



**Operational Research Society
of New Zealand**

**Proceedings of the
41st Annual Conference**

ORSNZ'06

November 30th – December 1st 2006

**University of Canterbury
Christchurch
New Zealand**

SPONSORS

We are most grateful for the financial support we have received from CRA, Transpower, the Electricity Commission, Orion New Zealand Limited and Hoare Research Software. We also acknowledge the generous support we received from the Department of Management and College of Business and Economics, University of Canterbury



PREFACE

The papers in this volume form the Proceedings of ORSNZ06, the 41st Annual Conference of the Operational Research Society of New Zealand (ORSNZ), held November 30 to December 1 at the University of Canterbury, Christchurch New Zealand.

As chair of the Conference committee it is a pleasure to see such a strong collection of papers and, especially, such a large group of students attending and presenting papers.

The Conference Committee would like to thank the sponsors Charles Rivers Associates, Transpower, the Electricity Commission, Orion New Zealand Limited and Hoare Research Software as well as the College of Business and Economics and University of Canterbury for their support.

The conference could not have been possible without the invaluable assistance of the Conference Committee:

Shane Dye (Chair)

Ross James (Proceedings Editor)

John George (YPP Coordinator)

E. Grant Read

Nicola Petty

Terri Green

John (Fritz) Raffensperger

John Giffin

Don McNickle

Pulakanam Venkateswarlu

Pavel Catska

Shane Dye

TABLE OF CONTENTS

THURSDAY

8:00 – 8:45 Breakfast	Coppertop
8:45 – 9:00 Opening and Welcome	Coppertop
9:00 – 9:45 T1 Plenary Address	Coppertop
Andy Philpott, “Optimization, Uncertainty and Equilibria”	1

10:00-11:00 T2A Environmental Applications	Comm 002
---	-----------------

Chair Terri Green

David Wood, Andrew Ball, Brent Gilpin, Wendy Gregory, Jeff Foote, and Marion Savill, “Modelling microbes: simulating farm management options”	3
Stuart Mitchell, “Yield Frontier Analysis of Forest Inventory”	7
John Raffensperger, “A tutorial on hydrogeological optimisation”	9

10:00-11:00 T2B Optimisation Applications	Comm 009
--	-----------------

Chair Ross James

Andrew Mason, “Faster Map Matching for Emergency Vehicle Trip Analysis”	19
Ziming Guan, “Dynamic Outer-Approximation Sampling Algorithm”	29
Golbon Zakeri, “How to set optimal line tariffs”	37

11:00-11:30 Morning Tea

11:30-12:45 T3 Young Practitioner Session 1	Comm 002
--	-----------------

Chair Grant Read

Sarah Marshall, “Farthest Insertion Heuristics for the Freeze-Tag Problem”	39
Richard Lusby, “Routing Trains Through Railway Junctions: A New Set Packing Approach”	49
Bronwyn Erasmuson, “On the Mean Cumulative Function of Censored Warranty Data”	61
Vitesh Bava, “Evaluating the Performance of Radiotherapy Design Models”	71

12:45-13:45 Lunch	Coppertop
--------------------------	------------------

13:45-15:15 T4 Young Practitioner Session 2	Comm 002
--	-----------------

Chair Don McNickle

Michael Frankovich, “Computational Models for Large Airline Network Revenue Management Problems”	81
Amir Joshan, “Analysis of network strategic decisions for airlines”	91
Andrea Raith, “A Comparison of Solution Strategies for Biobjective Shortest Path Problems”	101
Peter Ebdon, “Faster Shortest Path algorithms for Siren”	113
David Richards, “A Study of Optimised Ambulance Redeployment Strategies”	123

15:15-15:45 Afternoon Tea

15:45-17:15 T5 Young Practitioner Session 3 **Comm 002**

Chair Nicola Petty

Anthony Downward, “Competition Benefits of Line Capacity Expansions in Electricity Markets”	133
Stuart Donovan, “An improved mixed integer programming model for wind farm layout optimization”	143
David Craigie, “Peak Shaving and Price Saving: Algorithms for Consumer Generation”	153
Julie Jang, “Scheduling Product Pairs subject to Changeover Times and Mould Constraints”	163
Samuel Gordon, “Rogaining: a Prize-Collecting Orienteering Problem”	173

17:15-18:15 ORSNZ AGM **Comm 002**

18:15-18:45 ORSNZ Council Meeting **Comm 002**

18:30-19:30 Pre-dinner Drinks **Mona Vale**

19:30- Conference Banquet **Mona Vale**

FRIDAY

8:15-9:15 Breakfast **Coppertop**

9:15-10:00 F1 Plenary Address

Alan Stenger, “Reflections on Forty Years of Researching and Implementing Inventory Management Systems in Commercial Firms”	183
---	-----

10:00-10:30 Morning Tea

10:30-11:50 F2A Energy **Comm 002**

Chair Andy Philpott

Bhujanga Chakrabarti, “Pricing for Variations in Large Loads and Wind Generations”	193
Vladimir Krichtal, “National Instantaneous Reserve Market in the New Zealand Wholesale Electricity Market”	197
Grant Read and Deb Chattopadhyay, “Risk-Adjusted Discount Rates and Optimal Plant Mix: A New Formulation for Electricity Market Optimisation”	201
Andrew Kerr, “A mass-balance gas simulation model for assessing the benefits of gas network augmentation options”	203

10:30-11:50 F2B Reliability, Quality and Forecasting **Comm 009**

Chair Don McNickle

Alex Ruiz-Torres, Carey McCleskey, Kazuo Nakatani, Arun Pennathur, Russell Rhodes, Edgar Zapata, and Jianmei Zhang, “Reliability Based Assessment Model for Space Vehicles”	205
Dinu Corbu, Stefanka Chukova and Jason O’Sullivan, “Application of Two-Dimensional Renewal Processes in Modelling Product Warranties”	215
John Paynter and Gabrielle Peko, “Power to the people - improving the quality of information in the census.”	225
Fernando Beltrán, Lina María Gómez and Pablo Maya, “Forecasting telecommunications demand for services with short-history data”	235

11:50-13:00 Lunch **Coppertop**

13:00-14:20 F3A OR in Action **Comm 002**

Chair Terri Green

Nicola Petty, “If I'm doing it, it must be O.R—An example of the use of Operations Research in evaluating the effectiveness of Special Education provision”	245
Lizhen Shao, “Finding Representative Nondominated Points in Multiobjective Radiotherapy Planning”	249
Babul Hasan and John Raffensperger, “Decomposition and Pricing Methods for solving MILP Model for an Integrated Fishery”	261
John Paynter, Jim Sheffield, David Sundaram and Dan Trietsch, “Quality vs. Power? In defence of academic values - Community OR in action”	273

13:00-14:20 F3B Supply Chain and Scheduling **Comm 009**

Chair Alan Stenger

Kenneth Baker and Dan Trietsch, “Safe Scheduling”	283
Alex Ruiz-Torres, Francisco Lopez and Johnny Ho, “Minimizing Average Tardiness and Machine Outsourcing Cost”	293
Tillmann Böhme, Paul Childerhouse, James Corner, Ron Garland, and Richard Varey, “Power and Dependency Barriers to Supplier Integration: A New Zealand Case Investigation”	303
Natashia Boland, Irina Dumitrescu and Gary Froyland, “A New Aggregation-Based Method with Improved Processing Selectivity for the Open Pit Mine Production Scheduling Problem”	313

14:20-14:50 Afternoon Tea

14:50-15:50 F4A OR in Sport **Comm 002**

Chair John Giffin

Stephen Clarke, “SPORTSBET21: a successful application of statistical modelling”	315
Mark Johnston, “The Travelling Tournament Problem: Neighbourhoods and Visualisation”	321
Hamish Waterer, K. Chang and D. Ryan, “Tournament Construction Methods for Auckland Bowls”	331

14:50-15:50 F4B Scheduling and Location

Comm 009

Chair John Raffensperger

David Ryan, "Aluminium Production Scheduling Revisited"	333
Oliver Weide, "An Iterative Approach to Airline Scheduling"	335
Wang Hsaio-Fan and Ying-Yen Chen, "Solving a Multi-level Capacitated Facility Location Problem by DVAM"	347

