well as other MIP-based heuristics proposed in the literature. These heuristics also clearly outperformed a commercial MIP solver for the larger instances.

References

Absi, N., and S. Kedad-Sidhoum. 2007. "MIP-based heuristics for multi-item capacitated lot-sizing problem with setup times and shortage costs." *Rairo-Operations Research* 41 (2): 171–192.

Almada-Lobo, B., and R.J.W. James. 2010. "Neighbourhood Search Metaheuristics for Capacitated Lotsizing with Sequence-dependent Setups." *International Journal of Production Research* 48 (3): 861–878.

Almada-Lobo, B., D. Klabjan, M. A. Carravilla, and J. F. Oliveira. 2007. "Single Machine Multi-product Capacitated Lotsizing with Sequence-dependent setups." *International Journal of Production Research* 45 (20): 4873–4894.

Barany, I., T. J. Vanroy, and L. A. Wolsey. 1984. "Strong Formulations for Multi-Item Capacitated Lot Sizing." *Management Science* 30 (10): 1255–1261.

Bitran, G.R., and H.H. Yanasse. 1982. "Computational complexity of the capacitated lot size problem." *Management Science* 28 (10): 1174–1186.

Clark, A. R., B. Almada-Lobo, and C. Almeder. 2010. "Editorial on lotsizing and scheduling: industrial extensions and research opportunities." *Accepted for publication in International Journal of Production Research.*

Haase, K. 1996. "Capacitated lot-sizing with sequence dependent setup costs." *Operations Research Spektrum* 18:51–59.

Merce, C., and G. Fontan. 2003. "MIP-based heuristics for capacitated lotsizing problems." *International Journal of Production Economics* 85 (1): 97–111.

Pochet, Y., and L. A. Wolsey. 2006. *Production Planning by Mixed Integer Programming.* Springer Series in Operations Research and Financial Engineering. New York: Springer.

Scheduling Super Rugby

Mark Johnston
School of Mathematics, Statistics and Operations Research
Victoria University of Wellington, New Zealand
mark.johnston@msor.vuw.ac.nz

Mike Wright
Department of Management Science
University of Lancaster, United Kingdom
m.wright@lancaster.ac.uk

Abstract

We develop scheduling models of Super Rugby, the existing Super 14 (2006–2010) and the proposed Super 15 (from 2011), and the revised national provincial ITM Cup competition (from 2011). Developing schedules for these competitions involves a large number of competition design decisions and scheduling compromises between team welfare, travel, television, and revenue management. We show that Super 15 addresses some of the complications that arose in scheduling Super 14. The 2011 ITM cup features a very tight scheduling window due to the Rugby World Cup, with 10 matches per team over a 7 week period. The schedules developed show that it is possible to accommodate most of the (assumed) preferences of teams and organisers.

Key words: Scheduling, rugby.

1 Introduction

Operations Research is playing an increasing role in sport, both in terms of strategy and scheduling (Wright 2009) (Kendall et al. 2010). Scheduling of sports competitions is a difficult combinatorial optimisation problem with a large number of conflicting objectives involving travel, team welfare, television and revenue management. It is very difficult to please everyone involved, or perhaps anyone (Cleaveland Live 2009) (Herald Scotland 2010).

Problem description. In designing a *new* national or international sports competition, there are a number of *design decisions* that can only be addressed by proposing candidate schedules. In this paper we look at two examples of new and relatively small competition formats planned to begin in 2011: the Super 15 and the ITM Cup. The Super 15 is a new competition in 2011, an expansion of previous Super rugby competitions to 15 teams, but with a significantly different structure to previous years. The ITM Cup 2011 edition is a new competition structure featuring a number of significant one-off scheduling difficulties mostly due to the 2011 Rugby World Cup which immediately follows.

Research goals of this paper. Explore various scenarios under which the competitions could take place, and compare the difficulty or resulting complexity of the schedules they imply. Is it possible to quickly develop "good" schedules (with respect to given requirements and preferences) in order to address competition design decisions?

Outline of this paper. Section 2 provides an analysis of the existing Super 14 (2006–2011), Section 3 discusses modelling of the new Super 15, and Section 4 discusses modelling of the new ITM Cup structure. Finally, Section 5 offers some brief conclusions.

2 Super 14 (2006–2010)

Professional rugby in the southern hemisphere began with the Super 12 competition (1996–2005) featuring teams from Australia, New Zealand and South Africa. This was expanded to 14 teams for the Super 14 (2006–2010) (Wikipedia 2010b).

Competition structure. The Super 14 rugby competition involves 5 teams from New Zealand (Blues, Chiefs, Crusaders, Highlanders and Hurricanes), 4 teams from Australia (Brumbies, Force, Reds and Waratahs) and 5 teams from South Africa (Bulls, Cheetahs, Lions/Cats, Sharks and Stormers). The Cats featured in the Super 14 in 2006 and were replaced by the Lions from 2007. The teams play a single-round-robin tournament (each team plays all other teams exactly once, either at home or away), together with one bye per team, for a total of 14 rounds. Matches between pairs of teams alternate venue each year, e.g., Blues vs Hurricanes was played at the Blues home (2006, 2008 and 2010) and at the Hurricanes home (2007 and 2009). All teams play either six or seven home matches each year, alternating year on year. At most two teams have a bye in any round. The single-round-robin is followed by two semifinals (1 v 4 and 2 v 3) and a final. The competition features some extremely large travel distances.

Actual schedules. The actual 2006 and 2007 schedules played are given in Tables 1 and 2. Negative indicates an away match and upper case highlights away matches in another country. The 2008 schedule was the 2006 schedule played in reverse order of rounds, with the additional change that the matches and byes in rounds 10 and 11 (of the 2006 schedule) involving the Hurricanes, Highlanders, Brumbies and Waratahs were exchanged. The 2010 schedule is *exactly* the 2008 schedule played in reverse order of rounds, and the 2009 schedule is *exactly* the 2007 schedule played in reverse order of rounds. Obviously little account has been made of the availability of venues.

Analysis. The number of matches played in each country in each round is highly unbalanced, e.g., 2006 rounds 5 and 11 have only one match played in NZ, 2006 rounds 7–10 all have only one match played in SA, and 2007 round 3 has 5 matches played in SA. All NZ teams play either two SA teams and Force or three SA teams away in consecutive rounds. All SA teams play two AU teams plus either two or three NZ teams away in consecutive rounds. Every team beings and ends the competition on their own side of the Indian Ocean. There are some long sequences of home matches for SA teams, e.g., Stormers 2006 and Cheetahs 2007 have five then a bye. There are some away travel inefficiencies, e.g., Stormers 2007 have an away run of two NZ teams, two AU teams then back to NZ for one NZ team. Disappointingly, a bye seldom follows an away match in another country: in 2006 (once), 2007 (five

Table 1: Super 14 actual 2006 schedule. Negative indicates an away match and upper case highlights away matches in another country. Counts of home matches in each country for each round are appended.

	Round													
Team	1	2	3	4	5	6	7	8	9	10	11	12	13	14
(bl)Blues	hu	-hi	-RD	-cr		br	-WR	bu	st	fo	-SH	-CT	-LI	cf
(cf)Chiefs	-SH	-LI	-FO	rd	cr		-BR	hi	bu	st	ct	-hu	wr	-bl
(cr)Crusade	hi	-RD	sh	bl	-cf	li		-hu	wr	ct	-FO	-ST	-BU	br
(hi)Highlan	-cr	bl	-CT	-ST	-BU	sh	li	-cf	fo	hu		wr	-BR	-RD
(hu)Hurrica	-bl	fo	li	-CT	-ST	-BU	sh	cr		-hi	-BR	cf	rd	-WR
(br)Brumbie	-fo	-BU	-ST	li	sh	-BL	cf		ct	-wr	hu	rd	hi	-CR
(fo)Force	br	-HU	cf		-rd	wr	bu	st	-HI	-BL	cr	-LI	-CT	-SH
(rd)Reds	wr	cr	bl	-CF	fo		-CT	-SH	-LI	bu	st	-br	-HU	hi
(wr)Waratah	-rd	-ST	-BU	sh	li	-fo	bl	ct	-CR	br		-HI	-CF	hu
(bu)Bulls	-ct	br	wr		hi	hu	-FO	-BL	-CF	-RD	li	sh	cr	-st
(ct)Cheetah	bu	-sh	hi	hu		-st	rd	-WR	-BR	-CR	-CF	bl	fo	li
(li)Cats	st	cf	-HU	-BR	-WR	-CR	-HI		rd	sh	-bu	fo	bl	-ct
(sh)Sharks	cf	ct	-CR	-WR	-BR	-HI	-HU	rd		-li	bl	-bu	st	fo
(st)Stormer	-li	wr	br	hi	hu	ct		-FO	-BL	-CF	-RD	cr	-sh	bu
NZ home	2	2	2	2	1	3	2	3	4	4	1	2	2	2
AU home	2	1	2	2	3	1	3	2	1	2	3	1	1	2
SA home	3	4	3	2	2	2	1	1	1	1	2	4	4	3

times), 2008 (four times), 2009 (once), and 2010 (twice). Also, Chiefs 2006 and Highlanders 2007 have four home matches in a row, and Blues 2007 have three home matches in a row, then a bye, then another home match. Given the recycling of the schedules over the period 2006–2010, these issues have been repeated several times, inconveniencing the same teams each time.

Assumed preferences. Minimise long distance travel, i.e., each team crosses the Indian Ocean exactly twice (to and from SA, crossing 10 time zones) and schedule trips to Perth on the way to/from SA. When not on tour, NZ and AU teams play at most three consecutive matches at home and at most three consecutive matches away (to break long runs of home matches). At least one match in each country in each round (due to television). Attempt to schedule byes following Indian Ocean crossings. No byes in the first three or last three rounds. SA teams cross the Tasman Sea only twice.

Suggested schedules. An IP model of the Super 14 has been developed, building on previous work of (Ball 2008). The proposed schedule in Table 3 shows that it is possible to satisfy NZ and AU teams playing at most three consecutive home or away matches, but Lions still have a run of five home matches. Note that (While and Barone 2007) have also previously considered models of the Super 14.

3 Super 15 (2011–2015)

The Melbourne Rebels have been added to the competition (now the Super 15) so there are now five teams each from New Zealand, Australia and South Africa (Wikipedia 2010b). The design specification for the Super 15 was initially rather vague, and hence we set out to develop a number of scenarios to explore the consequences of the design decisions which were (at that time) yet to be revealed. Finally,

Table 2: Super 14 actual 2007 schedule.

	Round													
Team	1	2	3	4	5	6	7	8	9	10	11	12	13	14
(bl)Blues	cr	-BR	-hu	rd	hi	li		wr	-cf	ct	sh	-ST	-BU	-FO
(cf)Chiefs	br	hu	-ST	-BU	-CT		li	-RD	bl	-hi	fo	sh	-WR	-cr
(cr)Crusade	-bl	rd	-LI	-CT	-SH		bu	st	-WR	fo	-hi	hu	-BR	cf
(hi)Highlan	-FO	-LI	-SH	st	-bl	rd		bu	ct	cf	cr	-WR	-hu	br
(hu)Hurrica	-RD	-cf	bl	br	st	-FO	-SH	-LI	bu		ct	-cr	hi	wr
(br)Brumbie	-CF	bl	-rd	-HU	bu	st	-CT	-SH	-LI	wr		fo	cr	-HI
(fo)Force	hi	-ST	-BU	li	-wr	hu	rd		sh	-CR	-CF	-br	ct	bl
(rd)Reds	hu	-CR	br	-BL	li	-HI	-fo	cf		sh	-wr	ct	-ST	-BU
(wr)Waratah	-LI	-SH	-CT		fo	bu	st	-BL	cr	-br	rd	hi	cf	-HU
(bu)Bulls	-sh	ct	fo	cf	-BR	-WR	-CR	-HI	-HU		st	-li	bl	rd
(ct)Cheetah	st	-bu	wr	cr	cf	sh	br		-HI	-BL	-HU	-RD	-FO	-li
(li)Lions	wr	hi	cr	-FO	-RD	-BL	-CF	hu	br	-st		bu	-sh	ct
(sh)Sharks	bu	wr	hi		cr	-ct	hu	br	-FO	-RD	-BL	-CF	li	-st
(st)Stormer	-ct	fo	cf	-HI	-HU	-BR	-WR	-CR		li	-bu	bl	rd	sh
NZ home	2	2	1	3	2	2	2	3	3	3	4	2	1	3
AU home	2	1	1	1	3	3	2	1	2	2	1	3	3	1
SA home	3	4	5	2	2	1	2	2	1	1	1	2	3	3

we will compare our results with the actual published 2011 schedule.

Initial competition structure. Each team plays the other four teams from its country (called a *conference*) in a double-round-robin, i.e., once at home and once away. Each team also plays four of the teams from each of the other conferences once, two teams from each conference at home and two away. Hence, each team plays every other team except for one team from each of the other countries. Each team plays eight intra-conference matches and eight inter-conference matches, for a total of 16 matches in the round-robin phase. The top team from each conference, plus the three highest placed remaining teams, progress to a six team finals series played over three weekends.

Design questions. Clearly, some difficulties from the Super 14 scheduling can be overcome by the design, e.g., each NZ/SA team only plays two matches in SA/NZ and two matches in AU. However, given the initial description of the competition structure, it was still to be determined who each team specifically does not play, and the round-by-round structure of the tournament. Selection of teams not to play could have been resolved from rankings based on previous Super 14 finishing places or win-loss records against particular opponents, but were finally revealed to be based on alphabetical order of teams within each conference (The Australian 2010). Hence the sets of teams who do not play each other in 2011 are as follows: {Blues, Brumbies, Bulls}; {Chiefs, Force, Cheetahs}; {Crusaders, Rebels, Lions}; {Highlanders, Reds, Sharks}; and {Hurricanes, Waratahs, Stormers}.

It was initially unknown whether there would be fixed rounds of only intra-conference matches, how the venues would be decided for the inter-conference matches (would they be related to 2010 Super 14 venues), whether the whole competition is contiguous, and how many byes each team has. There are $4+4+4+4 = 16$ matches for each team for a total of $15 \times 16/2 = 120$ matches. Since there are at most seven matches per round we require at least 18 rounds to complete all the matches, hence each team must have at least two byes.

Table 3: Super 14 proposed alternative 2009 schedule.