AUTHOR INDEX

Author	Title	Page	Session
Kenneth Baker	Safe Scheduling	283	F3B
Andrew Ball	Modelling microbes: simulating farm management options	3	T2A
Vitesh Bava	Evaluating the Performance of Radiotherapy Design Models	71	T3
Fernando Beltrán	Forecasting telecommunications demand for services with short-history data	235	F2B
Tillmann Böhme	Power and Dependency Barriers to Supplier Integration: A New Zealand Case Investigation	303	F3B
Natashia Boland	A New Aggregation-Based Method with Improved Processing Selectivity for the Open Pit Mine Production Scheduling Problem	313	F3B
Bhujanga Chakrabarti	Pricing for Variations in Large Loads and Wind Generations	193	F2A
K. Chang	Tournament Construction Methods for Auckland Bowls	331	F4A
Deb Chattopadhyay	Risk-Adjusted Discount Rates and Optimal Plant Mix: A New Formulation for Electricity Market Optimisation	201	F2A
Ying-Yen Chen	Solving a Multi-level Capacitated Facility Location Problem by DVAM	347	F4B
Paul Childerhouse	A New Aggregation-Based Method with Improved Processing Selectivity for the Open Pit Mine Production Scheduling Problem	313	F3B
Stefanka Chukova	Application of Two-Dimensional Renewal Processes in Modelling Product Warranties	215	F2B
Stephen Clarke	SPORTSBET21: a successful application of statistical modelling	315	F4A
Dinu Corbu	Application of Two-Dimensional Renewal Processes in Modelling Product Warranties	215	F2B
James Corner	A New Aggregation-Based Method with Improved Processing Selectivity for the Open Pit Mine Production Scheduling Problem	313	F3B
David Craigie	Peak Shaving and Price Saving: Algorithms for Consumer Generation	153	T5
Stuart Donovan	An improved mixed integer programming model for wind farm layout optimization	143	T5
Anthony Downward	Competition Benefits of Line Capacity Expansions in Electricity Markets	133	T5
Irina Dumitrescu	A New Aggregation-Based Method with Improved Processing Selectivity for the Open Pit Mine Production Scheduling Problem	313	F3B
Peter Ebden	Faster Shortest Path algorithms for Siren	113	T4
Bronwyn Erasmuson	On the Mean Cumulative Function of Censored Warranty Data	61	T3
Jeff Foote	Modelling microbes: simulating farm management options	3	T2A
Michael Frankovich	Computational Models for Large Airline Network Revenue Management Problems	81	T4

Author	Title	Page	Session
Gary Froyland	A New Aggregation-Based Method with Improved Processing Selectivity for the Open Pit Mine Production Scheduling Problem	313	F3B
Ron Garland	A New Aggregation-Based Method with Improved Processing Selectivity for the Open Pit Mine Production Scheduling Problem	313	F3B
Brent Gilpin	Modelling microbes: simulating farm management options	3	T2A
Lina María Gómez	Forecasting telecommunications demand for services with short-history data	235	F2B
Samuel Gordon	Rogaining: a Prize-Collecting Orienteering Problem	173	T5
Wendy Gregory	Modelling microbes: simulating farm management options	3	T2A
Ziming Guan	Dynamic Outer-Approximation Sampling Algorithm	29	T2B
Babul Hasan	Decomposition and Pricing Methods for solving MILP Model for an Integrated Fishery	261	F3A
Johnny Ho	Minimizing Average Tardiness and Machine Outsourcing Cost	293	F3B
Wang Hsaio-Fan	Solving a Multi-level Capacitated Facility Location Problem by DVAM	347	F4B
Julie Jang	Scheduling Product Pairs subject to Changeover Times and Mould Constraints	163	T5
Mark Johnston	The Travelling Tournament Problem: Neighbourhoods and Visualisation	321	F4A
Amir Joshan	Analysis of network strategic decisions for airlines	91	T4
Andrew Kerr	A mass-balance gas simulation model for assessing the benefits of gas network augmentation options	203	F2A
Vladimir Krichtal	National Instantaneous Reserve Market in the New Zealand Wholesale Electricity Market	197	F2A
Francisco Lopez	Minimizing Average Tardiness and Machine Outsourcing Cost	293	F3B
Richard Lusby	Routing Trains Through Railway Junctions: A New Set Packing Approach	49	T3
Sarah Marshall	Farthest Insertion Heuristics for the Freeze-Tag Problem	39	T3
Andrew Mason	Faster Map Matching for Emergency Vehicle Trip Analysis	19	T2B
Pablo Maya	Forecasting telecommunications demand for services with short-history data	235	F2B
Carey McCleskey	Reliability Based Assessment Model for Space Vehicles	205	F2B
Stuart Mitchell	Yield Frontier Analysis of Forest Inventory	7	T2A
Kazuo Nakatani	Reliability Based Assessment Model for Space Vehicles	205	F2B
Jason O'Sullivan	Application of Two-Dimensional Renewal Processes in Modelling Product Warranties	215	F2B
John Paynter	Power to the people - improving the quality of information in the census.	225	F2B
John Paynter	Quality vs. Power? In defence of academic values - Community OR in action	273	F3A
Gabrielle Peko	Power to the people - improving the quality of information in the census.	225	F2B

Author	Title	Page	Session
Arun Pennathur	Reliability Based Assessment Model for Space Vehicles	205	F2B
Nicola Petty	If I'm doing it, it must be O.R—An example of the use of Operations Research in evaluating the effectiveness of Special Education provision	245	F3A
Andy Philpott	Optimization, Uncertainty and Equilibria	1	T1
John Raffensperger	A tutorial on hydrogeological optimisation	9	T2A
John Raffensperger	Decomposition and Pricing Methods for solving MILP Model for an Integrated Fishery	261	F3A
Andrea Raith	A Comparison of Solution Strategies for Biobjective Shortest Path Problems	101	T4
Grant Read	Risk-Adjusted Discount Rates and Optimal Plant Mix: A New Formulation for Electricity Market Optimisation	201	F2A
Russell Rhodes	Reliability Based Assessment Model for Space Vehicles	205	F2B
David Richards	A Study of Optimised Ambulance Redeployment Strategies	123	T4
Alex Ruiz-Torres	Reliability Based Assessment Model for Space Vehicles	205	F2B
Alex Ruiz-Torres	Minimizing Average Tardiness and Machine Outsourcing Cost	293	F3B
David Ryan	Tournament Construction Methods for Auckland Bowls	331	F4A
David Ryan	Aluminium Production Scheduling Revisited	333	F4B
Marion Savill	Modelling microbes: simulating farm management options	3	T2A
Lizhen Shao	Finding Representative Nondominated Points in Multiobjective Radiotherapy Planning	249	F3A
Jim Sheffield	Quality vs. Power? In defence of academic values - Community OR in action	273	F3A
Alan Stenger	Reflections on Forty Years of Researching and Implementing Inventory Management Systems in Commercial Firms	183	F1
David Sundaram	Quality vs. Power? In defence of academic values - Community OR in action	273	F3A
Dan Trietsch	Quality vs. Power? In defence of academic values - Community OR in action	273	F3A
Dan Trietsch	Safe Scheduling	283	F3B
Richard Varey	A New Aggregation-Based Method with Improved Processing Selectivity for the Open Pit Mine Production Scheduling Problem	313	F3B
Hamish Waterer	Tournament Construction Methods for Auckland Bowls	331	F4A
Oliver Weide	An Iterative Approach to Airline Scheduling	335	F4B
David Wood	Modelling microbes: simulating farm management options	3	T2A
Golbon Zakeri	How to set optimal line tariffs	37	T2B
Edgar Zapata	Reliability Based Assessment Model for Space Vehicles	205	F2B
Jianmei Zhang	Reliability Based Assessment Model for Space Vehicles	205	F2B

Optimization, Uncertainty and Equilibria

Andy Philpott
Department of Engineering Science
University of Auckland
New Zealand
a.philpott@auckland.ac.nz

Abstract

There has been a growing recognition in the OR/MS community that accounting for uncertainty in the decision models that we create is a key ingredient in making them useful in practice. This recognition has come about through the advocacy of some of the pioneers of planning under uncertainty, but also the growth of computational methods to allow realistic problems to be solved in practical settings. In this talk I will present my personal views of these developments, with a particular emphasis on modelling in electricity markets.

Modelling microbes: simulating farm management options

David Wood
Andrew Ball
Brent Gilpin
Wendy Gregory
Jeff Foote
Marion Savill

Institute of Environmental Science and Research Ltd (ESR),
P O Box 29-181, Christchurch 8540, New Zealand
David.Wood@esr.cri.nz

1. Introduction

Intensification of land use and urbanisation in New Zealand is widely believed to be a key reason for the deterioration of surface water quality. Managing the environmental load of microbes from diffuse sources, such as agriculture, is particularly challenging. Environmental microbial loading closely relates to the issues of diffuse nutrient and sediment pollution. Microbial contamination of water is a complex area of environmental management and our understanding of the subject is incomplete. The management of surface waters involves many different stakeholders, each with their own perspective about who is responsible for its quality and what should be done about it.