	Round													
Team	1	2	3	4	5	6	7	8	9	10	11	12	13	14
(bl)Blues	-BR	wr	-FO	hi	-cf	-hu	rd		li	sh	ct	-ST	-BU	cr
(cf)Chiefs	hu	-ST	-BU	-CT	bl	-RD		br	-WR	li	sh	-cr	-hi	fo
(cr)Crusade	fo	-LI	-CT	-SH	st		hu	-hi	bu	rd	-WR	cf	-BR	-bl
(hi)Highlan	-WR	-FO	st	-bl	-LI	-SH	bu	cr	ct		rd	br	cf	-hu
(hu)Hurrica	-cf	-SH	-LI	st	wr	bl	-cr	bu	br	ct		-FO	-RD	hi
(br)Brumbie	bl	-CT	-SH	-LI	bu	st		-CF	-HU	wr	fo	-HI	cr	-rd
(fo)Force	-CR	hi	bl	-wr	rd		li	sh	-ST	-BU	-br	hu	ct	-CF
(rd)Reds	-ST	-BU	-wr		-fo	cf	-BL	li	sh	-CR	-HI	ct	hu	br
(wr)Waratah	hi	-BL	rd	fo	-HU	bu	st		cf	-br	cr	-LI	-SH	-CT
(bu)Bulls	ct	rd	cf		-BR	-WR	-HI	-HU	-CR	fo	st	-sh	bl	-li
(ct)Cheetah	-bu	br	cr	cf		-li	sh	st	-HI	-HU	-BL	-RD	-FO	wr
(li)Lions	-sh	cr	hu	br	hi	ct	-FO	-RD	-BL	-CF		wr	-st	bu
(sh)Sharks	li	hu	br	cr		hi	-ct	-FO	-RD	-BL	-CF	bu	wr	-st
(st)Stormer	rd	cf	-HI	-HU	-CR	-BR	-WR	-ct	fo		-bu	bl	li	sh
NZ home	2	1	1	2	3	1	3	3	4	4	3	2	1	3
AU home	2	1	2	1	2	3	2	2	2	1	2	2	3	1
SA home	3	5	4	3	1	2	1	1	1	1	1	3	3	3

Assumptions and scenarios. We assume that the venues for inter-conference matches are decision variables. All teams begin and end the competition on their own side of the Indian Ocean and cross it exactly twice. We propose three basic scenarios for the structure of the competition: *phased*, *semi-phased* and *unphased*. These scenarios are illustrated and analysed as follows. In each case, we have used subcost-guided simulated annealing (Wright 2001), with move structures based on Kempe chains (Wright 1994), to find good solutions.

3.1 Phased Competition

Suppose that the first five rounds consist of a single-round-robin for each conference and the last five rounds consist of another single-round-robin for each conference (with venues reversed). The middle rounds hold all $15 \times 8/2 = 60$ inter-conference matches. Since there are at most seven matches per round we require at least nine rounds to complete all the inter-conference matches. Table 4 gives an example of a phased competition with $5 + 5 + 9 = 19$ rounds. Here each team has three byes. Notice that this creates the situation where many teams have (ignoring the bye) a run of four home matches and a run of four away matches in the inter-conference rounds. Also, the Rugby World Cup 2011 puts pressure on the desirable number of weeks for the 2011 competition, so 19 rounds is undesirable if 18 are possible. Note that there are two rounds with no SA home matches and two rounds with five SA home matches.

3.2 Semi-Phased Competition

Suppose that the first five and last five rounds are as in the phased competition, but that at most one inter-conference match can also be played in each of these rounds. Hence all intra-conference matches must occur within the first five and last five rounds. Table 5 gives an example of a semi-phased competition with 18 rounds

Table 4: Super 15 proposed *phased* 2011 schedule.

	[				]				Round						[				]
Team	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
(bl)Blues	cf	-cr		hi	-hu	fo	-RB	-WR	-ST	-LI	ct	sh		rd	hu	cr	-hi		-cf
(cf)Chiefs	-bl	-hu	hi	cr		-BU	-ST	-FO	br		sh	rd	-WR	li	-cr		hu	-hi	bl
(cr)Crusade		bl	-hu	-cf	-hi	-RB		br	wr	bu	st	-FO	-CT	-SH	cf	-bl		hu	hi
(hi)Highlan	hu		-cf	-bl	cr		fo	-CT	-BU	-RD	-BR	st	li	rb		-hu	bl	cf	-cr
(hu)Hurrica	-hi	cf	cr		bl	-SH	-LI	-RD		wr	bu	ct	rb	-BR	-bl	hi	-cf	-cr	
(br)Brumbie	fo	-rd		rb	wr	-LI	-SH	-CR	-CF	ct	hi		st	hu	-rb	-fo	-wr	rd	
(rb)Rebels	wr	fo	rd	-br		cr	bl	-ST	-LI	sh		bu	-HU	-HI	br	-rd	-fo		-wr
(rd)Reds		br	-rb	-wr	fo	-ST	-CT	hu	sh	hi		-CF	bu	-BL	wr	rb		-br	-fo
(wr)Waratah	-rb		fo	rd	-br	-CT	-BU	bl	-CR	-HU		li	cf	st	-rd		br	-fo	rb
(fo)Force	-br	-rb	-wr		-rd	-BL	-HI	cf	ct		li	cr	-SH	-BU		br	rb	wr	rd
(bu)Bulls	-li	st		ct	-sh	cf	wr		hi	-CR	-HU	-RB	-RD	fo		li	-ct	-st	sh
(ct)Cheetah	st	-sh	li	-bu		wr	rd	hi	-FO	-BR	-BL	-HU	cr		-li	-st	bu	sh	
(li)Lions	bu		-ct	sh	-st	br	hu		rb	bl	-FO	-WR	-HI	-CF	ct	-bu	-sh		st
(sh)Sharks		ct	st	-li	bu	hu	br		-RD	-RB	-CF	-BL	fo	cr	-st		li	-ct	-bu
(st)Stormer	-ct	-bu	-sh		li	rd	cf	rb	bl		-CR	-HI	-BR	-WR	sh	ct		bu	-li
NZ home	2	2	2	2	2	1	1	1	2	2	4	4	2	3	2	2	2	2	2
AU home	2	2	2	2	2	1	1	3	2	3	2	3	3	2	2	2	2	2	2
SA home	2	2	2	2	2	5	5	2	3	1	0	0	2	2	2	2	2	2	2

(each team with two byes). There is now a better spread of tours and slightly more balanced number of home matches in each country.

3.3 Unphased Competition

Suppose that there are no restrictions of when intra-conference and inter-conference matches are played. Table 6 gives an example of an unphased competition with 18 rounds (each team with two byes). The spread of number of home matches in each country in each round is slightly improved over semi-phased (only round 12 has four home matches in one country) but this is still far from the goal of at least two matches in each country in each round. However, there are some particularly bad features: Stormers and Bulls play each other twice within the first three rounds; three byes in round 3 (very early) and in round 17 (very late); Chiefs play Crusaders twice before playing Blues (many instances of this); several runs of five matches without an away matches (Hurricanes, Brumbies, Lions, and six for Sharks); and Brumbies and Force each have a run of four away matches.

3.4 Actual Competition

Table 7 gives the actual published schedule for the 2011 competition (Wikipedia 2010b). This is clearly an unphased competition design. A Canadian company Optimal Planning were involved in producing the competition schedule (Optimal Planning 2010). They list a large number of well known national sports leagues in their client list, including NFL and AFL (USA), and NRL and A-league (Australia). In general this is a remarkably good schedule and a considerable improvement upon the schedule in Table 6: at least two home matches in each country in every round; good separation between return fixtures; no NZ or AU team has a run of four away matches; and all except Stormers have a maximum two consecutive home matches. Some minor criticisms: many instances of byes between home matches;

Table 5: Super 15 proposed *semi-phased* 2011 schedule.

	[				]				Round					[				]
Team	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
(bl)Blues	-cf	hi	-cr		-hu	rd	-WR	li	-SH	-CT	st		rb	-BR	cr	-hi	hu	cf
(cf)Chiefs	bl	-cr	hu	hi		-ST	-BU	ct	rb		-FO	sh	-BR	cr	-hu	wr	-hi	-bl
(cr)Crusade	-hi	cf	bl	-hu	hi	-RB		-RD	bu	st	br		fo	-cf	-bl	hu	-LI	-CT
(hi)Highlan	cr	-bl	fo	-cf	-cr		li	wr	-RD	-RB	sh	-BU	-ST	hu		bl	cf	-hu
(hu)Hurrica	-LI	-SH	-cf	cr	bl	-WR	rd		ct	br	bu	-FO		-hi	cf	-cr	-bl	hi
(br)Brumbie	fo	wr	rd		-rb	ct	-ST	-SH	li	-HU	-CR		cf	bl	-fo	-rd	rb	-wr
(rb)Rebels		fo	-wr	-rd	br	cr	ct	bu	-CF	hi	-LI	-ST	-BL	rd	wr	-fo	-br	
(rd)Reds	wr		-br	rb	fo	-BL	-HU	cr	hi	bu	-CT	-LI	sh	-rb		br	-wr	-fo
(wr)Waratah	-rd	-br	rb	-fo		hu	bl	-HI	st	sh		-CT	-BU	fo	-rb	-CF	rd	br
(fo)Force	-br	-rb	-HI	wr	-rd	-BU	-SH	st		li	cf	hu	-CR	-wr	br	rb		rd
(bu)Bulls	-ct	li		-st	sh	fo	cf	-RB	-CR	-RD	-HU	hi	wr	-sh		ct	st	-li
(ct)Cheetah	bu	-st	sh		li	-BR	-RB	-CF	-HU	bl	rd	wr		-li	st	-bu	-sh	cr
(li)Lions	hu	-bu	st	sh	-ct		-HI	-BL	-BR	-FO	rb	rd		ct	-sh	-st	cr	bu
(sh)Sharks	-st	hu	-ct	-li	-bu		fo	br	bl	-WR	-HI	-CF	-RD	bu	li		ct	st
(st)Stormer	sh	ct	-li	bu		cf	br	-FO	-WR	-CR	-BL	rb	hi		-ct	li	-bu	-sh
NZ home	2	2	3	2	2	1	2	3	3	2	4	1	2	2	2	3	2	2
AU home	2	2	2	2	2	3	2	3	3	4	1	1	2	3	2	2	2	2
SA home	3	3	2	2	2	2	3	1	1	1	2	4	2	2	2	2	3	3

Lions play ten matches before first bye (after Stormers second bye); Highlanders play Crusaders twice before Blues (also for Bulls, Lions and Sharks); Crusaders play all NZ conference teams away before playing any at home; and Blues and Chiefs have simultaneous byes twice.

4 ITM Cup

The ITM Cup (since 2010) is the highest level domestic rugby competition in New Zealand (Wikipedia 2010a), featuring semi-professional teams from 14 provincial unions (see Table 8). It was previously known as the Air New Zealand Cup (2006–2009). In 2009 and 2010, a single-round-robin competition was played, with the home/away status of individual matches alternating between years. It has been proposed that the 2011 competition follow a new competition design.

> "The new competition format from 2011 will see 14 teams split into two divisions of seven teams based on their on-field finishing positions in 2010. The top seven teams will form the Premiership and the bottom seven the Championship. Teams will play all other teams in their division plus four other teams from the other division (there will be an innovative new process for teams to select their cross-division opponents with the detail to be finalised in the first quarter next year). All matches will carry full competition points. The winner of the Championship will receive automatic promotion to the Premiership replacing the 7th placed team in the Premiership which will be relegated to the Championship. In 2011, due to New Zealand hosting RWC 2011, the competition window will be restricted to eight weeks. As a result, in 2011 only, there will be three mid-week matches and no semi-finals."
>
> (Manawatu Rugby Union 2010)

Table 6: Super 15 proposed *unphased* 2011 schedule.

	Round																	
Team	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
(bl)Blues	cr	rb		hi	-hu	ct	rd	-LI	-ST	-FO	sh	-cf		-WR	hu	-hi	-cr	cf
(cf)Chiefs	hi	wr	-cr	rb	-BR	-RD	-hu	cr	-hi	sh		bl	-ST	-LI	bu	hu		-bl
(cr)Crusade	-bl	-hi	cf	-SH	-BU	st	ct	-cf	hu		wr	-RD	-BR	hi	fo		bl	-hu
(hi)Highlan	-cf	cr	li	-bl	st	-RB		fo	cf	-BU	-CT	-WR	hu	-cr		bl	-hu	br
(hu)Hurrica	-FO	-CT	-SH	li	bl		cf	br	-cr	rd	-RB		-hi	bu	-bl	-cf	hi	cr
(br)Brumbie	-wr	rd	-rb	ct	cf		rb	-HU	-LI	-ST	-fo	sh	cr		wr	fo	-rd	-HI
(fo)Force	hu	li		-BU	-SH	wr	st	-HI	-rd	bl	br		rb	rd	-CR	-br	-wr	-rb
(rb)Rebels	-rd	-BL	br	-CF	ct	hi	-br	rd		-wr	hu	bu	-fo	-ST	-SH	wr		fo
(rd)Reds	rb	-br	wr	st		cf	-BL	-rb	fo	-HU		cr	bu	-fo	-LI	-CT	br	-wr
(wr)Waratah	br	-CF	-rd		li	-fo	-BU	-CT	sh	rb	-CR	hi		bl	-br	-rb	fo	rd
(bu)Bulls	st		-st	fo	cr	-li	wr	-sh	ct	hi		-RB	-RD	-HU	-CF	sh	li	-ct
(ct)Cheetah	sh	hu		-BR	-RB	-BL	-CR	wr	-bu	li	hi	-st	-li	-sh	st	rd		bu
(li)Lions		-FO	-HI	-HU	-WR	bu	-sh	bl	br	-ct	st		ct	cf	rd	-st	-bu	sh
(sh)Sharks	-ct	-st	hu	cr	fo		li	bu	-WR	-CF	-BL	-BR		ct	rb	-bu	st	-li
(st)Stormer	-bu	sh	bu	-RD	-HI	-CR	-FO		bl	br	-li	ct	cf	rb	-ct	li	-sh	
NZ home	2	3	2	3	2	2	3	3	2	2	2	1	1	2	3	2	2	3
AU home	3	2	2	2	3	3	2	1	2	2	2	4	3	2	1	2	2	2
SA home	2	2	2	2	2	1	2	3	3	3	2	1	2	3	3	3	2	2

Design challenges. What are the design decisions as a result of compressing the competition into seven weekend and six midweek rounds? Assume that all teams play all seven weekend rounds (maximum TV coverage) and that the number of midweek matches each week is minimised (assume either three or four midweek matches each week). Then each team will play exactly three midweek matches. An important design question is then whether it is possible to ensure that each team plays at most three matches in a row without a bye. *This is extremely important in terms of player welfare and the size of squad that must be employed.* This implies that six teams play midweek rounds $\{2, 6, 10\}$, six teams play midweek rounds $\{4, 8, 12\}$ and the remaining two teams either both play one of these, $\{4, 6, 10\}$ or $\{4, 8, 12\}$. The schedule proposed in Table 8 shows that this is indeed possible. But what is the effect upon the schedule of enforcing this requirement?

Assumed preferences. Home-away-home runs should be avoided. Travel distance should be minimised. If long-distance travel is necessary, e.g., Northland to Southland, then this is within a mini-tour of two or more away matches. At least two teams from each division are involved in each round, ensuring that there is some mixing of the premiership and championship matches in the midweek rounds.

Proposed schedule. Table 8 gives a proposed schedule (but using team seedings from the 2009 competition finishing order). Two rounds, marked [7] and [9], are interdivision-only rounds. The << and >> symbols indicate a *flexible* match in which the two teams both have a bye immediately before/after, and hence it would be possible to shift that "weekend" match earlier/later. The resulting schedule shows that: there a some problems for Southland (away to North Harbour, home, then away to Wellington) and Counties (mini-tour away to Southland, Northland and Tasman); the final round has no flexible matches; but there are a lot of very nice little sequences of away matches. Note that no actual schedule for the 2011 ITM Cup has yet been published (the 2010 final was played on 5th November 2010).

Table 7: Super 15 *actual* published 2011 schedule.

	Round																	
Team	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
(bl)Blues	cr	-SH	-LI	-FO	hu	-cf	ct		wr	rb	-hi	-hu	-RD	st		cf	-cr	hi
(cf)Chiefs	-BR	-hi	rb	-hu	sh	bl	-WR		cr	-LI	-BU	hi	st	-cr		-bl	hu	rd
(cr)Crusade	-bl	-hu	wr	br	-hi	sh		bu	-cf	hi	-FO	-ST	-CT	cf	-RD		bl	hu
(hi)Highlan	-hu	cf	-BU	-ST	cr		br	ct	-RB	-cr	bl	-cf	hu		li	fo	-WR	-bl
(hu)Hurrica	hi	cr		cf	-bl	-RB	bu	-BR	-CT	-SH	rd	bl	-hi		fo	li	-cf	-cr
(br)Brumbie	cf	-rb	rd	-CR		wr	-HI	hu	fo		-CT	-SH	li	-fo	st	-rd	rb	-wr
(fo)Force	-rd		sh	bl	-LI	-ST	rb	wr	-br	bu	cr	-wr		br	-HU	-HI	rd	-rb
(rb)Rebels	wr	br	-CF	sh	-rd	hu	-fo		hi	-BL	-wr	rd	-BU	-CT		st	-br	fo
(rd)Reds	fo	-wr	-br		rb	ct	-LI	-ST	bu	wr	-HU	-rb	bl		cr	br	-fo	-CF
(wr)Waratah	-rb	rd	-CR		ct	-br	cf	-fo	-BL	-rd	rb	fo		li	-SH	-BU	hi	br
(bu)Bulls	-li	-ct	hi		st	li	-HU	-CR	-RD	-FO	cf		rb	-sh	ct	wr	-st	sh
(ct)Cheetah	-sh	bu	-st	li	-WR	-RD	-BL	-HI	hu		br	-li	cr	rb	-bu	sh		st
(li)Lions	bu	-st	bl	-ct	fo	-bu	rd	-sh	st	cf		ct	-BR	-WR	-HI	-HU	sh	
(sh)Sharks	ct	bl	-FO	-RB	-CF	-CR	st	li		hu	-st	br		bu	wr	-ct	-li	-bu
(st)Stormer		li	ct	hi	-bu	fo	-sh	rd	-li		sh	cr	-CF	-BL	-BR	-RB	bu	-ct
NZ home	2	2	2	2	3	2	3	2	2	2	2	2	2	2	2	3	2	3
AU home	3	2	2	2	2	3	2	2	3	2	2	2	2	2	2	2	3	2
SA home	2	3	3	2	2	2	2	2	2	2	3	3	2	2	2	2	2	2

5 Conclusions

Super 14. It is very difficult to ensure that SA teams don't have long sequences of home matches, schedule byes usefully, and balance number of matches in each country in each round.

Super 15. The phased and semi-phased competition designs produce noticeable difficulties in fairly scheduling the middle of the competition, whereas the flexibility of an unphased competition gives scope for consideration of other preferences. There remains some modelling issues in terms of separation of byes, separation of return fixtures, spreading of home matches through the season, whether it matters if one side has played two more matches than another at any stage, and whether it is preferred not to repeat last year's venue for matches between teams from different countries.

ITM Cup. The schedule design looks achievable with careful attention to timing of individual matches.

References

Ball, C. 2008. "Scheduling super 14 rugby." *Proceedings of the 43rd Annual Conference of the Operational Research Society of New Zealand.* Wellington, 86–95.

Cleaveland Live. 2009. Dissatisfaction guaranteed: major league baseball can't please anyone with scheduling. `http://www.cleveland.com/tribe/index.ssf/2009/05/dissatisfaction_guaranteed_maj.html`.

Herald Scotland. 2010. Rangers accuse SPL of favouring Celtic in top-six fixture schedule. `http://www.heraldscotland.com/sport/spl/rangers/`

Table 8: ITM Cup proposed 2011 schedule. Negative indicates away match, upper case for premiership teams, lower case for championship teams, dedicated interdivision only rounds 7 and 9, odd rounds are weekends, and even rounds are midweek.

	Round												
Team	1	2	3	4	5	6	[7]	8	[9]	10	11	12	13
(CN)Canterbury	ts		<<-SL	HB	WK>>		-nl	-cm	ot		WL	-BP	-AK
(WL)Wellington	-WK	-nl	BP		-HB	mn	-nh>>		tn	AK	-CN		SL
(SL)Southland	HB>>		<<CN	-BP	-AK		-ot	WK	cm>>		-nh	tn	-WL
(HB)Hawke's Bay	-SL>>		AK	-CN	WL		<<-tn	BP	ts>>		-mn	-WK	nl
(AK)Auckland	-BP	WK	-HB		SL	nh	-cm		<<nl	-WL	-ot>>		CN
(WK)Waikato	WL	-AK	cm		-CN>>		-mn	-SL	nh		<<-ts	HB	BP
(BP)Bay of Plenty	AK		-WL	SL	ot		<<-ts	-HB	mn		<<-tn	CN	-WK
(tn)Taranaki	mn		nl	-cm	-nh		<<HB	ts	-WL		<<BP	-SL	-ot
(ts)Tasman	-CN	ot	nh>>		<<-nl		<<BP	-tn	-HB>>		<<WK	cm	-mn
(ot)Otago	-cm	-ts	mn>>		-BP	nl	SL		-CN	-nh	AK>>		tn
(mn)Manawatu	-tn	nh	-ot>>		cm	-WL	WK		-BP	-nl	HB		ts
(nh)North Harbour	nl	-mn	-ts>>		tn	-AK	WL>>		-WK	ot	SL		-cm
(nl)Northland	-nh	WL	-tn		<<ts	-ot	CN		<<-AK	mn	cm		-HB
(cm)Counties Man	ot		-WK	tn	-mn		AK	CN	-SL>>		-nl	-ts	nh
Flexible:	0/1		1/2		1/1		2/1		1/2		2/1		0/0

rangers-accuse-spl-of-favouring-celtic-in-top-six-fixture-schedule-1.1021531.

Kendall, G., S. Knust, C.C. Ribeiro, and S. Urrutia. 2010. "Scheduling in sports: an annotated bibliography." *Computers and Operations Research* 37:1–19.

Manawatu Rugby Union. 2010. 2 Pools of 7 for Air New Zealand Cup from 2011 onwards. http://www.manawaturugby.co.nz/article/1863.html.

Optimal Planning. 2010. Client list. http://www.optimalplanning.com/clients/client_list.htm.

The Australian. 2010. Getting a good run in the new Super 15 draw could be as easy as ABC. http://www.theaustralian.com.au/news/sport/getting-a-good-run-in-the-new-super-15-draw-could-be-as-easy-as-abc/story-e6frg7mf-1225851594077.

While, L., and L. Barone. 2007. "Super 14 rugby fixture scheduling using a multi-objective evolutionary algorithm." *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling (CI-Sched 2007).* Honolulu: IEEE, 25–42.

Wikipedia. 2010a. ITM Cup. http://en.wikipedia.org/wiki/ITM_Cup.

———. 2010b. Super Rugby. http://en.wikipedia.org/wiki/Super_Rugby.

Wright, M.B. 1994. "Timetabling county cricket fixtures using a form of tabu search." *Journal of the Operational Research Society* 45:758–770.

———. 2001. "Subcost-guided simulated annealing." Edited by P. Hansen and C.C. Ribeiro, *Essays and Surveys in Metaheuristics.* Dordrecht: Kluwer Academic Publishers, 631–639.

———. 2009. "Fifty years of operational research in sports." *Journal of the Operational Research Society* 60:161–168.

Demand Learning and Dynamic Pricing for Multi-Version Products*

Guillermo Gallego
Department of Industrial Engineering and Operations Research
Columbia University
New York, USA
ggallego@ieor.columbia.edu

Masoud Talebian
School of Mathematics, University of Newcastle
University of Newcastle
Newcastle, Australia
New Zealand
Masoud.Talebian@newcastle.edu.a

Extended Abstract

Dynamic pricing deals with demand-management decisions and the required methodologies and systems for making these decisions. The demand function is an important element in such decision makings. While most papers assume that the provider is aware of the demand function, this is rarely the case in practice; indeed, market uncertainty is prevalent in most cases. This unrealistic assumption is one of the reasons that, despite the enormous amount of data made available to decision makers, intelligent systems that balance the supply and demand and improve profits have found limited use in the industry. In addition to the manager's experience of the market based on historic sales, managers incorporate sales to update demand estimates. This is known as demand learning.

A major source of motivation for this paper is the case of demand learning in the event management industry, where event tickets are priced over time and demand is uncertain. Events include, but are not limited to, sports, concerts, and theater shows. In particular, we consider tours of unique events that are organized in different cities. In these cases, uncertain demand plays a major role in decisions regarding new shows.

In this business environment, capacity provider needs to learn two elements of demand: market size and the core value of the event, to properly price different variants of the product. Market size means the expected number of interested

*The article is appearing Journal of Revenue and Pricing Management customers for the event, or what is referred to as the arrival rate. The core value of an event refers to the value of the performance, i.e., the customers' willingness to pay to see the performance. The core value is usually unknown. Seats in specific locations are assumed to have values that are known multiplier of the core value. These multipliers depend on the seat location and are assumed to be the same for all events in a specific category. By analyzing previous event ticket sales data and owing to her experience, the capacity provider is usually aware of the premium/discount effect of each seat location, e.g., balcony compared to orchestra, on the total value of the event perceived by customers. We do not assume any specific relation between the core value and the provider's costs; we also assume that the provider's variable cost is negligible. We use market prices and sales information to learn the arrival rate and the core value during the sales horizon.

To model this situation, we consider a capacity provider who offers multiple versions of a single product, such as different seat locations for an event. We assume that the different versions share an unknown core value and command a known premium or discount relative to the core value. Customers arrive at an unknown arrival rate during a finite sales horizon. We assume that the provider has some initial belief and prior knowledge about the arrival rate which is updated using Bayesian rule. Estimates of the core value are updated using maximum likelihood estimation. We describe an analytical framework to model the process of learning and pricing and apply this framework to address a practical problem. The main contributions of this paper can be summarized as follows.

First, we apply our framework to cases where there exist more than one version of a product. This is in contrast to the models of most previous studies that assume a single version. The literature focus on one-version products is, in part, due to the fact that models considering customer behavior when choosing from different versions of a product are relatively new. In the present study, we use the theory of multinomial logit (MNL) choice to model customers' behavior in choosing from different versions. This theory not only accurately models the real world by capturing the stochastic nature of customers' choice but is also analytically tractable, and can be easily estimated.

Second, we consider the case where both the arrival rate and the core value are unknown and should be estimated. Most previous studies assume that there exists only one unknown feature of demand: either the arrival rate or the core value. The reason behind this limitation may be the mathematical complications that arise from the combination of learning two unknowns. If we are not aware of the arrival rate and the core value, we cannot estimate the missed demand, i.e., interested customers who choose not to buy any version of the product because they cannot afford to buy it. This case is more realistic because the missed demand usually cannot be estimated in practice.

In this paper, we propose a method to learn demand and price a multi-version product. We show how to simultaneously estimate the unknown parameters as the sales evolve and how to price the products to maximize revenues under a rolling horizon framework. The following figures show the effectiveness of the proposed method in this case when demand is time variant. The demand is assumed to be of the quadratic form, consistent with Bass models. As we can see, the error percentage converges to zero as time increases. Moreover, we observe that with the increase of the prior coefficient of variation, C.V., the learning becomes more effective and the gap between the prior assumptions and the reality decreases more quickly. This means that a more diffuse prior, or a higher C.V., results in faster convergence. A higher C.V. can be interpreted as a lower confidence in the prior parameters.

Key Words: Demand Learning, Dynamic Pricing, Multinomial Logit Choice, Bayesian Update, Maximum Likelihood Estimation.