This project focuses on what part the agricultural sector, and in particular individual farmers, can play to reduce their contribution to environmental microbial loading. We used a system dynamics (SD) approach (Maani and Cavana, 2000). The outputs from the project will be developed into tools that can assist farmers manage microbial contamination of water.

2. Workshop

A workshop with farmers, policy makers and scientists was held at the beginning of the project with a view to gaining an understanding of the factors relating to environmental loading by micro-organisms from agriculture from a range of perspectives. The workshop focused on dairy farming and a bacterium called *Campylobacter*. This organism is found in the faeces of many dairy cattle. As far as we know it has no detrimental effect on cows, but it is pathogenic to humans and causes gastroenteritis. This organism was chosen because it is reasonably well understood, contributes significantly the burden of illness in communities, and in principle can be controlled. We also believed we could derive a generic understanding of environmental microbial loading and agricultural practices based on this organism. The main output of the workshop was an influence diagram.

3. Influence diagram

The influence diagram included on-farm activities, the aquatic environment, and human health as well as economic and regulatory drivers associated with agriculture. It was noted that many of the factors were interrelated and that there were a number of feedback loops within the system.

4. Identification of initial leverage points

Analysing the influence diagram in conjunction with other information from the workshops allowed the identification of leverage points that may form the basis of effective management options. As the objective of the work was to help farmers manage environmental microbial loading, some of the options suggested were beyond the farm gate, and so could not be used by individual farmers. The remaining options were presented to a group of farmer consultants for discussion in terms of their feasibility and acceptability.

5. Modelling

The initial part of the project, in particular the development of the influence diagram, provided an understanding of the issues and some potential solutions. However, the level of impact of individual or combined interventions remained unclear. Discussions with stakeholders revealed the dynamic nature of the issues and the presence of feedback loops, which suggested SD simulation as an appropriate technique to address the issues. The primary need was to improve our understanding of the behaviour of the system, rather than to provide a simple prediction of the efficiency of various interventions (see Ford, 1999, p. 10); this confirmed that SD simulation was a useful approach.

Simulation work of a single farm was focussed on, as this could be thought of as being on the scale of an individual decision unit. We acknowledge that there are cumulative impacts from microbial loading on larger scales – this issue will be handled using more traditional hydrological modelling approaches, and is not discussed here.

6. Building the model

Once the decision had been made to model an individual farm, there was a question about what should be done and how we should go about doing this. We used an approach described by Wilson (1992), who suggested the farming production system was the overlap of the ecosystem and the social system.

The influence diagram developed earlier described the system. The descriptive model needed to be converted into level and rate equations. Much of this information came from a literature review that brought together the research evidence associated with the ecosystem and the farming production system. The simulation model was used to integrate the available evidence: for example, one key element of the model was the transport of microbes from the land surface to water (Jameson et al, 2004). It was important that the simulation took into account this body of knowledge if it was to be credible in the eyes of many of the stakeholders such as farmers and scientists.

The model was developed around the concept of a farm being a microbial reservoir, the size of the reservoir being dependent on the number of microbes being shed by animals, and the die-off rate of these microbes in the environment. Microbes can move out of the farm microbial reservoir to the surface water environment. The primary mechanism for this appears to be surface runoff from artificial (irrigation) and natural precipitation events. We also considered the possibility of the number of microbes being shed being related to the size of the microbial reservoir.

Once the model had been developed, its behaviour was tested to ensure it produced results that were either physically plausible or similar to those found from other modelling or biophysical studies.

7. Testing interventions or best management practices

Once we were confident that the model was a reasonable representation of the farming system, we tested some of the interventions or best management practices. The potential interventions could reduce the number of microbes in the farm reservoir or reduce the export of microbes in the farm reservoir.

It appeared possible theoretically to reduce the number of microbes in the reservoir, for example by reducing the microbial carriage rate by domestic animals, but the scientific evidence base to support these types of interventions is quite weak. Stronger experimental and observational evidence was available relating to interventions that reduced the export of microbes from farms.

8. Developing management options

We can now assess the effectiveness of various management options. This will be achieved by debating which options are acceptable and technically feasible in the light of current agricultural practice and knowledge. Clearly, some of the options could involve considerable expenditure. Some of the options are recognised as current best practice by the farming industry. The feedback we receive will determine whether we need to revisit the simulation work and develop new options or whether the current options require costing and then development into management tools for the farming industry.

9. Conclusions

The wide variety of perspectives, knowledge and interests associated with working with complex environmental and social systems is challenging. It takes considerable effort for researchers to come to grips with these issues. Using system dynamics, either in qualitative or quantitative ways, can help improve our understanding of these issues.

Though we can make informed suggestions as to how to improve the situation, time will tell whether these suggestions will lead to changes in practice and improved management of microbial loading to the environment.

10. Acknowledgments

This work has been funded by the Sustainable Farming Fund (grant no. 45/005) and Dairy Insight. We would like to thank the people who have taken part in the project workshops and provided useful feedback to the project team.

11. References

- Ford, A., 1999. *Modeling the Environment. An Introduction to System Dynamics Modeling of Environmental Systems*. Island Press, Washington.
- Jamieson R., R. Gordon., D. Joy., H. Lee., 2004 "Assessing microbial pollution of rural surface waters a review of current watershed scale modeling approaches." *Agricultural water management*. 70, 1-17
- Maani, K., R. Cavana, 2000. *Systems Thinking and Modelling: Understanding Change and Complexity*. Prentice Hall, Auckland.
- Wilson, J., 1992. *Changing Agriculture. An Introduction to Systems Thinking*, 2nd Edition, Kangaroo Press, Kenthurst, Australia.

Yield Frontier Analysis of Forest Inventory

Stuart Mitchell
University of Auckland
s.mitchell@auckland.ac.nz

Abstract

Standing inventory analysis allows a forester to predict the yield of a section of forest before the trees have been felled. Current standing inventory tools produce a point estimate of the yield for given input data. This paper presents a method of standing inventory analysis that seeks to characterise all possible yields for a section of forest and present them as the feasible region of a Linear Program.

A tutorial on hydrogeological optimisation

John F. Raffensperger
Dept. of Management, Private Bag 4800
University of Canterbury, Christchurch, New Zealand
john.raffensperger@canterbury.ac.nz
27 October 2006

Abstract

This paper presents a tutorial on hydrogeological optimisation (HO). The paper will cover maximising well water extraction and contaminant remediation, based on examples from the literature. The paper briefly explains the required hydrology simulation, but the main focus is on the operations research. The paper is intended to introduce an important new field of application to operations researchers without hydrology background. The motivations are to encourage new efforts in this important area, to demonstrate the immediate applicability to an urgent global challenge, and to specify the policy changes required to address that challenge. The paper describes how HO can be used to solve critical problems currently facing New Zealand, and concludes with a proposal for a national workbench for open management of fresh water.

1 Why should operations researchers care about water?

The world water crisis is getting worse. World-wide, most water is used for agriculture, and much of this comes from groundwater. Thus, contention for ground water is increasing, as agricultural users attempt to reduce their risk while trying to grow food for the world's increasing population. The world is already experiencing destruction of waterways and aquifers on a massive and accelerating scale. Agriculture subsidies are quickening this destruction.

Operations researchers can do much to help address these historic problems. We know how to manage resources, and we have the right methods for making hard decisions. The required methods are classical: decision analysis, simulation, and math programming. The applications are equally classical: risk reduction, forecasting, maximising profit subject to constraints – in this case, constraints that ensure sustainability. The domain is new to many operations researchers, but is very much worthy of attention: hydrological optimisation (HO).

This paper gives two simple examples of HO taken from the literature. The models are not difficult, but have the potential for a huge impact. Following the examples, I describe how HO can be used to allocate water, and I propose a national workbench in open management of fresh water, with a call to action.

As linear programming is less than 60 years old, it's no surprise that the application of LP to groundwater is younger still, and the literature is dominated by a few authors. The interested researcher should start with Ahlfeld & Mulligan (2000), who have written an excellent textbook in this topic. That textbook also gives an excellent literature survey. Gorelick has been especially prolific; see for instance Feyen & Gorelick (2004), which describes a stochastic optimization for HO.

Greenwald (1998) developed quite general software called MODMAN that connected an existing deterministic hydrological simulation called MODFLOW (Harbaugh & McDonald, 1996) to a linear program. MODFLOW is a deterministic

FORTTRAN simulator of water flow in three dimensions and over time, based on approximations to the differential equations that describe the water flow. The simulation calculates the effects of pumping on each location of interest. These effects become coefficients in the LP constraint matrix. The user then uses standard LP software to solve and interpret the resulting LP. MODMAN could create the appropriate LP for a variety of different management problems, two of which are given below. Ahlfeld & Mulligan (2005) developed GWM2000, a program similar to MODMAN, but that uses an updated version of MODFLOW and contains its own LP solver.

3 A typical problem in HO: maximising water take.

3.1 Eighteen wells, but not enough water.

Suppose a water users' group in a small catchment with 18 wells asks you for help with their water allocation problem. The users' group wants to take as much water as possible, while satisfying government regulations about sustainability. In particular, a stream must have a minimum flow, the aquifer cannot be drawn down below a certain level in three places, and the aquifer near the coast must be maintained to prevent intruding saltwater which would make the fresh water aquifer salty. A diagram is shown in Figure 1. This example is from Greenwald (1998).

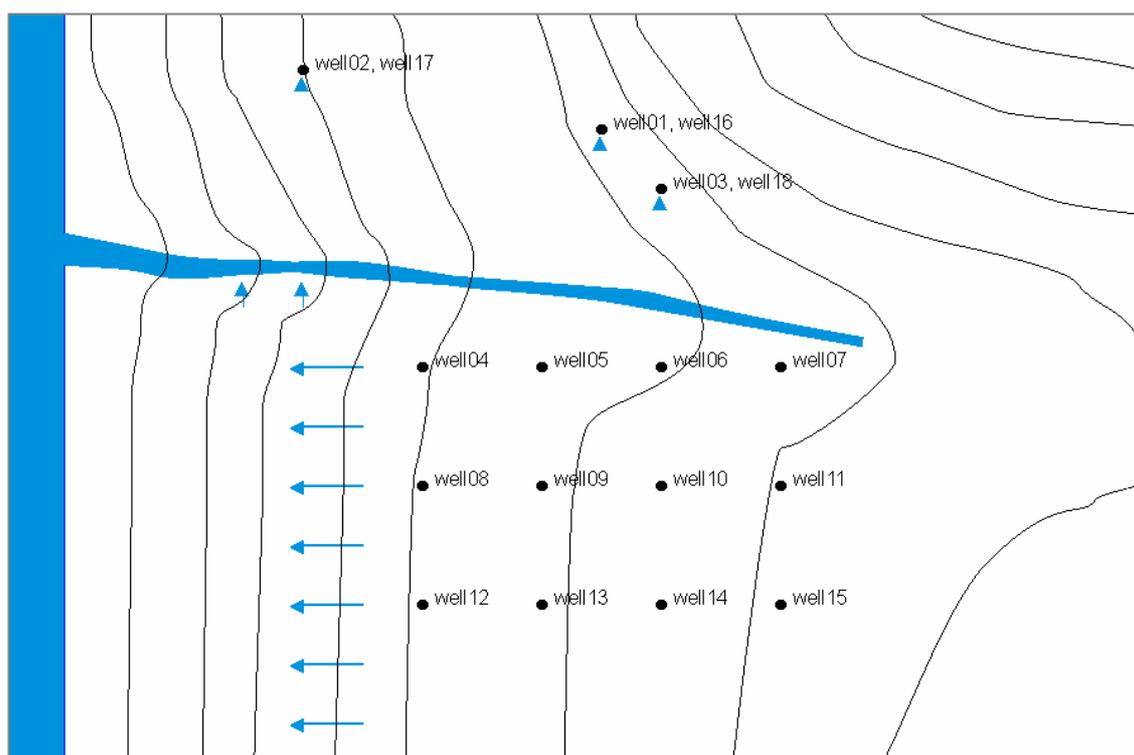


Figure 1. A catchment near the coast, with 18 wells and a stream. Arrows and triangles indicate environmental constraints. Curves indicate lines of equal hydrological pressure.

This problem can be solved using a combination of simulation and linear programming (LP). For this paper, let's assume a hydrologist has built and can operate the simulation, and we will focus on the LP. The LP decision variables are the amount that each well pumps, and the users' group wants to maximise that sum. But the exact structure of the linear program is still unclear. How do we model the constraints? How does pumping interact with the stream and aquifer? To complete the LP, we need to

know the marginal effect of each well on the constraint locations. These marginal effects depend on the character and behaviour of water flow. When a farmer pumps water from the ground, the level of water – the *head* – falls nearby. Over hours, days, and weeks, this effect radiates out away from the well, wave-like, with a smaller effect further away and later in time. These effects of pumping on head will be the coefficients in our LP constraints.

For example, consider the constraint associated with maintaining flow in a stream. The stream flow depends on the aquifer around it. If irrigators take groundwater near the stream, the stream will tend to fall as it recharges the aquifer. Similarly, if the aquifer rises, the stream tends to rise as it is recharged by the aquifer. The aquifer head, at each location and time period, is therefore the key state variable, and thus we need to know how pumping at one location affects the aquifer head at other locations. A local hydrologist or government agency can usually supply this data to us. Indeed, many catchments already are modelled with MODFLOW. The local hydrologist can also tell us the required head levels to ensure that the sustainability constraints will be satisfied.

Consider the format of the marginal effect data, which we denote as F_{ijt} = drawdown at location i due to pumping at well j at time t . Hydrologists call the F_{ijt} table the *response matrix*. A hydrologist will find these F_{ijt} coefficients using MODFLOW. To get the response matrix, the hydrologist simulates pumping one well at a time, and the simulation gives the resulting effect at all other locations. For the first well, $i=1$, the output gives F_{1jt} , for $j=1, \dots, N$, for $t=1, \dots, T$. This is the change in aquifer head at each location and future time period for a unit change in pumping at location 1 and period 1. The hydrologist then repeats the simulation for every well, $i=2, \dots, N$, one well at a time.

3.2 An LP to maximise water take.

Given decision variables q_{it} and response matrix F_{ijt} , we can calculate the drawdown at location j in period t as $\sum_{i=1}^n \sum_{u=1}^t F_{iju} q_{iu}$. We thus have the information needed, conceptually, to write our LP.

Indices: i, j, k , well or location, $i=1, \dots, N$.
 u, t , periods, $t=1, \dots, T$.

Parameters: F_{ijt} = drawdown at location i due to pumping at well j at time t ,
 L_{it}, U_{it} = lower and upper bounds on head at location i at time t .
 C_i = maximum amount of water that can be pumped at well i at any time.

Variables: $q_{it} \geq 0$, well abstraction rate at location i , period t .

- (1) **Model MaxWater:** maximise $\sum_{i=1}^n \sum_{u=1}^t q_{iu}$,
- (2) Environmental flows, $L_{it} \leq \sum_{j=1}^n \sum_{u=1}^t F_{iju} q_{ju} \leq U_{it}$, for $i=1, \dots, n$, $t=1, \dots, T$.
- (3) Well capacity, $q_{it} \leq C_i$, for all i, t .
- (4) Cannot inject water: $q_{it} \geq 0$, for all i, t .

From an O.R. point of view, this is an easy LP. Typically the catchment has 300 wells, but only a half dozen or so environmental control points. The number of time periods can vary, but often T is small, e.g., 3 weeks. The matrix F_{ijt} is dense, but this causes little difficulty with modern solvers. Thus, we are looking at solving a quite ordinary linear program of perhaps a few thousand columns and a few hundred rows.

One issue may come to mind for the experienced operations researcher: Is this hydrological behaviour linear? If not, can we really solve the problem with an LP? The answer is, no, but yes. The behaviour is not in general linear. However, hydrologists have found through experience that the hydrological behaviour is often well approximated by linear equations. Furthermore, we can update our LP in case it is

inaccurate. We can ask the hydrologist to simulate our current solution, q_{it}^* . If the simulation indicates that the environmental constraints would be violated – or if they have much more slack than we expected – then we simply update the response matrix F_{ijt} and re-run the LP. We may need several iterations before we are satisfied.

This modest linear program has potential to solve some of the world's most urgent problems. Equation 2 literally enforces sustainability. The constants L_{it} and U_{it} must be set with care, but for managing on-going water allocation, the fact that *sustainability constraints are set at all* is a relatively new idea. Sustainability constraints are not new to experts in hydrological optimisation, but I know of no government agency anywhere in the world that plans water allocation using optimisation and explicit sustainability constraints. If our little water users' group would heed the advice given by our model MaxWater, they would literally simulate every take of water to ensure it was sustainable. The market for consulting work in this area is very much wide open!