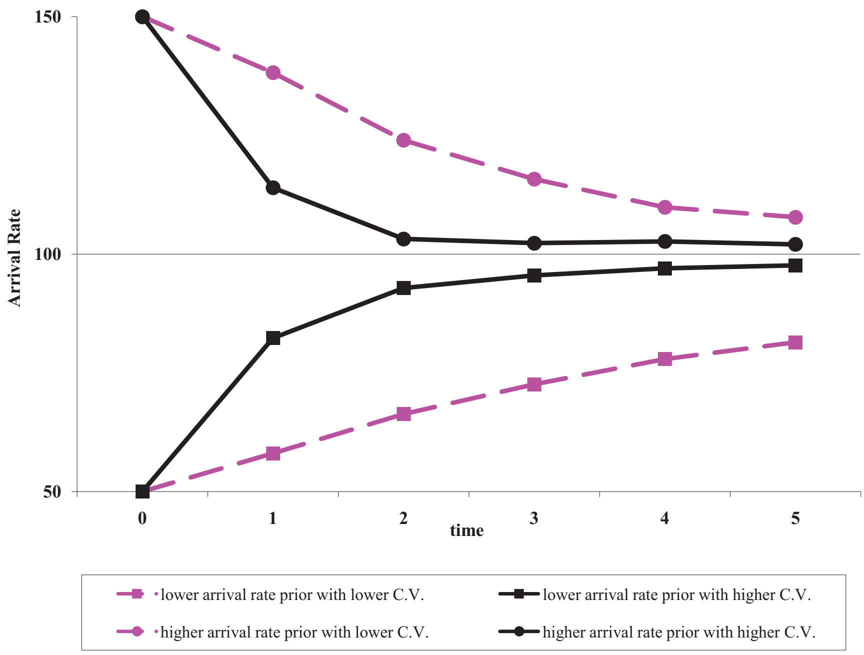


Figure 1: Effect of prior C.V. on learning arrival rate

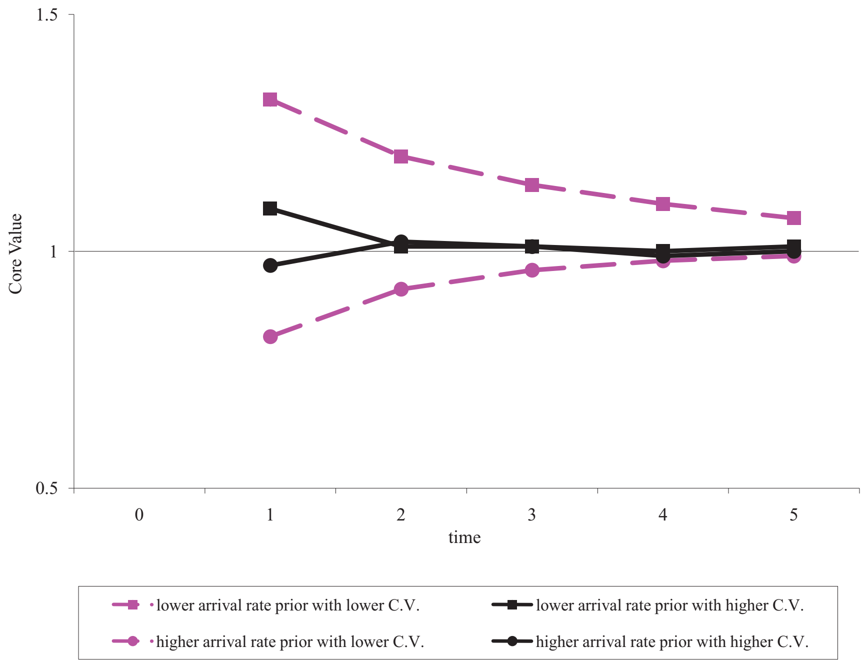


Figure 2: Effect of prior C.V. on learning core value

Financial Transmission Rights Auctions: Entry and Efficiency

Golbon Zakeri and Anthony Downward
Electric Power Optimization Centre
Department of Engineering Science
University of Auckland
New Zealand
g.zakeri@auckland.ac.nz

Abstract

Financial transmission rights (FTRs) are currently a topic of debate in New Zealand. Their purpose is to alleviate the risk associated to locational marginal pricing of electricity. In this paper we review the FTR auction design in practice and discuss the efficiency results for such auctions.

Key words: Financial Transmission Rights, Auctions, Electricity Markets.

1 Introduction

In the New Zealand wholesale electricity market, as well as many other jurisdictions world-wide, electricity supply is scheduled using an optimization model that minimizes the total cost of generation of electricity while complying with physical network constraints. (In New Zealand, this optimization problem is the Scheduling, Pricing and Dispatch (SPD) model that is solved by the transmission owner and operator, Transpower.) The solution provides optimal dispatch of electricity and the economically efficient spot prices for each node of the transmission network (these prices are referred to as locational marginal prices).

Although the wholesale market clearing problem, mentioned above, is an effective mechanism in integrating the power flow constraints, as a by product, it poses financial risk for generators and consumers, who have to pay the locational price of electricity. Figure 1 below (reproduced from the Locational Price Risk Management Analysis presentation (Sept 2010,) available from the the Electricity Commission's website,) depicts the average monthly price differences between the Otahuhu and Benmore nodes since January 2008. It is easy to observe from this figure that locational price differences can be substantial. A financial transmission right (FTR), otherwise known as a transmission congestion contract (TCC), is a financial instrument designed to manage risk associated with locational marginal price volatility. FTRs were first introduced by Hogan (Hogan 1992) and are used in several jurisdictions in the US including the Pennsylvania-New Jersey-Maryland (PJM) and

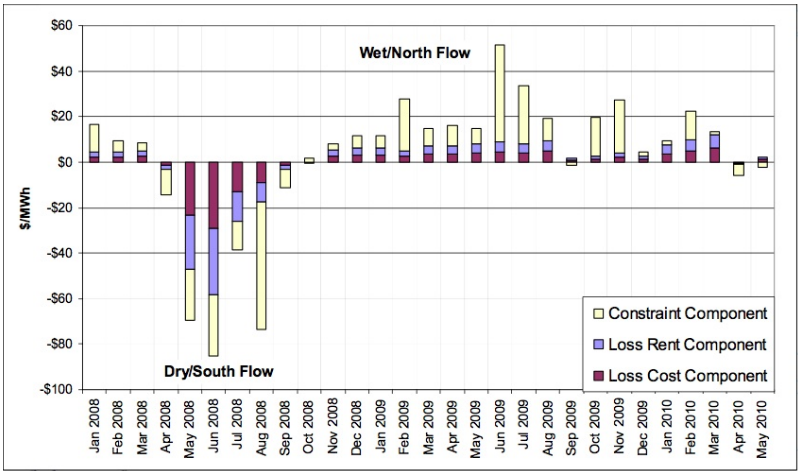


Figure 1: Average monthly price differences across Benmore and Otahuhu.

New York markets. Suppose that for a given period that the price at Otahuhu is \$40.00 more than that at Benmore. Then a firm equipped with an FTR of 100 MW from Benmore to Otahuhu, for that period, would be paid a coupon payment of $100 \times \$40.00 = \$4000.00$.

In the simplest form of FTR, the holder specifies a volume in MW and two nodes, an upstream and a downstream node, in a transmission network. The price at the downstream node minus that at the upstream node is multiplied by the FTR volume (specified in MW) and is paid to the holder as a coupon payment, for each period when the FTR is valid. These FTRs are referred to as *balanced point-to-point FTRs*. If the payment for an FTR can be exercised as an option, that is, it is only exercised if the downstream price exceeds the upstream price, then we have a so called *option FTR*. In absence of this requirement, the holder must pay the system operator any price difference between the downstream and upstream nodes multiplied by the FTR volume and this is termed an *obligation FTR*. An FTR can be specified more generally by a vector of nodal loads and injections, specified in MW, where the loads are taken to be positive. The coupon payment for any period is the inner product of this vector with the vector of nodal prices. In this paper we review revenue adequacy and simultaneous feasibility of extant FTRs. We will then outline the form of the current FTR auctions and provide some simple examples. We will discuss the efficiency of these auctions.

2 Revenue Adequacy, Simultaneous Feasibility and the FTR auction problem

In this section we will discuss the problem of revenue adequacy in a transmissions rights auction and provide results that ensure revenue adequacy under a set conditions referred to as the simultaneous feasibility conditions. Subsequent to this we will present the FTR auction clearing problem that is currently in use in various jurisdictions such as PJM and NYISO. We end this section by examining some simple examples to illustrate certain properties of these auctions. The examples attempt to capture the key features, as such we have tried to keep them as simple as possible.

Once an auction is settled and FTRs are allocated, the system operator is bound by an agreement to pay the FTR holder the corresponding coupon payment on the FTR for every period for which the FTR is valid. In each period, the ISO collects any congestion rent and redistributes these rents through FTRs. Therefore to maintain its credit standing, the ISO must ensure that the revenue collected with locational prices in the dispatch should at least be equal to the payments to the holders of FTRs in the same period. This property is referred to by the term *revenue adequacy.*

Revenue adequacy is guaranteed through the *simultaneous feasibility test* for the general case of economic dispatch problems when the transmission constraints are convex. To make this precise, we will follow the notation from Philpott and Pritchard (Philpott and Pritchard 2004). Consider the economic dispatch problem given by

$$\begin{array}{lll} \text{EDP: minimize} & \sum_i \sum_{j \in O(i)} c_j x_j & \\ \text{subject to} & g_i(f) + \sum_{j \in O(i)} x_j = d_i, & i = 1, 2, \cdots, n \\ & x \in X & \\ & f \in U. & \end{array}$$

Here

- x_j is the level of dispatch of tranche $j \in O(i)$ where $O(i)$ indicates the set of offered tranches at node i of the transmission network.
- c_j is the offer price (therefore the cost to the system) of tranche $j \in O(i)$.
- d_i is the demand at node i.
- f denotes the vector of flows on the transmission network links.
- $g_i(f)$ is a concave function that gives the amount of power flow entering node i when link flows are f. The function g takes account of any power losses in the network and we are assuming in this model that this function is concave. Clearly concave piecewise quadratic or linear loss functions would qualify here.
- We assume that X is a convex set that defines the tranche levels.
- We assume that the set of flows f lies in a convex set U that encapsulates any line capacities and other electrical constraints such as the loop flow constraints.
- The first constraint in the economic dispatch problem simply states that demand must be met at every node by production at the node and any electricity flowing into that node.

Using the constraints from EDP we can define the simultaneous feasibility test for obligation FTRs succinctly. Let the vector $h(\alpha)$ denote an extant obligation FTR contract for $\alpha = 1, 2, \cdots, A$ (this set is the index set for the FTRs, each or a number of which may belong to a player in the FTR market). For example a balanced point-to-point FTR of magnitude τ with upstream node i and downstream node j will be represented by the vector h where

$$h_k = \begin{cases} \tau & \text{if } k = j \\ -\tau & \text{if } k = i \\ 0, & \text{otherwise.} \end{cases}$$

The set of FTRs $h(\alpha)$ for $\alpha = 1, 2, \cdots, A$ are simultaneously feasible if and only if there exists a vector y where

$$\begin{array}{ll} \text{SFT:} & g_i(y) = \sum_\alpha h_i(\alpha), \quad i = 1, 2, \cdots, n \\ & y \in U. \end{array}$$

That is, treated as injections and withdrawals, the set of all FTRs that are extant for a single period, must comply with the transmission network constraints simultaneously.

A series of results prove that conditions specified in SFT are sufficient to guarantee revenue adequacy. The first of these results is by Hogan (Hogan 1992) for lossless networks, extended to quadratic losses by Bushnell and Stoft (Bushnell and Stoft 1996). Hogan subsequently proved a more general result extending the previous results to the case of nonlinear but smooth constraints. Philpott and Pritchard (Philpott and Pritchard 2004) prove that if EDP can be replaced by a convex problem, that is if the convexified version of the EDP delivers the same optimal solution as the EDP, then SFT is sufficient for revenue adequacy. In general EDP is not equivalent to a convex problem. For instance for periods when some nodal prices are negative (in presence of losses), EDP can not be reformulated as a convex problem that will deliver the same solution. Philpott and Pritchard (Philpott and Pritchard 2004) have demonstrated that in such periods the ISO may be faced with revenue inadequacy.

2.1 The FTR Simultaneous Feasibility Auction

As mentioned above, the ISO must ensure revenue adequacy for meeting its obligation of paying out the extant FTR coupon payments. Revenue adequacy must be ensured under every possible ensuing network configuration. Therefore the simultaneous feasibility test introduced above is replicated for all so called $n - 1$ contingencies and these constraints together form the constraints for the auction problem.

FTR market participants submit their benefit functions to the auctioneer. The objective of the auction is to maximize FTR revenues which is the same as maximizing the aggregate benefit function of the buyers. This is a form of sealed-bid, divisible good, uniform price auction. This auction accommodates multiple units as the market clears bids for FTRs involving different sets of nodes at once. In the PJM and the NYISO, participants commit a single quantity and bid price (PJMeFTR 2007), much like the tranches bid into the NZEM. The prices are set to the marginal clearing bids for each FTR.

In the literature (e.g. (Biskas, Ziogos, and Bakirtzis 2007) and (Deng, Oren, and Meliopoulos 2010),) the FTR simultaneously feasible auction (FTR-auction) is commonly defined using the concept of the power transfer distribution factor (PTDF) matrix. In the next subsection, we will formulate the FTR-auction, for the point-to-point obligation FTRs that uses the flow balance constraints, much like the EDP constraints, but is equivalent to the PTDF formulation.

2.1.1 Mathematical Formulation of the FTR Simultaneous Feasibility Auction

In this subsection we will present a formulation for the simultaneously feasible FTR auction for balanced point-to-point obligation FTRs. This model can easily be extended to include unbalanced FTRs (as well as option FTRs and reserved transmission rights).

$$
\begin{array}{lll}
\text{TRA-ND: maximize} & \sum_l \sum_{i,j} \beta^l_{ij} \tau^l_{ij} & \\
\text{subject to} & g_i(f_c) = \sum_l (\sum_{j \neq i} \tau^l_{ij} - \sum_{k \neq i} \tau^l_{ki}) & \forall c \in C \quad \forall i = 1, 2, \cdots, n \\
& f_c \in U_c & \forall c \in C \\
& 0 \leq \tau^l_{ij} \leq T^l_{ij}. &
\end{array}
$$

Here

- β^l_{ij} and T^l_{ij} denote respectively the bid price and quantity for the lth FTR with downstream node j and upstream node i.
- τ^l_{ij}, the decision variable, is the total amount of FTR l awarded between nodes i (upstream) and j (downstream).
- q_i is the net injection/withdrawal at node i as a result of awarding τ^l_{ij} FTRs.
- K_c denotes the vector of line capacities for the lines in the transmission network for contingency c.
- $g_i(f_c)$ is a function that maps the flows to injection/withdrawals and can include various forms of losses.

Note that $c \in C$ denotes the index of a contingency and f_c and U_c are respectively the flow and the electricity transmission constraints under contingency c.

Let π^c_i denote the optimal dual for constraint i, in contingency c in the first set of constraints in TRA-ND, that is the flow balance constraints. Then the market clearing price of the FTRs with upstream node i and downstream node j is defined as the difference $\sum_c \pi^c_j - \pi^c_i$ see e.g. (Biskas, Ziogos, and Bakirtzis 2007) and (PJMeFTR 2007).