3.3 Numerical example and solution

Remembering that the variable subscripts are “q,well, time”, here is part of the objective for our linear program.

$$\text{Max } q_{4,1} + q_{4,2} + q_{4,3} + q_{5,1} + q_{5,2} + q_{5,3} + \dots + q_{17,1} + q_{17,2} + q_{17,3} + q_{18,1} + q_{18,2} + q_{18,3}.$$

As the LP has many drawdown constraints, here is the one for period 3, location 16.

$$30 \leq d_{3,16} \leq 99,$$

$$\begin{aligned} d_{3,16} &+ 0.1765 q_{4,1} + 0.3077 q_{5,1} + 0.3443 q_{6,1} + 0.3147 q_{7,1} + 0.0874 q_{8,1} \\ &+ 0.1402 q_{9,1} + 0.1590 q_{10,1} + 0.1502 q_{11,1} + 0.0360 q_{12,1} + 0.0546 q_{13,1} \\ &+ 0.0626 q_{14,1} + 0.0611 q_{15,1} + 0.7038 q_{16,1} + 0.2254 q_{17,1} + 0.6654 q_{18,1} \\ &+ 0.1545 q_{4,2} + 0.3077 q_{5,2} + 0.3288 q_{6,2} + 0.2415 q_{7,2} + 0.0535 q_{8,2} \\ &+ 0.0959 q_{9,2} + 0.1048 q_{10,2} + 0.0844 q_{11,2} + 0.0153 q_{12,2} + 0.0254 q_{13,2} \\ &+ 0.0283 q_{14,2} + 0.0246 q_{15,2} + 1.1239 q_{16,2} + 0.1847 q_{17,2} + 0.9715 q_{18,2} \\ &+ 0.0602 q_{4,3} + 0.1644 q_{5,3} + 0.1711 q_{6,3} + 0.0829 q_{7,3} + 0.0125 q_{8,3} \\ &+ 0.0274 q_{9,3} + 0.0292 q_{10,3} + 0.0180 q_{11,3} + 0.0022 q_{12,3} + 0.0043 q_{13,3} \\ &+ 0.0047 q_{14,3} + 0.0034 q_{15,3} + 25.043 q_{16,3} + 0.0730 q_{17,3} + 1.8170 q_{18,3} \\ &= 18.5727, \end{aligned}$$

$$\text{Well capacity constraints: } 0 \leq q_{4,1}, q_{4,2}, q_{4,3}, q_{5,1}, \dots, q_{18,1}, q_{18,2}, q_{18,3} \leq 1.$$

The experienced operations researcher may immediately have some questions. Variables $q_{1,t}$, $q_{2,t}$, and $q_{3,t}$ seem to have disappeared. Where did they go? Answer: notice in Figure 1 that wells 1 and 16, wells 2 and 17, and wells 3 and 18, are all very close together. The wells are so close that MODFLOW cannot distinguish between them. Thus, we must treat each of these pairs of wells as one well.

This numerical example is not quite in the form we gave algebraically in model MaxWater. The example has variable $d_{16,3}$, the drawdown in period 3 at location 16. But because the next row is an equality, variable $d_{16,3}$ may be substituted out easily.

What is the meaning of the right hand side, 18.5727? The answer may be seen when all $q_{it} = 0$. The right hand side is the natural head when no one pumps any water. What is the meaning of the constraint coefficients? For example, what is the meaning of the coefficient 0.1765 on variable $q_{4,1}$? This is the drawdown in head at location 16 in period 3, for 1 megalitre of pumping at well 4 in time period 1. How can we change the model, if some wells watered more valuable crops? To give different values to water at different locations and time periods, we can simply change the objective coefficients.

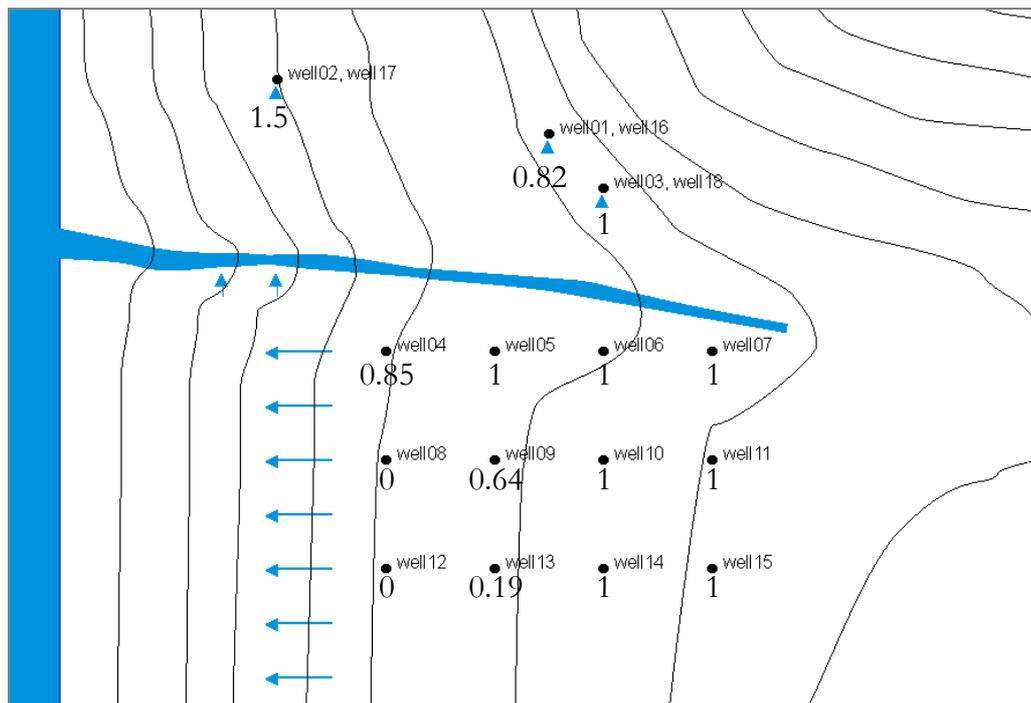


Figure 2. Solution to the numeric example.

In the solution, observe that no water is taken near the coast at wells 8 and 12, and well 13 can have less than the average amount of water. The saltwater intrusion constraints are tight. However, the stream flow constraints are loose. The aquifer drawdown constraints near wells 16, 17, and 18 are tight.

Thus, we see that water availability depends on where and when it is taken. There is no one fixed quantity of water available. Rather, the quantity that can be sustainably taken depends on how close the withdrawals are to environmentally sensitive locations, and withdrawals close to those sensitive locations tend to disproportionately reduce the amount of water that is available further away.

4 How to clean up a hydrological mess with LP

4.1 Someone spilled something bad.

Figure 3 shows an example where a chemical spill has resulted in an underground plume that must be mitigated. To prevent the plume from spreading, we can install a strategically-located well, and pump the contaminant out, which can then be treated. What is the lowest rate we can pump, but still contain the plume? The key decision variable is the pumping rate. This problem is due to Javandel and Tsang (1986), who solved it analytically. Greenwald (1998) showed that we can do a better job with LP.

We will assume that a hydrologist has already developed a MODFLOW simulation of the area. The rectangular grid in Figure 3 shows how the hydrologist chose to divide the area into cells; the simulation then treats each cell as a point. The simulation provides the response matrix, as described in section 2.1.

We need optimisation for this problem because extraction of the plume depends on the direction of aquifer flow, which depends on the difference in head between every pairs of cells. Remembering the water flows downhill, higher water levels in one cell tends to create flow toward adjacent cells with lower water levels, and this effect also depends on the distance. Moreover, the absolute value of head changes over time. Since the pumping rate affects head levels, we need additional decision variables to measure

head differences at key locations, which are right at the edge of the plume. The goal is to force the direction of flow to be inward, relative to the capture curve, toward the well.