As evident from either the TRA-ND or TRA-PTDF formulation, the FTR simultaneous feasibility auctions have embedded in them the transmission network constraints. Hence similar to the economic dispatch problem, they suffer from some peculiarities caused by the transmission network constraints. In the next subsection, we will attempt to demonstrate this with a small example.

2.2 Properties of the FTR Simultaneous Feasibility Auction

In this section we will provide a small illustrative example that explores the properties of FTR simultaneous feasibility auctions.

Example, two nodes. We start with a simple two node example. Consider a network consisting of two nodes (nodes 1 and 2 as depicted in Figure (2) and a 100MW line that links these nodes. Suppose that there is a bid for 99MW of FTR from node 1 to node 2, at price \$10.00. The FTR simultaneous feasibility auction will solve the optimization problem Ex-2-node.

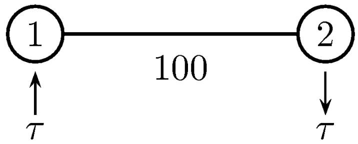


Figure 2: FTR over a simple 2 node transmission network.

$$\begin{array}{lll} \text{Ex-2-node:} & \text{maximize} & 10\tau \\ & \text{subject to} & \tau - f_{12} = 0 \quad [\pi_1] \\ & & f_{12} - \tau = 0 \quad [\pi_2] \\ & & f_{12} \leq 100 \quad [\eta^+] \\ & & -f_{12} \leq 100 \quad [\eta^-] \\ & & \tau \leq 99 \quad [\lambda] \\ & & \tau \geq 0. \end{array}$$

The optimal solution to this problem is given by $\tau = f_{12} = 99, \lambda = 10, \eta = 0$, and $\pi_1 = \pi_2$. This means that the clearing price of the FTR from node 1 to node 2 is zero (which is also the clearing price of the counterflow FTR going from node 2 to node 1). On the other hand if the there was a bid for 101MW of FTR from node 1 to node 2, at the same price of \$10.00, problem Ex-2-node would change slightly and the fourth constraint would be replaced by $\tau \leq 101$. This change would in turn change the solution to $\tau = f_{12} = 100, \lambda = 0, \eta = 10$, and $\pi_1 - \pi_2 = 10$. This would of course mean that the clearing price of the FTRs from node 1 to node 2 is now \$10.00. This is a byproduct of the transmission constraint, namely the capacity on the line from node 1 to node 2. When the volume of bids into the FTR auction is small, in this case less than the available capacity (100 MW) to be sold, the clearing price is well below what the participants are willing to pay. This is a feature of this uniform price auction that incorporates network constraints through simultaneous feasibility constraints.

3 Theoretical Results on the Efficiency of FTR Simultaneous Feasibility Auction

Deng et. al. (Deng, Oren, and Meliopoulos 2010) perform a theoretical analysis on current FTR markets and demonstrate that FTR markets enforcing the simultaneous feasibility constraints have inherent inefficiencies in the sense that auction clearing prices do not reflect expected nodal price differences. Their paper is motivated through empirical observations that in the New York ISO's TCC market, the clearing prices of TCC resulting from a simultaneous feasibility auction, differ significantly and systematically from the realized congestion revenues that determine the accrued payoffs of these rights (see e.g. (Siddiqui et al. 2005) and the discussion in the next section). Their model demonstrates that this inefficiency is not the result of lags in price discovery, rather the byproduct of the auction mechanism and the fact that cleared quantities of FTRs in the auction at all nodes is bounded (this is the case as the bid volumes for different FTRs is clearly bounded). They start with a premise where the FTR bidders are risk neutral. They further assume that the bidders have perfect foresight (i.e. know the expected nodal price differences,) and they bid in at these prices. They show that the FTR auction can produce clearing prices that are different from the expected prices.

To make this clear, we have investigated an small example that will explain the effect. Consider the network depicted below with three nodes and two lines. Note that there are two contingencies. In the first the capacity of the line linking nodes 2 and 3 is 200 MW and in the second, the capacity of this line reduces to 50MW. In this simple model, demand bids and supply offers remain the same in the

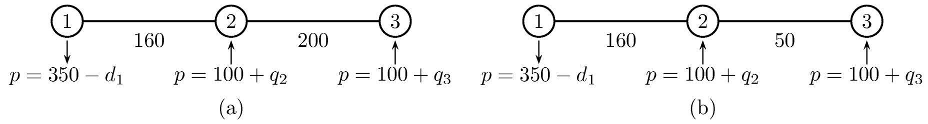


Figure 3: FTRs over a 3 node linear transmission network.

two contingencies. The demand bid and the supply offers are all linear functions indicated on the diagram. We will assume that the two contingencies are equally likely in our model. We have solved the conventional EDP market clearing problem in contingency (a) and (b) and have reported the clearing prices and quantities in Tables 1 and 2 respectively.

	(a)	(b)	E[π]
π_1	190.0	200.0	195.0
π_2	180.0	200.0	190.0
π_3	180.0	150.0	165.0

Table 1: Prices

	(a)	(b)	Ave
d_1	160.0	150.0	155.0
q_2	80.0	100.0	90.0
q_3	80.0	50.0	65.0

Table 2: Injections / Offtakes

Note that $E(\pi_1 - \pi_2) =$ \$5.00 and $E(\pi_2 - \pi_3) =$ \$25.00. The flows in each contingency scenario are provided in Table 3 below. Suppose now that the FTR auction participants are risk neutral and have perfect information about the price differences (namely they know that $E(\pi_1 - \pi_2) =$ \$5.00 and $E(\pi_2 - \pi_3) =$ \$25.00. This would mean that all participants would bid \$5.00 per MW for FTRs from node 2 to node 1 and all participants would bid \$25.00 per MW for FTRs from node 3 to node 2. If we now suppose that the total amount of FTRs bid in the FTR auction is the same as the average line flows, namely 155 MW FTR from node 2 to node 1 and, 65 MW FTR from node 3 to node 2, then the FTR market clearing problem

	(a)	(b)	Ave
f_{21}	160.0	150.0	155.0
f_{32}	80.0	50.0	65.0

Table 3: Flows

becomes.

$$\begin{aligned} \max \quad & 5q_{21} + 25q_{32} \\ \text{s.t.} \quad & d_1 = q_{21} \\ & q_2 = q_{21} - q_{32} \\ & q_3 = q_{32} \\ & q_2 + q_3 \leq 160 \\ & q_3 \leq 200 \\ & q_3 \leq 50 \\ & 0 \leq q_{21} \leq 155 \\ & 0 \leq q_{32} \leq 65 \end{aligned}$$

The optimal solution to this problem is given by

$$q_{21} = 155, \quad q_{32} = 50,$$

with FTR clearing prices of \$25.00 for the FTRs from node 3 to node 2 and \$0.00 for FTRs from node 2 to node 1. Note that this is a similar effect to that already discussed in the two node example in section 5.2. The average flow on the line from node 2 to node 1 is 155 MW which is strictly less than the capacity of that line (which does not vary in either contingency). Therefore, the price of FTRs for this line will clear at \$0.00, which is different from the average nodal price difference between nodes 2 and 1.

Deng et al. assert that when FTRs serve primarily as hedging instruments, bid quantities for FTRs tend to be close to expected transaction volumes and there are many point-to-point FTR pairs. Such a large range will have the effect (through the market clearing mechanism that embeds network effects) of imposing quantity limits on certain FTR awards which will cause the prices to deviate from the initial bid prices. They state that in more speculative markets where there are fewer FTR types and FTR quantities exceed hedging needs, the clearing prices are more likely to be close to efficient prices.

The following theorem demonstrates that although in the realistic sized examples provided by Deng et al. and in our simple illustrative examples the FTR auction clears inefficiently, there is incentive to correct this inefficiency through increasing the bid volume.

Theorem 1. *Consider the simultaneous feasibility FTR auction problem over any network where the bids for FTR prices are expected nodal price differences. Suppose that we*

- *solve the auction problem,*
- $(\forall i)$ *and* $(\forall j)$, *if* $\pi_j - \pi_i < \beta_{ij}$ *then raise* T_{ij} *by 1 MW and go to 1.*

Then we will stop in a finite number of iterations with the final clearing prices equal to the bid prices i.e. $\pi_j - \pi_i = \beta_{ij}$.

References

Biskas, P. N., N. P. Ziogos, and A. G. Bakirtzis. 2007. "Analysis of a monthly auction for financial transmission rights and flow-gate rights." *Electric Power Systems Research* 77:594–603.

Bushnell, J. B., and S. E. Stoft. 1996. "Electric Grid Investment Under a Contract Network Regime." *Journal of Regulatory Economics* 10:61–79.

Deng, S., S. Oren, and A. Meliopoulos. 2010. "The inherent inefficiency of simultaneously feasible financial transmission rights auctions." *Energy Economics* 32:779–785.

Hogan, W. W. 1992. "Contract networks for electric power transmission." *Journal of Regulatory Economics* 4 (3): 211–242.

Philpott, A., and G. Pritchard. 2004. "Financial transmission rights in convex pool markets." *Operations Research Letters* 32 (2): 109–113.

PJMeFTR. 2007. "PJM eFTR Users Guide." Downloadable from `http://www.epjmmarkets.com/documents/downloads/user-guides/eftr-user-guide-annual-ftr-auction.pdf`.

Siddiqui, A., E. S. Bartholomew, C. Marnay, and S. S. Oren. 2005. "On the Efficiency of the New York Independent System Operator Market for Transmission Conestion Contracts." *Managerial Finance* 31:1–45.

A Natural Gas LP Formulation to Enhance Allocation with Market Pricing Mechanisms

S.R. Starkey, E.G. Read and S. Dye
University of Canterbury
New Zealand.
E-mail: stephen.starkey@pg.canterbury.ac.nz
B.J. Ring
Market Reform,
Australia.
W. Pepper
ICF International,
USA.

Abstract

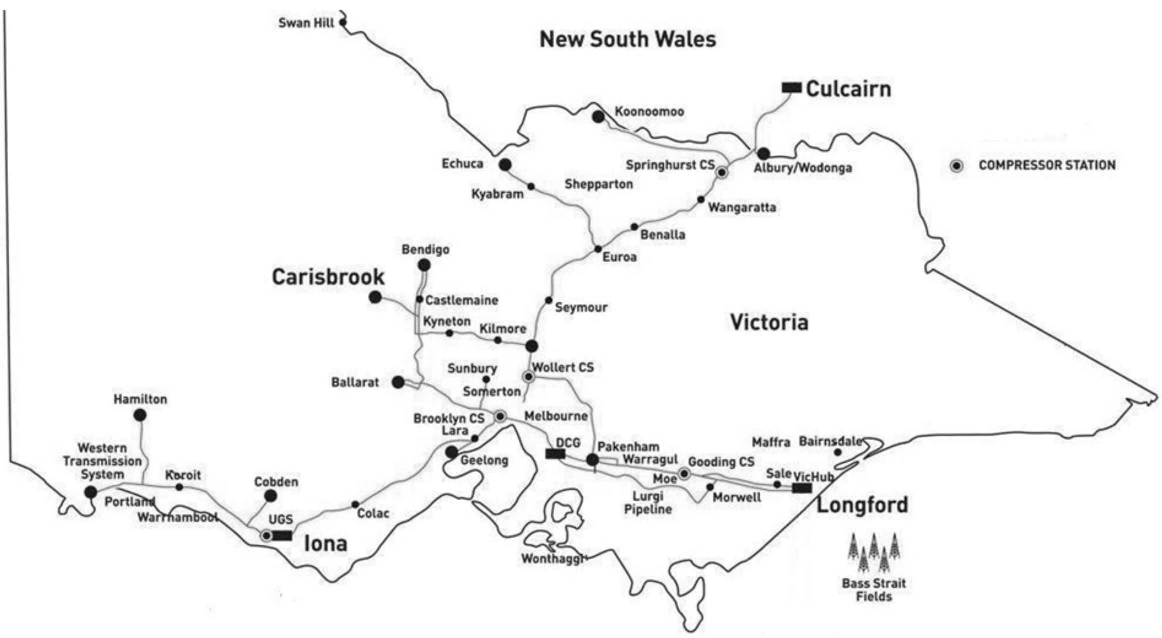
Many electricity markets are now cleared using a Linear Programming (LP) formulation that simultaneously determines an optimal dispatch, and corresponding nodal prices, for each market dispatch interval. Although natural gas markets have traditionally operated in a very different fashion, the same basic concept can be applied. Since 1999, the Australian state of Victoria has operated a gas market designed to operate in this fashion, based on an LP approximation to the underlying inter-temporal non-linear gas flow optimization problem. The simplified formulation presented here covers the key physical relationships defining the flow of gas though the system, along with definition of practical offer and bid forms. The dual variables on key constraints imply prices which vary by location, as for electricity markets, but also by time. But the gas flow equations mean that gas is both delayed and stored within the transportation system itself. This raises a number of operational, pricing, and hedging issues which could be ignored in the case of electricity, but become important when operating this kind of market in a gas supply network. Some of those issues will also be important for the design of markets for other commodities, such as water, where there are delays and storage within the "transportation system", over which the market operates.