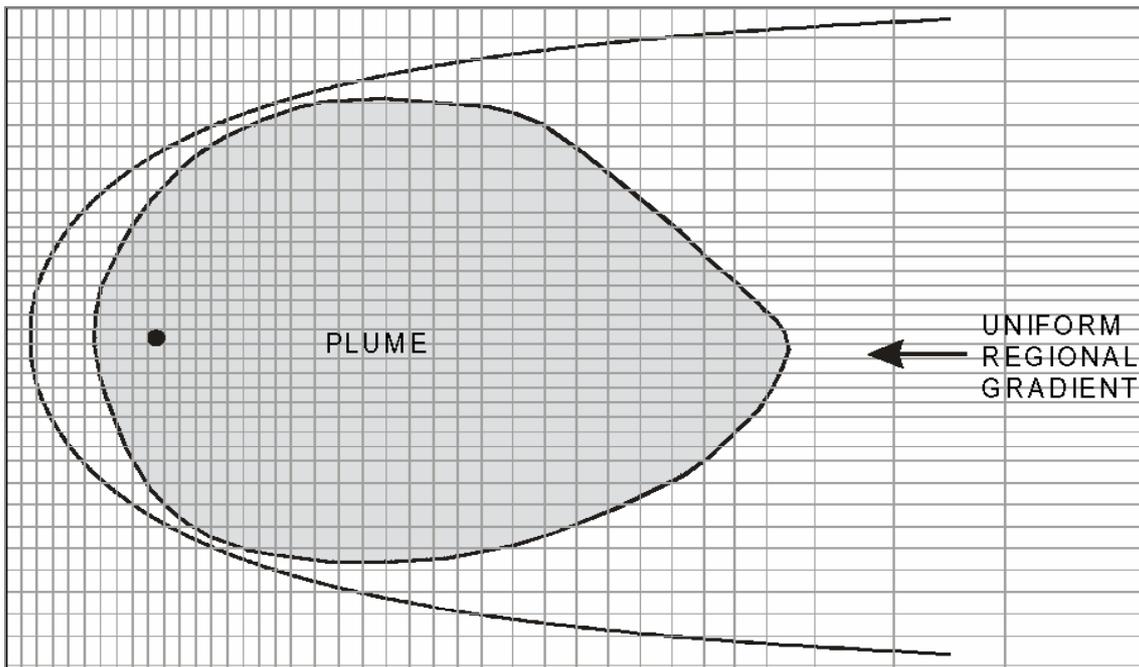


Figure 3. Plume containment example (Greenwald 1998). Aquifer flow is right to left.

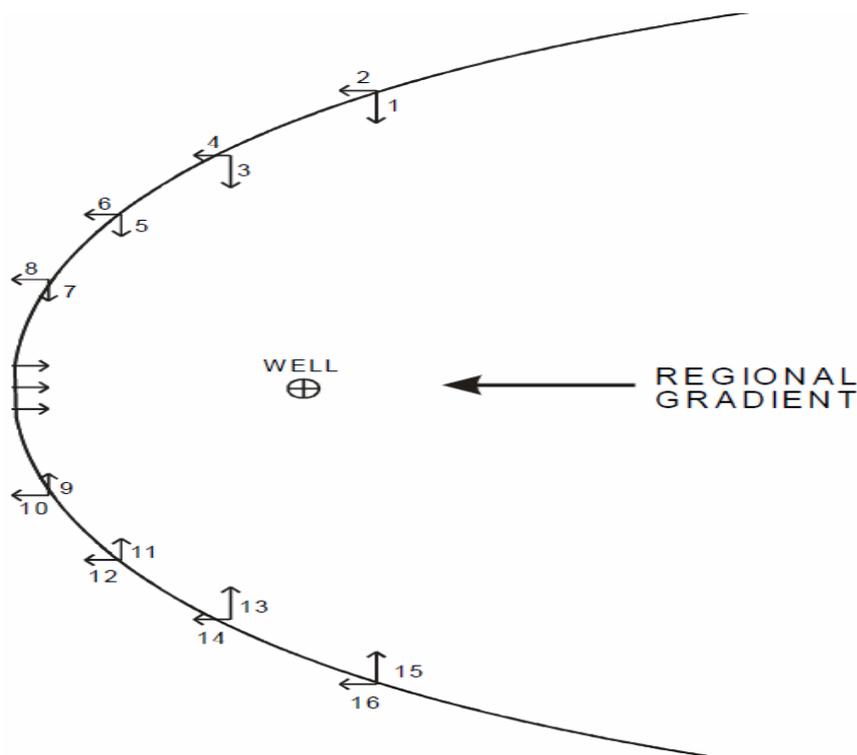


Figure 4. Required gradients to contain the plume (Greenwald 1998).

As shown in Figure 4, at the left-most edge, we need three head difference constraints. A head difference simply compares the head between two adjacent cells. Furthermore, we will use 16 gradient constraints along the capture curve boundary. A gradient constraint is similar to a head difference constraint, but the cells need not be adjacent, and so the constraint must be scaled by distance. For each pair of constraints (such as 1 and 2), we are comparing the heads in a horizontal pair of cells. We also

compare the heads in a vertical pair of cells. Each set of pairs have one common cell right on the boundary.

Finally, we will use eight *relative gradient* constraints. We want to be sure that the relationships between the gradients are such that the flow is the correct direction. If we think of the gradients as vectors, we want the ratio of the two vectors to follow the tangent of the desired angle of flow. These angles are given as the angles between the gradient vectors in Figure 4. Figure 5 shows a diagram of the angles.

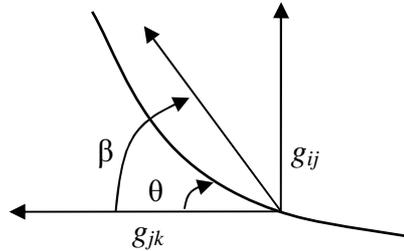


Figure 5. How a relative gradient constraint works. θ is the limiting flow direction. β is the resulting optimal flow (Greenwald 1998).

4.2 LP for plume containment.

We can write the LP for plume containment as follows. The model has no subscript for time, as it is intended to produce a long-run steady state. Also, with a bit of effort, we could substitute away the variable g_{ij} .

Indices: i, j, k , location, where $i, j, k = 1, \dots, N$.

Parameters: C_i = pumping capacity of well i ,

D_{ij} = distance between locations i and j .

F_{ij} = drawdown at location i due to pumping at well j ,

G^L_{ij}, G^U_{ij} = lower and upper limits on gradient between locations i and j ,

H_i = natural head at location i ,

L_{ij}, U_{ij} = lower and upper bounds on head difference between i and j ,

$\theta_{i,j,k}$ = minimum angle between gradients from cells (i,j) and cells (j,k) .

Variables: $q \geq 0$, well abstraction rate at location i , period t ,

$g_{ij} \geq 0$, gradient between locations i and j ,

$h_{ij} \geq 0$, head difference between locations i and j .

(5) **Model Remediate:** minimise q , subject to

(6) **Head differences:** $L_{ij} \leq (F_i - F_j)q \leq U_{ij}$, for locations pairs $i, j \in 1, \dots, N$.

(7) **Gradient constraint:** $D_{ij}g_{ij} + (F_i - F_j)q = H_i - H_j$, for locations pairs $i, j \in 1, \dots, N$.

(8) **Bounds on gradient:** $G^L_{ij} \leq g_{ij} \leq G^U_{ij}$,

(9) **Relative gradient:** $g_{ij} \leq \tan \theta_{ijk} g_{jk}$,

(10) **Well capacity:** $0 \leq q \leq C$.

4.3 A numerical example.

Minimise the required pumping: $\min q$, subject to

Head difference constraints, for the three locations right at the tip of the plume:

$$0.104523 q \geq 0.1, \quad 0.106498 q \geq 0.1, \quad 0.104523 q \geq 0.1,$$

Gradient constraints:

$$75 g_1 + 0.089 q = 0,$$

$$50 g_9 + 0.052 q = 0,$$

$$50 g_2 + 0.006 q = 0.1,$$

$$50 g_{10} - 0.073 q = 0.1,$$

$$75 g_3 + 0.115 q = 0,$$

$$50 g_{11} + 0.073 q = 0,$$

$$50 g_4 - 0.017 q = 0.1,$$

$$50 g_{12} - 0.047 q = 0.1,$$

$$50 g_5 + 0.073 q = 0,$$

$$75 g_{13} + 0.116 q = 0,$$

$$50 g_6 - 0.047 q = 0.1,$$

$$50 g_7 + 0.052 q = 0,$$

$$50 g_8 - 0.073 q = 0.1,$$

Required gradient bounds: $g_2, g_4, g_6, g_8, g_{10}, g_{12}, g_{14}, g_{16} \leq 2$.

Relative gradient constraints:

$$g_1 \leq 0.6 g_2,$$

$$g_3 \leq 0.97 g_4,$$

$$g_5 \leq 1.43 g_6,$$

$$g_7 \leq 2.61 g_8,$$

Well capacity:

$$50 g_{14} - 0.017 q = 0.1,$$

$$75 g_{15} + 0.089 q = 0,$$

$$50 g_{16} + 0.006 q = 0.1.$$

$$g_9 \leq 2.61 g_{10},$$

$$g_{11} \leq 1.43 g_{12},$$

$$g_{13} \leq 0.97 g_{14},$$

$$g_{15} \leq 0.6 g_{16}.$$

$$0 \leq q \leq 2,$$

The solution is to pump at a rate of $q=1.074$, fortunately below the capacity of 2.

4.4 Other similar examples

What other interesting problems can be solved with HO? Greenwald (1998) gives one other interesting example, which is to maximise pumping at a well near a river. The river acts as a leaky boundary, and for environmental reasons this boundary must be respected. Thus, the constraint is to avoid drawing water from the other side of the river.

Ahlfeld & Mulligan (2000) give more examples, including that of dewatering a construction site. The construction requires a deep hole and maintaining that hole without water during the construction, despite a surrounding aquifer. The solution is to position wells around the construction to remove water sufficiently to keep the hole dry. HO includes stochastic and integer problems as well.

5 Using HO to solve NZ's problems of freshwater management.

5.1 Complex societal problems.

Compared to the modest LPs above, the world is now facing much bigger and more complicated problems associated with water. Most importantly, agriculture depends heavily on groundwater, and government agencies that manage water must decide how to allocate this water, and must be able to determine how much water remains at any given time. The data available is quite spotty. In NZ, Tasman and Marlborough currently have relatively good hydrology models available, as these areas are relatively small with simple hydrology. Canterbury, on the other hand, has 30 or 40 times the groundwater of the rest of NZ put together, and Canterbury's aquifer structure is extremely complicated and still not well mapped. The few hydrology simulations available are difficult to use (in one case, taking literally weeks to run), and in some cases their reliability has been questioned successfully. But the water authorities are not using the available data in the best fashion. Consents are approved or rejected one at a time, yet the problem would best be solved as a simultaneous decision.

How can a unitary authority manage water within one catchment? Answer: simply solve the MaxWater LP in section 3.2 for the catchment. Existing consents may be fixed, or set to historical takes, or set to forecasted takes. The decision variables q_{it} correspond to new consents. The O.R. approach allows rich analysis. Dual price information on the environmental constraints tells where the catchment needs special attention. Since the variables are subscripted by time, the authority has the opportunity to offer consents by month or season, rather than a fixed yes-or-no for the whole of the many-year consent period. The LP further offers a tool to manage consents take-backs during drought.

5.2 National workbench for open management of fresh water.

To solve these difficult problems requires data, and, for the most part, NZ is rather short of it. National leaders have lately been fond of saying that NZ needs “a better return on its research dollars.” With a growing family, the head of house is always looking for a better value on potatoes, but at some point, the family simply needs more potatoes. FRST 2006-7 had virtually no research funding for water, despite the national consensus on its importance, and just as the country’s water crisis is deepening. Auckland is getting over \$2 billion/year in new funding just for roads. Surely the nation’s water supply is as important! The first priority is a **national commitment of significant funding** to solving the problems of water management.

The second priority is for a **central water authority**. NZ’s governance of water is split among Ministry for the Environment, Ministry for Agriculture and Forestry, and the local regional and district councils. This is one reason why our water management is rated poorly internationally (Dinar & Saleth, 2005). A central water authority should be tasked with creating systems for reporting and modelling, to show how much water NZ has, how much it will have, who is using it, and for which purposes. One possible task of this authority would be to consolidate the hydrology and climatology research functions of the Crown research organizations, to better fund them, and to eliminate the unnecessary and destructive competition among them.

The third priority is for **better data, and made publicly available**. The U.S. Geological Survey makes water management tools and data available throughout the U.S. Similarly, the Australian CSIRO is developing a Water Resources Observation Network. We need to play catch-up.

NZ should develop a standardised format for water data, in collaboration with CSIRO, USGS, the U.S. UCAR, and other water managers and earth science researchers abroad. Data should be stored in standardised format, so anyone may obtain that data, and at no charge, both by ordinary web page and by web services. The U.S. National Weather Service has made headway on data formats with “Hydrology XML.” The data required is as follows.

- Well data, including consents, historical abstractions, and real time abstractions. (All wells with consents above a given rate should be metered, and this rate should be reduced every year.) Because water is a public good, and every abstraction affects everyone else, water consents and abstractions cannot be private data.
- Remote sensing for lake and reservoir levels, with storage of this information for historical information. Similarly, remote sensing for stream flow data, starting with the larger streams, and progressively implemented to smaller streams, with storage of this information for historical information.
- Remote sensing for quality, with storage of this data for historical information.
- Climatology data, real time and historical, including data that will work with the hydrology models. Geology and seismic data should be standardised and stored in publicly available databases.
- Required environmental flows and quality standards in every catchment. This includes specification of water bodies of national importance.
- **Data for hydrology modelling**, with methodical development of hydrology models for all major NZ catchments. Each model should be reviewed for quality, and inaccurate models should be improved. The NZ government, with science advice, should set standards of accuracy for hydrology and hydrogeology modelling. A model with the

prescribed standard of accuracy should stand as evidence in court. (Texas and Utah have done this.)

The fourth priority is for convenient public tools for accountability. The public has a right to verify that the environmental standards are indeed met. Users, researchers, and the public should have the ability to observe historical, up-to-date, and forecasted water flow information. Everyone in NZ should have the right to download hydrology and climatology software and data, and observe the effects of water takes. This software should be sufficiently user friendly that lay people can operate it. For example, anyone should be able to display current data for a particular dam on their computer screen. Similarly, anyone should be able to see real time and forecasted data on rivers flows, to know when they would receive their entitlement, or to know whether the required environmental flows are satisfied.

5.3 Conclusion.

The above examples demonstrate that HO is a relatively straightforward task, with software that is already available. For the operations researcher, the difficulty is understanding the hydrology, so work is best done in collaboration with a hydrologist, but the O.R. is not difficult. For the policymaker and for the public, the difficulty is the lack of good data. Addressing the increasingly acute problem of freshwater management will require a major new commitment from national government. I encourage my colleagues in operations research to take on study of natural resources.

References

- Ahlfeld, D.P., Barlow, P.M., and Mulligan, A.E., 2005, *GWM—A ground-water management process for the U.S. Geological Survey modular ground-water model (MODFLOW-2000)*, U.S. Geological Survey Open-File Report 2005-1072, <http://pubs.usgs.gov/of/2005/1072/> accessed 15 July 2006.
- Ahlfeld, D.P., and Mulligan, A.E., *Optimal management of flow in groundwater systems*, Academic Press, San Diego, CA, 2000.
- Dinar A.; and Saleth R. M.; “Can water institutions be cured? A water institutions health index,” *Water Sci & Technology: Water Supply*, v5, n6, pp. 17-40, 2005.
- Feyen, L., & S. M. Gorelick (2004), “Reliable groundwater management in hydroecologically sensitive areas,” *Water Resour. Res.*, 40, W07408, doi:10.1029/2003WR003003.
- Greenwald, R. (1998), *MODMAN: An optimization module for MODFLOW Version 4.0*, GeoTrans, 2 Paragon Way, Freehold, New Jersey 07728, pp. 6-1 to 6-8. [On line at: www.geotransinc.com/modman.html.]
- Harbaugh, Arlen W., and McDonald, Michael G. (1996), *User's Documentation for MODFLOW-96, an Update to the U.S. Geological Survey Modular Finite-Difference Ground-water Flow Model*: U.S. Geological Survey Open-File Report 96-485 [Available on line at: <http://water.usgs.gov/software/modflow-96.html>].
- Javandel, I., and C.F. Tsang, (1986), “Capture-zone type curves: a tool for aquifer cleanup,” *Ground Water*, 24(5): 616-625.
- McCarthy, Michael, “Global warming will threaten millions say climate scientists,” http://www.nzherald.co.nz/category/story.cfm?c_id=26&ObjectID=10404255, accessed 17 Oct 2006, citing an unpublished study by Dr Eleanor Burke, from the Met Office's Hadley Centre for Climate Prediction and Research.

Faster Map Matching for Emergency Vehicle Trip Analysis

A.J. Mason

Department of Engineering Science
University of Auckland, New Zealand
www.esc.auckland.ac.nz/Mason

Abstract

We build on earlier work by outlining a new approach for determining the likely route taken by a vehicle given a set of GPS locations and times recorded by an AVL (automatic vehicle location) system. Our version of this ‘map-matching’ problem is unusual in that the GPS datapoints are widely spaced. By using a modified Dijkstra’s algorithm, and making a number of assumptions about the likely form of the underlying vehicle route, we propose a new map matching algorithm with a faster running time than our earlier approach. This paper focuses on the model development for this new approach. Detailed experimental results will be presented in subsequent work.