Presentation

Outline

1. Victorian Natural Gas System
2. Electricity v Natural Gas
3. Modelling the Market
4. Modelling the Natural Gas Transportation System
5. Prices and their Interpretation
6. Conclusions

1. Victorian Natural Gas System



2. Electricity v Natural Gas

Both transfer bulk quantities over large distances via interconnected networks,

Each unit of the respective commodity seeks path of least resistance through the network,

Electricity transfers energy instantaneously through "on/off" network of lines,

Natural Gas transfers mass through pipes, with:

- Delays and storage in transit
- Greater control, via valves and compressors

3.1 Market Concept

Market Operator clears the market:

- Determining price for each node and period
- Issuing dispatch orders for all participants

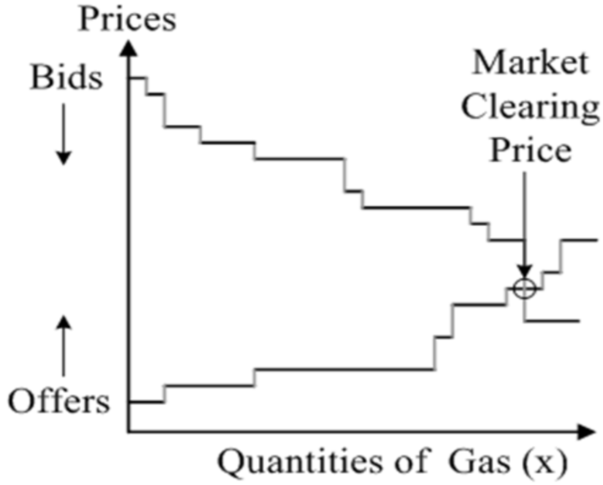
In OR terms, find the optimum solution, while balancing gas over all nodes and time periods, within constraints

Need to model:

- Market clearing at each node and period
- Gas flow dynamics between nodes and periods

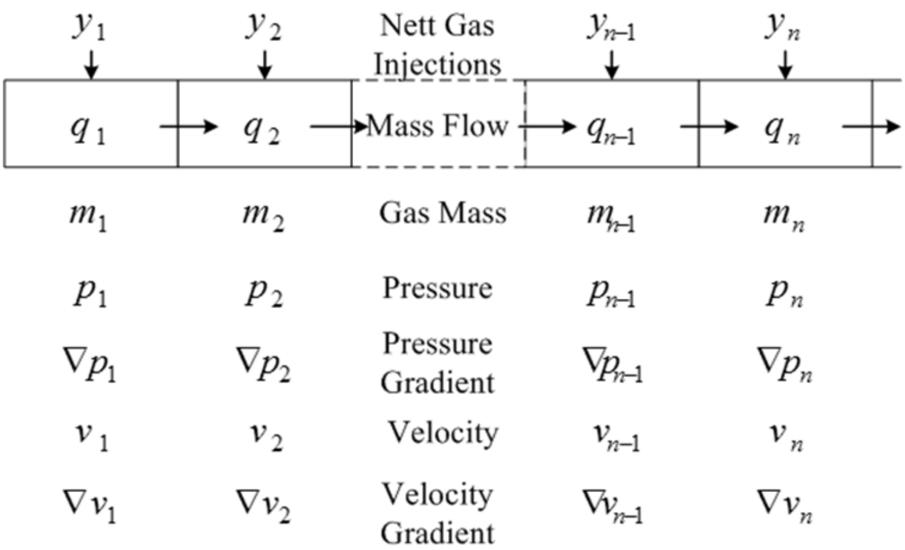
3.2 Market Clearing

$$\sum_{t}\sum_{n}\left(\sum_{d\in Dn}\sum_{i} Bid^{t}_{di}x^{t}_{di}-\sum_{s\in Sn}\sum_{i} Offer^{t}_{si}x^{t}_{si}\right)$$



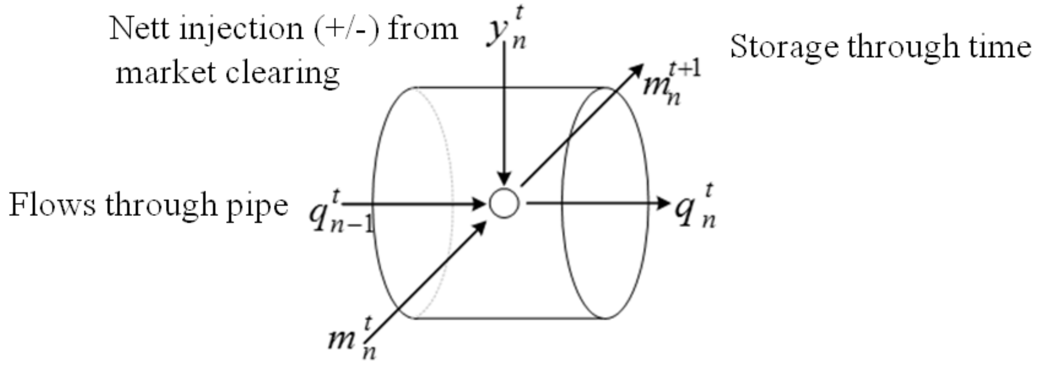
Plus constraints to bound and sum bid/offer tranches

4.1 Gas Flow Variables



All variables assumed to be at middle of cells

4.2 Mass Balance



$$m_n^{t+1} = m_n^t + y_n^t + q_{n-1}^t - q_n^t$$

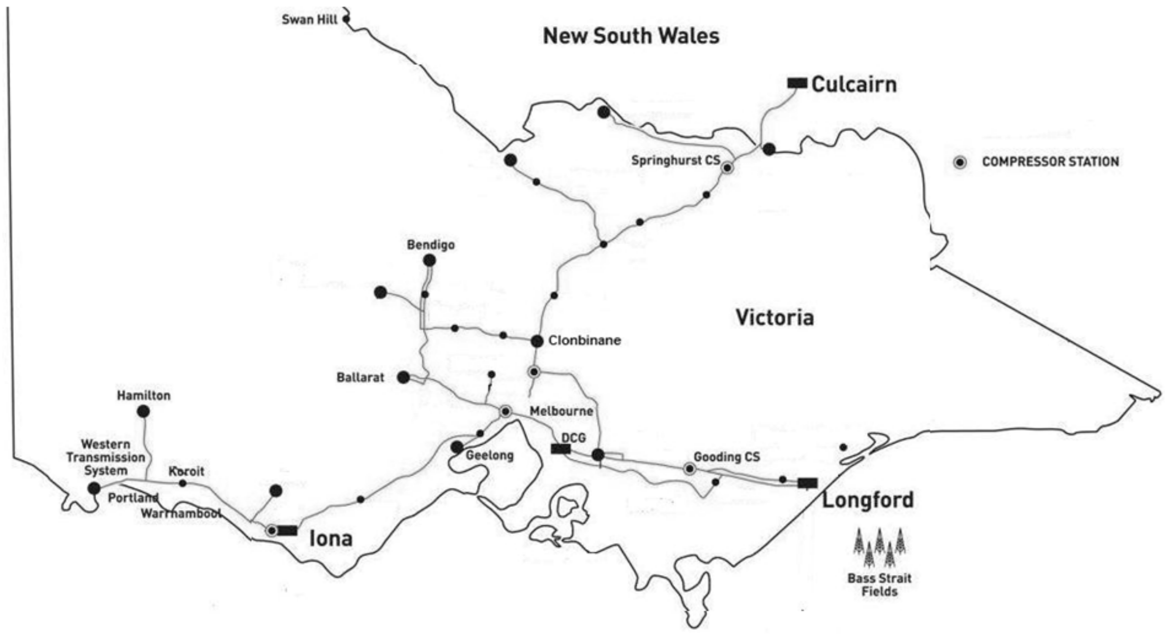
As for electricity networks:

- Except gas flows through time and space

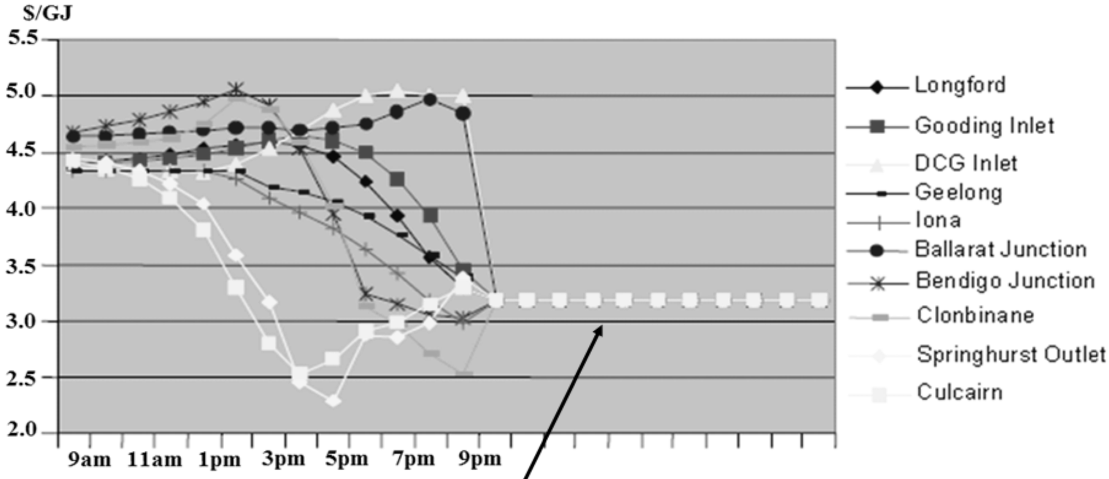
As for water networks:

- Except gas is compressible so storage and flow are both driven by pressurisation

5.1 Victorian System

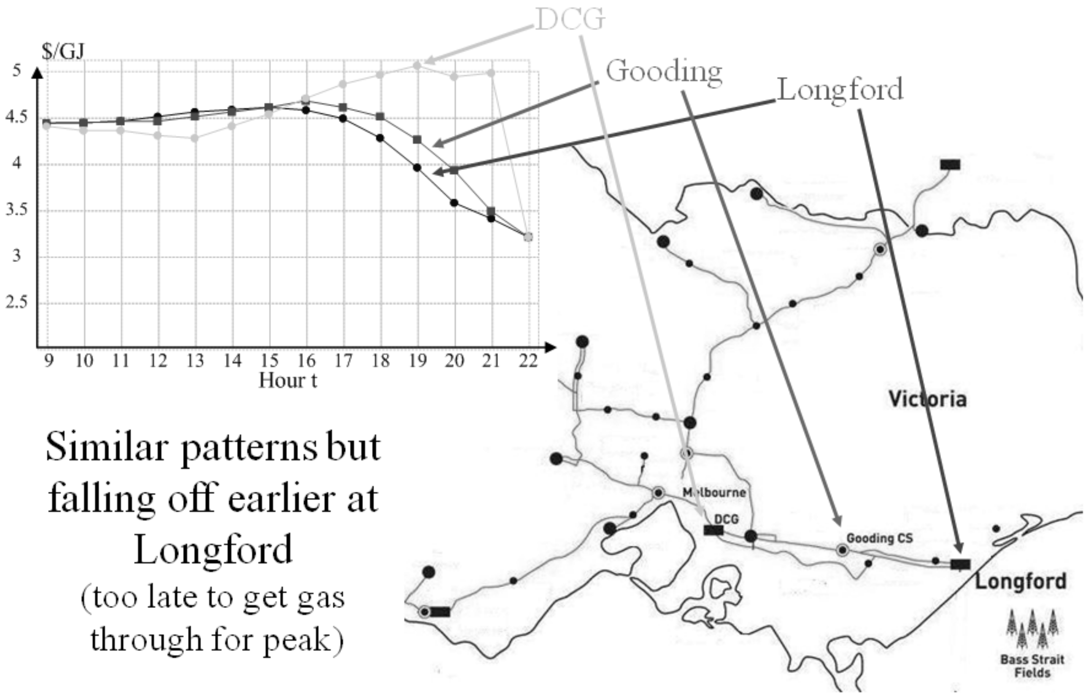


5.2 Price Scenario: General Relationships



Prices all equal because end-of-day gas assumed to be the same price everywhere, due to system "re-set"

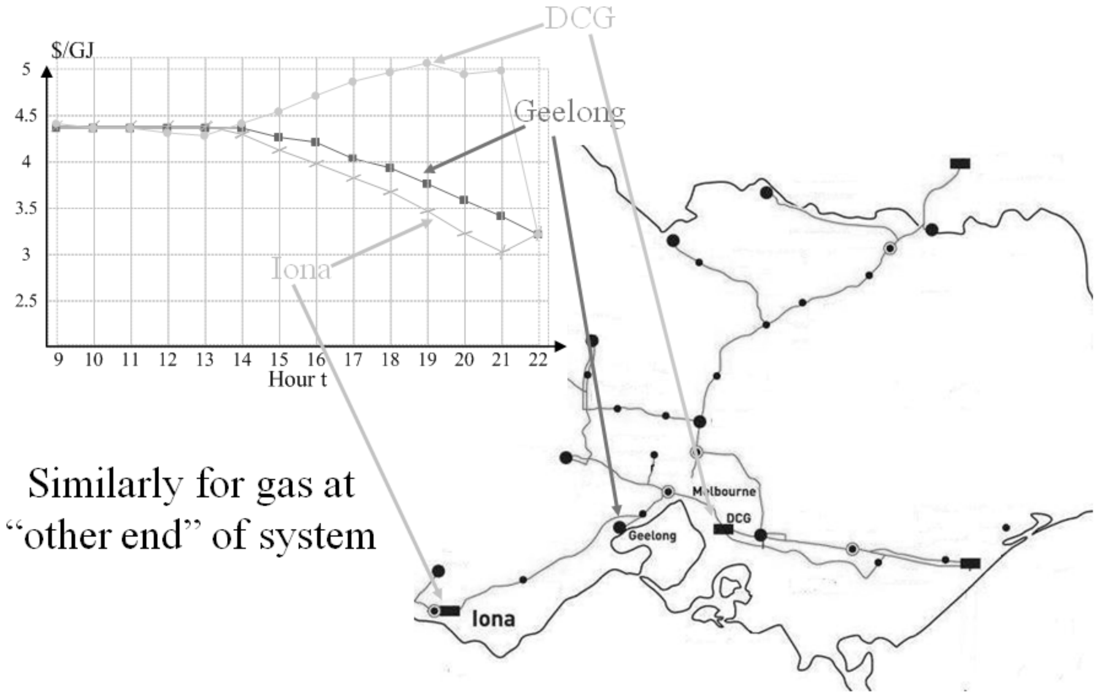
5.3 South-East Price Relationships



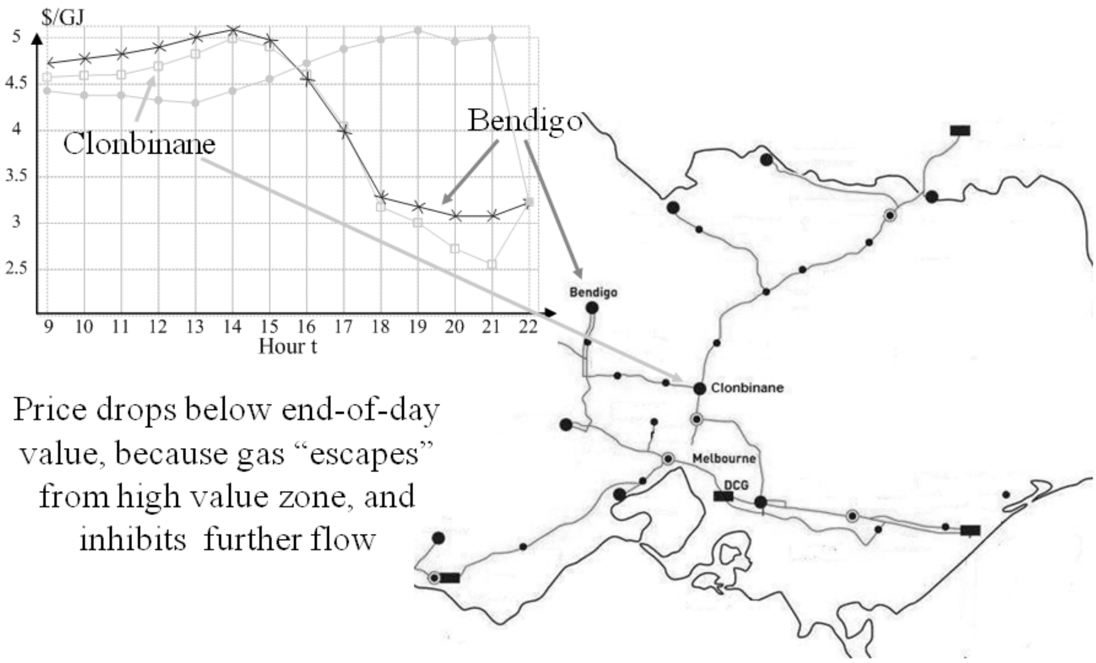
Similar patterns but falling off earlier at Longford

(too late to get gas through for peak)

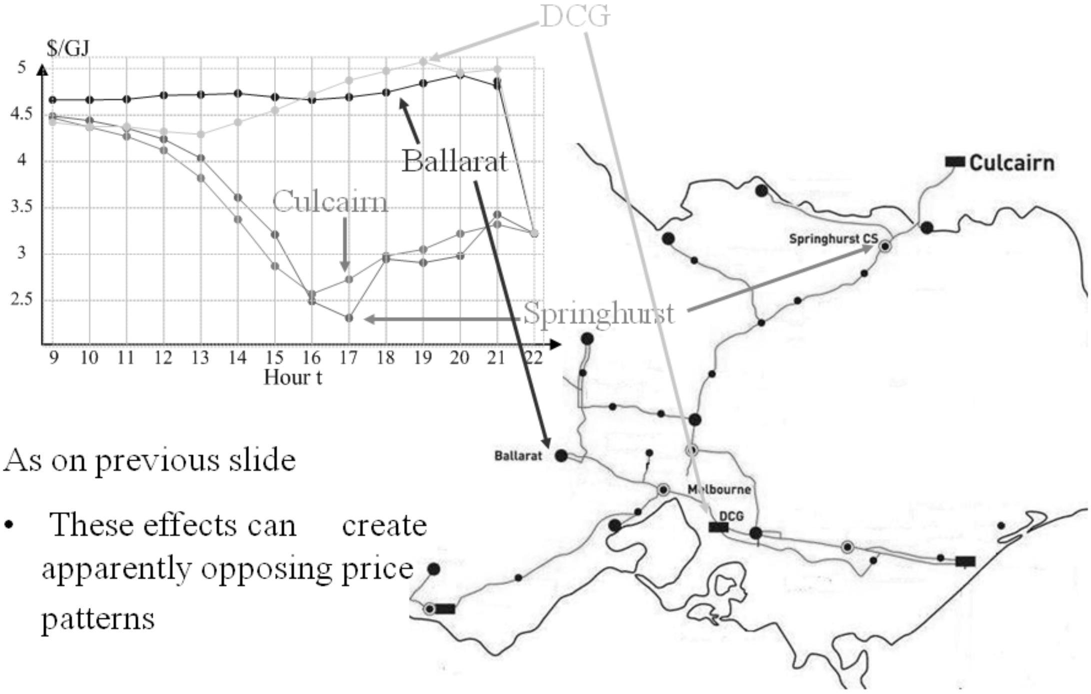
5.4 South-West Price Relationships



5.5 North-Central Price Relationships



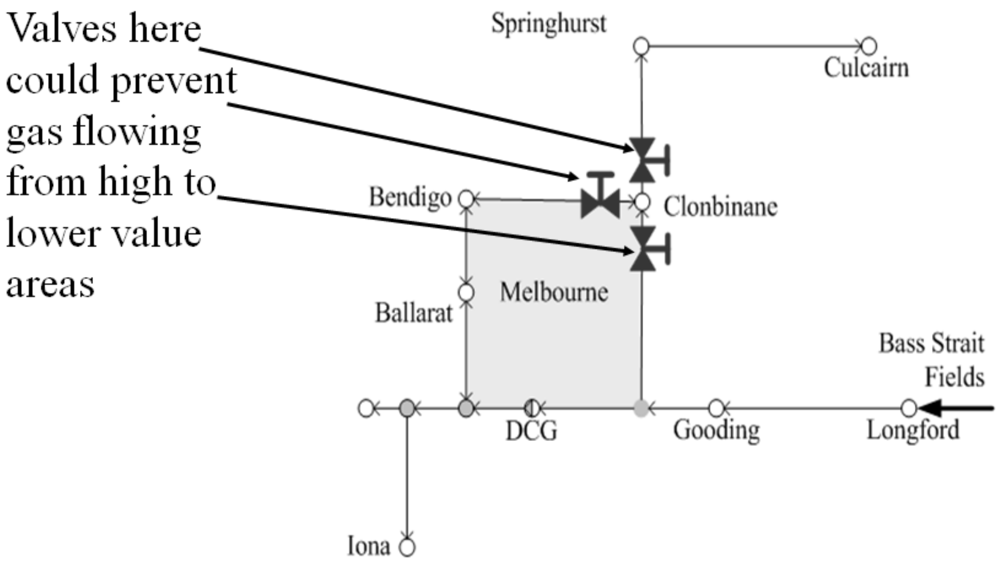
5.6 Central v North-East Price Profiles



As on previous slide

- These effects can create apparently opposing price patterns

5.7 Example Price Information Use



Price information can allude to enhancing system control to better manage flows as the gas transmission network evolves

6. Conclusions

For the gas market in Victoria:

- Congestion can generate significant price signals
- But it is rare enough that actual market is highly simplified to 1 node and 6 (re-)pricing periods
- Detailed model is used for operational dispatch though

For gas markets elsewhere:

- The LP formulation works, and can be applied
- Significant benefits likely where congestion occurs

For similar markets with system delay and storage:

- Transferable knowledge to other systems such as water

Outline of a Market for Ecological Connectivity

John F. Raffensperger
Department of Management, University of Canterbury,
Private Bag 4800, Christchurch 8140, NZ,
john.raffensperger@canterbury.ac.nz

Abstract

This paper presents some preliminary thinking about a possible market for ecological connectivity. A recent article in the journal *Nature* identified the world's biodiversity loss as far exceeding humanity's safe level. Other work has identified ecological connectivity – the ability of species to move across land – as critical for biodiversity.

The paper starts with a short literature review on ecological connectivity. This paper then explores one idea for using a market to improve ecological connectivity. The ecological value of connecting two regions can depend on whether regions are connected to other regions. Consequently, the problem will need to be solved with an integer program. This paper explores how such a combinatorial market might be operated.

The market should incentivise relevant parties, such as government agencies and farms, to implement specific changes to roads and other barriers to wildlife, to improve ecological connectivity within a given budget.

Finally, the paper briefly examines the rights associated with ecological connectivity. Ideally, users would trade with each other, rather than simply have an expensive government procurement. But this will require an enormous change in mindset, and the nature of the tradable rights is highly complicated.

Key words: Biodiversity, Ecological connectivity, Combinatorial auction, Smart market.

1 Intro and literature review on ecological connectivity

This paper gives some very early and preliminary thinking about a possible market for ecological connectivity, a critical support for biodiversity. The problem seems important enough. The United Nations declared 2010 to be the International Year of Biodiversity (CBD 2010). In a recent *Nature* article, Rockström *et al* (2009) identified the current rate of biodiversity loss as far exceeding humanity's safe level. "Today, the rate of extinction of species is estimated to be 100 to 1,000 times more than what could be considered natural. As with climate change, human activities are the main cause of the acceleration." They go on to describe how biodiversity loss erodes the biosphere's resilience, especially with climate change and disruptions to the nitrogen cycle.

In a recent *Science* article, Marton-Lefèvre (2010) writes, "…the International Union for Conservation of Nature documents the extinction risk of 47,677 species: 17,291 are threatened, including 12% of birds, 21% of mammals, 30% of amphibians, 27% of reef-building corals, and 35% of conifers and cycads." She points to cost estimates of this loss "between 1.35 and 3.1 trillion U.S. dollars." The Harvard entomologist EO Wilson (2004) estimated that half of all species could be extinct by 2050.

Biodiversity loss has many causes, including climate change, pollution, introduction of exotic species, commercial overharvesting, and conversion of natural habitats for human use (McCallum and Dobson, 2002). This paper focuses on the last of these. Perhaps surprisingly, the most ecological diversity is found not in forests or nearby savannahs, but rather in the ecological gradient – the *ecotone* – between them (Smith et al, 2005). However, humans criss-cross the environment with sharp lines, for aesthetics (such as building to the edge of waterways), for legal boundaries, and especially for transportation. Transportation networks break up regional ecologies into islands, and create barriers between the islands. In the few places where society has attempted to maintain a region with ecological health, the region has generally been identified by a uniform structure, such as a forest, rather than an ecotone (Smith et al, 2005). By constructing sharp boundaries through the middle of ecotones, in between the uniform islands (Figures 1 and 2), ecological diversity tends to be reduced.

Figure 1. Riccarton Bush, Christchurch, is an example of an ecological island.
Source: Google Maps, 4 Nov 2010.

Researchers have studied how ecologies are improved by connections, and have developed a variety of measures for ecological connectivity (Calabrese and Fagan, 2004; Hartig and Drechsler, 2008a; Kindlmann and Burel, 2008; Marulli and Mallarach, 2005; Moilanen and Nieminen, 2002; Nikolakaki, 2004; Schulte et al., 2006). The measures range from simple to elaborate, can depend on land characteristics, and can vary by species. The measures sometimes include the probability that a species survives.

Work has also been done on models to solve the connectivity problem (Duque et al., 2007; Hajkowicz et al., 2007; Haunert, 2007; Tóth et al., 2006). These models, sometimes called "regionalization models," are typically very hard integer programs, and can be difficult simply to formulate, much less to solve. A land parcel may be modelled as a node, and neighbour adjacency may be modelled as arcs. The regionalization problem, then, is to find a forest where each tree satisfies bounds on attributes, such as the probability that a species will survive. This paper recognizes that a connected set of arcs need not be a tree, but could be a donut shape, or even a nearly dense graph (in the two-dimensional plane), but the general problem is almost the same.

Figure 2. Roads cause ecological fragmentation in Indiana Dunes National Park, USA.
Source: Wikipedia, 4 Nov 2010.

Faith et al (2003) expressed pessimism about the regionalization models, pointing out that the solutions have never been implemented. They propose “policy algorithms,” and gave as an example a simplistic auction. The auction would be funded by a central source, so it would be a procurement. The authors develop an elaborate framework for quantifying biodiversity.

Hartig and Drechsler (2009) made a case for using spatial incentives. Jack et al (2008) gave an overview of payment systems, focusing mainly on government procurements and the developing world. They point out that traditional conservation payments can result in weak results with uncertain outcomes, and perverse behaviours where users stay in business simply to collect payments.

A particularly interesting market approach is Nemes et al (2008), who apparently have actually implemented a market to incentivize cultivation of native vegetation in Australia. They defined a “habitat score” based on the quality of the area. Trades must satisfy spatial and quality restrictions. The market trades contracts, which commit landowners to manage native vegetation for a specified period of time, after which the site is protected permanently. Contracts are held in custody by government; government’s role is limited to facilitating the market, and designing and monitoring contracts. Buyers pay for monitoring and compliance. This market design addresses the several issues, such as the temptation of landowners to avoid contractual obligations, the temporary reduction in quality when exotic vegetation is cleared, and the probabilistic success of native vegetation.

While the market of Nemes et al (2008) incentivises cultivation of native plants, it omits connectivity. Trade is multilateral rather than a procurement. Reeson et al (2008) considered connectivity, but as a procurement. Those authors observed a significant coordination problem, in which connectivity requires neighbours to work together. They also observed the potential for gaming, where a land owner in the middle of a potential corridor could raise their price to extract extra gains well above the owner's cost. In a lab setting, the authors found that an iterated (multi-round) auction allowed participants to work around others' gaming behaviours.

2 An outline of a market for eco-connectivity

This paper attempts to develop a "policy algorithm" as Faith et al (2003) seek, while addressing the lack of spatial connectivity in Nemes et al (2008). The goal is a multilateral market, not a procurement (or not only a procurement), that will incentivize the spatial solutions. In this market, government desires to maintain or increase the total ecological connectivity within a given region.

Buyers may be landowners with insufficient eco-connectivity rights to their own property (possibly lost through some process such as eminent domain), or they may be landowners who wish to reduce the eco-connectivity of their property. The government may also be a buyer, seeking to raise the eco-connectivity of a given region. Sellers would be land owners who offer to improve the eco-connectivity of their land above its current status.

To help buyers avoid sellers who may have hold-out power (such as in the middle of a developing corridor), all bids would be visible to all participants, as in Nemes et al (2008). The market could suffer moral hazard, where a contract appears attractive but is actually difficult to implement. To remedy this, the market manager may require sell bids to be vetted by an ecologist and a contract manager. As in Nemes et al (2008), sellers pay for this process (e.g., for site visits).

Given the set of bids, the market manager would enumerate sets of connected edges, $t = 1, \ldots, T$, each with ecological value $V_{s,t}$ for species s. Each edge set is associated with a set of bids which satisfy constraint sets 2 and 3 below. The model could then be cleared by Model EcoConnect below.

In this model, eco-connectivity is recognized explicitly as two-way. The owner of parcel i may attempt to increase eco-connectivity towards parcel j, but the eco-connectivity between the two also depends on owner j's effort (or existing status) of eco-connectivity toward parcel i.

Indices

b, contract.

$i, j = 1,\ldots,I$, owner, assumed to be in one-to-one correspondence with land parcels; owner i owns parcel i.

$(i, j) \in Edges$, indicating the adjacency of parcels i and j.

$s = 1,\ldots, S$, species

$t = 1,\ldots,T$, enumerated sets of connected edges.

Parameters

$A_{i,j,t} = 1$ if edge (i,j) is part of set t, else 0.

Budget = maximum amount that the market manager is willing to pay.

$C_{s,i,j,b}$ = increase in directed eco-connectivity $i \rightarrow j$ obtained for species s, if the market manager accepts contract b from user i. $C_{s,i,j,b} > 0$ for sell bids and $C_{s,i,j,b} < 0$ for buy bids.

$K_{s,i,j}$ = previously recognized eco-connectivity for parcel i, to connect to parcel j, for species s.

$BuyPrice_{i,j,b}$ = price on the buy bid from user i, to disconnect to parcel j, contract b.

$SellPrice_{i,j,b}$ = price on the sell bid from user i, to connect to parcel j, contract b.

$Buyset_i$ = user i's constraints on their bids (e.g., if $buybid_{i,j,b} = 1$, then $buybid_{i,k,b} = 1$).

$Sellset_i$ = user i's constraints on their bids (e.g., if $sellbid_{i,j,b} = 1$, then $sellbid_{i,k,b} = 1$).

T_s = overall ecological connectivity target for species s.

$V_{s,t}$ = ecological value of set t for species s.

Variables

$sellbid_{i,j,b}$ = 1 if the market manager accepts sell bid from user i, to connect parcel i to parcel j, contract b, else 0.

$buybid_{i,j,b}$ = 1 if the market manager accepts buy bid from user i, to disconnect parcel i to parcel j, contract b, else 0.

$z_{s,i,j}$ = undirected connectedness of edge (i, j) for species s. This could be a percentage, where $0 \leq z_{s,i,j} \leq 1$, but is assumed more general here.

$y_{s,t}$ = 1 if set t for species s is selected, else 0.

Model EcoConnect

1. Maximize $\sum_{(i,j)}\sum_b (BuyPrice_{i,j,b}\ buybid_{i,j,b} - SellPrice_{i,j,b}\ sellbid_{i,j,b})$.
2. $sellbid_{i,j,b} \in Sellset_i$, for all $(i,j) \in Edges$, and all contracts b,
3. $buybid_{i,j,b} \in Buyset_i$, for all $(i,j) \in Edges$, and all contracts b.
4. $z_{s,i,j} \leq \sum_b (C_{s,i,j,b}\ buybid_{i,j,b} + C_{s,i,j,b}\ sellbid_{i,j,b}) + K_{s,i,j}$ for all $(i,j) \in Edges$, and all species s,
5. $z_{s,i,j} \leq \sum_b (C_{s,j,i,b}\ buybid_{j,i,b} + C_{s,j,i,b}\ sellbid_{j,i,b}) + K_{s,j,i}$ for all $(i,j) \in Edges$, and all species s.
6. $y_{s,t} \leq A_{i,j,t}\ z_{s,i,j}$ for all relevant s, i, j, t.
7. $\sum_t A_{i,j,t}\ y_{s,t} \leq 1$ for all $(i,j) \in Edges$.
8. $\sum_t V_{s,t}\ y_{s,t} \geq T_s$ for each species s.
9. $-\sum_{(i,j)}\sum_b (BuyPrice_{i,j,b}\ buybid_{i,j,b} - SellPrice_{i,j,b}\ sellbid_{i,j,b}) \leq Budget$.
10. $buybid_{i,j,b}, sellbid_{i,j,b}, y_{s,t} \in \{0,1\}, z_{s,i,j} \geq 0$.

Explanation

The objective (1) maximizes the buyer and seller surplus.

Constraint sets 2 and 3 allow traders to specify restrictions on their bids. For example, a trader may wish that any two of three bids must be taken together, to ensure some economy of scale in connecting to more than one neighbour.

Constraint sets 4 and 5 requires that connectivity be both ways.

Constraint set 6 allows set t only if all bids for set t are selected.

Constraint set 7 requires that at most one tree with edge (i, j) can be selected. Otherwise, the connectivity of a given edge could be counted more than once.

Constraint set 8 requires that target connectivity be met for each species.

Constraint set 9 is a budget constraint that may be imposed by the market manager.

3 Discussion

3.1 Initial rights

To obtain an initial right $K_{s,j,i}$, owner i should be able to bring a project to government for approval, presumably at the owner's expense. This would not require a bid in the market. To make this process easier for land owners, government could publish a list of best management practices which government would accept as increasing eco-connectivity to some initial right $K_{s,j,i}$.

3.2 Poor definition of rights

Ideally, the market would allow a land owner in one area of the region to buy "connectivity" in another area, thereby allowing the increase in the seller's connectivity to offset a loss of connectivity on the buyer's property. To make this work, the buyer must understand exactly how his or her land will change in regard to connectivity, and also exactly the nature of the connectivity purchased. Further, the market manager would have to recognize that the buyer would now hold a right to connectivity, though the buyer's own property lacked it.

The nature of the right is ill-defined here because the measure of connectivity is ill-defined. (That it is multi-dimensional due to many species makes the rights more complicated, but in principle still manageable.) Given the poor agreement on connectivity measures in the ecological literature, society's ability to find agreement seems even less likely. One solution is to raise funds through taxes, and run the market as a procurement with no buy bids. The problem with a tax-and-procure approach is that it provides no obvious way to require a particular land owner to improve their land's connectivity. Furthermore, any land owner could hold out for an arbitrarily large amount of money, with sufficiently high targets T_s.

3.3 Neighbour coordination

Through constraint sets 4 and 5, the proposed market partially addresses the coordination problem observed by Reeson et al (2008), by explicitly managing two-way connectivity. But this is only a partial attack on the problem, as neighbours must still offer more or less matching bids in the same auction.

Because the market manager must approve contracts, the manager can point out to each bidder i which neighbour j would be of particular interest for coordination. This process could be assisted by the process of edge set enumeration, as the market manager would be able to determine which edges are likely to allow bidder i to join a satisfactory set.

3.4 Individual property requirements

Suppose that some measure can be agreed upon, or that government is willing to impose whatever measure it has chosen. Instead of tax-and-procure, government could require that every land parcel i were part of some local set satisfying a given target T_s. That is, the government tells land owner i, "Your land must be part of some subset with total connectivity T_s" without specifying exactly which subset. This corresponds to dropping the summation on t in constraint set 8: $V_{s,t}y_t \geq T_s$ for each species s. This requirement is analogous to zoning, such as when government allows a given land use in a commercial district, but not in a residential district. If the area were rezoned, all land use in the area would have to satisfy the new requirements.

In this case, some owner *i* can buy while another land owner *k* sells, but owners *i* and *k* must be part of the same connected set *t* for set *t* to meet its target; the rights are accrued to all land owners in the set. Suppose further, that some time after this market, one owner *i* wanted to reduce the connectivity across his or her land. To remain part of a satisfactory set, owner *i* would have to pay some other owners (other than *k*) to increase the connectivity across their land. The reduction of connectivity across owner *i*'s land could still affect the previous owner *k*, especially if owner *k* had previously paid for some other owner *l* to improve the connectivity of *k*'s edge set. In that case, the market manager initializes the parameter $K_{s,k,m}$, indicating that owner *k* has previously obtained sufficient connectivity. When owner *i* attempts to buy, feasibility requires that owner *k* is still in a feasible set, even though owner *k* did not participate in the market; the responsibility is on owner *i* to offer a sufficient amount to offset the loss of connectivity on parcel *i*, even as it affects other parcels.

Furthermore, if owner *i* buys, reducing connectivity from *i* to *j*, now the existing connectivity from *j* to *i* is lost as well. To offset the loss of connectivity in both directions between *i* and *j*, owner *i* will therefore have to pay a fairly high price.

3.5 Revenue adequacy

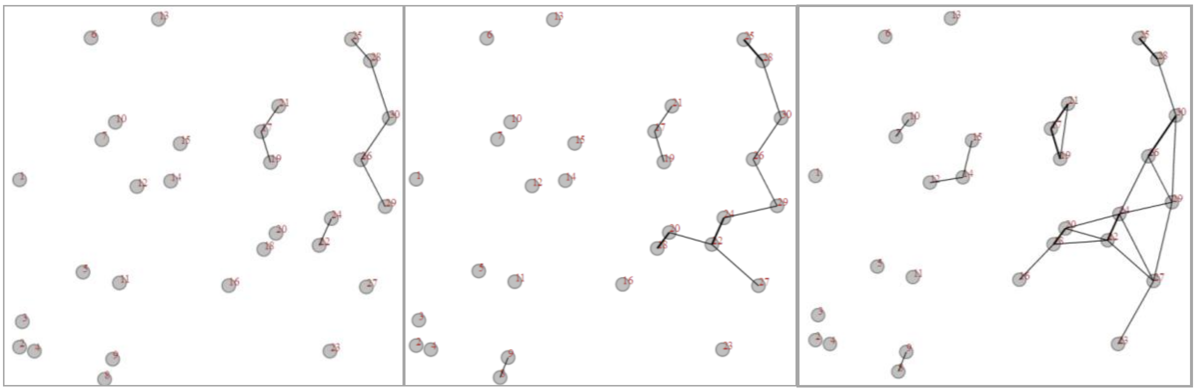
Net revenue to the market manager is unlikely to be exactly zero because bids are discrete. Nemes et al (2008) solve this problem by allowing any buyer or seller to search for feasible bid sets, and then to clear those bids as a market maker; any positive revenue then accrues to the market maker who found the feasible bid set. This gives an incentive for bidders to bid reasonably themselves as well, and this reduces the incentive for blocking. Such incentive could be partially done here, where every set would be required to be revenue adequate, with the excess revenue returned to land owners following some formula.

Alternatively, the government could choose to maintain a running budget. This running budget could have occasional or periodic injections from government. Or the market manager could be required to maintain market revenue between some negative lower limit and some positive upper limit, nearly balancing over time. The extent to which government pays land owners to improve ecological connectivity will likely be subject to political power.

4 Trivial example

This trivial hypothetical example assumes one species. Thirty land owners offer to sell to government, which pays for all work. For simplicity, the value of eco-connectivity is lowest with the lowest numbered land owners (more urban), and increase with node number (more rural). It was solved with a simpler model than EcoConnect.

Figure 3 shows how the solution changes with an increasing target. Note that the market manager may choose to increase connectivity within an already-connected region. In this case, the eco-connectivity is somewhat linear in the edges, so no extra gain is made by combining two small sets into one larger set beyond the one arc.



Target T = 15. Cost 34.73. T = 30. Cost \$73.58 T = 60, cost \$167.23.

Figure 3. With an increasing budget, connectivity may increase within existing connected areas. Edge thickness indicates degree of connectivity.

References

Calabrese, J.M., Fagan, W.F., "A comparison-shopper's guide to connectivity metrics," *Frontiers in Ecology and the Environment* 2, 529-536, 2004.

Convention on Biological Diversity (CBD), www.cbd.int/2010/welcome, accessed 4 Nov 2010.

Duque, J.C., Ramos, R., Suriñach, J., "Supervised regionalization methods: a survey," *International Regional Science Review* 30, 195, 2007.

Faith, D.P., Carter, G., Cassis, G., Ferrier, S., Wilkie, L., "Complementarity, biodiversity viability analysis, and policy-based algorithms for conservation," *Environmental Science & Policy* 6, 311-328, 2003.

Hajkowicz, S., Higgins, A., Williams, K., Faith, D.P., Burton, M., "Optimisation and the selection of conservation contracts," *Australian J. of Ag & Res Econ* 51, 39-56, 2007.

Hartig, F., Drechsler, M., a, "Stay by thy neighbor? Structure formation, coordination and costs in tradable permit markets with spatial trading rules," http://www.ucl.ac.uk/bioecon/10th_2008/14.Drechsler.pdf, accessed 8 Nov 2010.

Hartig, F., Drechsler, M., "Smart spatial incentives for market-based conservation," *Biological Conservation* 142, 779-788, 2009.

Haunert, J.-H., "Optimization methods for area aggregation in land cover maps," *Proc. of 10th ICA Workshop on Generalisation and Multiple Representation*, Moscow, Russia, 2007.

Jack, B.K., Kousky, C., Sims, K.R.E., "Designing payments for ecosystem services: lessons from previous experience with incentive-based mechanisms," *Proceedings of the National Academy of Sciences* 105, 9465-9470, 2008.

Kindlmann, P., Burel, F., "Connectivity measures: a review," *Landscape Ecology* 23, 879-890, 2008.

Marton-Lefèvre, Julia, "Biodiversity Is Our Life," *Science*, 5 Mar 2010, v327, n5970, p1179.

Marulli, J., Mallarach, J.M., "A GIS methodology for assessing ecological connectivity: application to the Barcelona Metropolitan Area," *Landscape and Urban Planning* 71, 243-262, 2005.

McCallum, H., Dobson, A., "Disease, habitat fragmentation and conservation," *Proceedings of the Royal Society B: Biological Sciences* v269, pp. 2041-2049, 2002.

Moilanen, A., Nieminen, M., "Simple connectivity measures in spatial ecology," *Ecology* v.83, pp 1131-1145, 2002.

Nemes, V., Plott, C.R., Stoneham, G., "Electronic BushBroker Exchange: Designing a Combinatorial Double Auction for Native Vegetation Offsets," 2008 http://www.marketbasedinstruments.gov.au/MBIsinaction/Nationalprograms/NationalMBIPilotProgram/Round2/tabid/71/Default.aspx, accessed 8 Nov 2010.

Nikolakaki, P., 2004. A GIS site-selection process for habitat creation: estimating connectivity of habitat patches. Landscape and Urban Planning 68, 77-94.

Reeson, A., Rodriguez, L., Whitten, S., Williams, K., Nolles, K., Windle, J., Rolfe, J., 2008. "Applying competitive tenders for the provision of ecosystem services at the landscape scale," www.csiro.au/resources/LandscapeTendersPaper.html, accessed 8 Nov 2010.

Rockström, Johan, et al, "A safe operating space for humanity," *Nature*, v.46, 24 Sep 2009, pp. 472-5.

Schulte, L.A., Mitchell, R.J., Hunter, M.L., Franklin, J.F., Kevin McIntyre, R., Palik, B.J., "Evaluating the conceptual tools for forest biodiversity conservation and their implementation in the US," *Forest Ecology and Management* v232, pp1-11, 2006.

Smith, T. B., R. K. Wayne, D. Ginnan, and M. W. Bruford, "Evaluating the divergence-with-gene-f1ow model in natural populations: the importance of ecotones in rainforest speciation," pp. 148-165, in E. Bermingham, CW. Dick, and C. Moritz (Eds.). *Tropical Rainforests: Past, Present, and Future*. Univ. of Chicago Press, Chicago & London. 2005.

Tóth, S.F., McDill, M.E., Rebain, S., "Finding the efficient frontier of a bi-criteria, spatially explicit, harvest scheduling problem," *Forest Science* v. 52, pp. 93-107, 2006

Wilson, E.O., *The Future of Life*, New York: Alfred A. Knopf, 2002.