Key words: Map matching, GPS, ambulances, dynamic programming, Dijkstra.

1 Introduction

In earlier work [3], we presented an algorithm being used in the Siren ambulance software [6] for deducing the likely route for an emergency vehicle to have travelled in a road network given a set of sparse GPS datapoints generated during the vehicle’s travel. This problem is known as map-matching and is a form of prize-collecting shortest path problem. In our map-matching problems, we are given a set of GPS datapoints giving the approximate locations of the vehicle at a sequence of times, with successive GPS points typically being spaced at 30 second intervals. (The GPS datapoints may also specify vehicle speed, heading, and distance travelled since the last datapoint, but the use of these in our algorithms is optional.) This 30 second spacing is larger than that considered in other works (e.g. [5], [4], [2]), which makes the problem much more difficult. In our earlier work, we presented a detailed algorithm for reconstructing vehicle routes that explores many possible candidate vehicle locations for each GPS datapoint, and explicitly considers the travel between pairs of candidate locations. This original algorithm is robust to a wide range of inputs, can penalise deviations between observed and

modelled travel times and travel distances, and can exploit serial correlation in the errors in successive GPS datapoints.

In this paper we present a new faster, simpler algorithm. To avoid repetition of the detail in [3], we present our algorithm as the more general problem of building a route on some given road network that minimises the total distance travelled less the accumulated value of the GPS datapoints collected along the route. A GPS datapoint can be collected by traversing an arc that passes close to the datapoint's location. The value realised by collecting a GPS datapoint diminishes rapidly as the distance between the datapoint and the point of collection increases. Although the datapoints must be collected in order, it is possible to skip the collection of some datapoints (typically because they have very large positional errors placing them far from the vehicle path).

Our new algorithm attempts to reduce the running time required for the map matching by exploiting the near-shortest path nature we expect in the underlying vehicle routes. In developing this algorithm, we have focused on using this assumption to modify the standard Dijkstra algorithm to meet our map-matching needs. There has been a lot of research work focused on developing highly efficient implementations of Dijkstra's algorithm for large road networks (e.g. [1]); our goal is to exploit this for maximum computational efficiency.

We develop our algorithm by first presenting a new dynamic programming view of the map-matching problem. Our original dynamic programming model explicitly considered (and penalised appropriately) travel between pairs of candidate vehicle locations associated with GPS datapoints. Our new dynamic programming model differs from that presented earlier in that it does not explicitly model travel between candidate vehicle locations, but instead develops a shortest path tree that tracks datapoint collections using a state-expanded version of the underlying road network.

2 A State Expanded Road Network

Our first dynamic programming approach is to introduce states that store information on the GPS points collected, and to expand the road network to include this state information. We assume our road network is represented by a set of vertices $v \in V$ and directed edges $e \in E$, where edge e goes from a head node $h(e) \in V$ to a tail node $t(e) \in V$. Edge e has an associated positive distance (or travel time) $d(e) > 0$. To model the GPS datapoint collection, we let $b_c(e)$ be the ('bonus') value realised by collecting GPS datapoint g_c , $c = 1, 2, 3, \dots, n$ when traversing arc e , $e \in E$. (In calculating $b_c(e)$, we assume the datapoint collection occurs at the closest point on e to g_c .) If arc e is too far from g_c to collect the datapoint (i.e. outside g_c 's 'collection area'), then $b_c(e) = 0$. We say that e is a collection arc for g_c (or equivalently that e can collect g_c) if $b_c(e) > 0$. An arc is termed a 'collection arc' if it collects any g_c , $c \in \{1, 2, \dots, n\}$. Note that if e is used to collect g_c in some solution, then e contributes $d(e) - b_c(e)$ to the objective. Because $d(e) - b_c(e)$ can be negative, a simple shortest path approach is likely to fail because of cycling.

To construct our dynamic program, we expand the road network defined by (V, E) by adding a state $c \in \{0, 1, 2, \dots, n\}$ to each node, where being at a node v in state c , denoted by $[v, c]$ in the expanded state network, indicates that we have

collected the c 'th GPS point, and thus none of the GPS points g_1, g_2, \dots, g_c are able to be collected in any continuation from $[v, c]$. We think of c as being a height, where GPS point g_c is typically collected by travelling from height $c - 1$ to height c in the network. For each arc e in the road network, our expanded network contains $n + 1$ copies of this arc going from $[t(e), c]$ to $[h(e), c]$ for all states $c = 0, 1, 2, \dots, n$, with each copy having distance $d(e)$. The collection of a GPS point while traversing arc e is handled in the expanded network by including additional copies of arc e going from one state to a higher state. Specifically, our network contains arcs from $[t(e), c_{tail}]$ to $[h(e), c]$ for all $c_{tail} = 0, 1, 2, \dots, c - 1$, each with (potentially negative) distance $d(e) - b_c(e)$. (In practice, we only include a collection arc if the corresponding road network arc e is sufficiently close to GPS datapoint g_c to gain a strictly non-zero collection value $b_c(e) > 0$.) We note that our network has no downward-directed arcs, and that the (potentially) negative arc lengths occur only on arcs going upwards. Thus, any shortest path in this expanded network will be acyclic.

We can now represent our dynamic program using the following recursion. Let $f[v, c]$ be the optimal objective function value associated with being at node v in state c (i.e. having collected GPS datapoint g_c). Now,

$$f[v, c] = \min \left\{ \begin{array}{l} \min_{e:h(e)=v} f[t(e), c] + d(e), \\ \min_{e, c_{start}:h(e)=v, c_{start}=0,1,2,\dots,c-1} f[t(e), c_{start}] + d(e) - b_c(e) \end{array} \right\}$$

We assume we know the starting node v_{start} and finish node v_{end} of the trip, and so set $f[v_{start}, 0] = 0$, and $f[v, 0] = \infty$ for all $v \neq v_{start}$. We seek to minimise $f[v_{end}, n]$, being the total travel distance, less the value of any GPS datapoints collected, at the end node. (We note that if the start and/or end node is not known, then other start and finish conditions are possible. For example, if we put $f[v, 0] = 0 \forall v \in V$ then all nodes become start points, and we can then seek to minimise $\min_v (f[v, n])$ allowing any node as an end point. This can be further generalised to allow any $[v, c]$ to be a start and/or end point, or indeed to allow a start and/or end point to occur partway along an arc.)

3 Non-Backtracking Solutions

There are a number of inefficiencies that arise when we solve this new formulation using Dijkstra's algorithm (or any equivalent dynamic programming solver). These include the handling of arcs with negative distances while retaining the efficiency of Dijkstra's algorithm; we address this particular issue later. At this point, let us consider the 1-dimensional example in Figure 1. This figure shows nodes $V = \{v_1, v_2, \dots\}$ and arcs $E = \{e_1, e_2, \dots\}$, which form a subset of a larger network. (Where an arc direction is not shown, it implies there are arcs in both directions. However, for convenience our arc names e_1, e_2, \dots refer to those arcs directed from left to right.) We assume $d(e) = 1 \forall e \in E$. We wish to start at node v_1 , and finish at some node v_{20} (not shown), giving an optimal path of $([v_1, 0], [v_2, 0], [v_3, 0], [v_4, 0], [v_5, 1], [v_6, 1], \dots$. We notice that datapoint g_1 should be collected as we traverse e_4 .

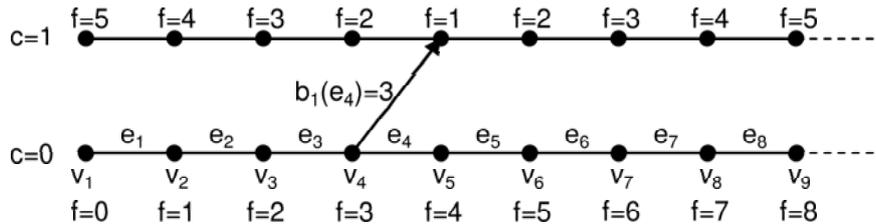


Figure 1: A simple 1D example of a state-expanded network for map matching; all road arcs are unit length. Where arc directions are not shown, two arcs exist in opposite directions. The search starts at node v_1 . The objective function values $f[v, c]$ for each node v in states $c = 0$ and $c = 1$ are shown (denoted ‘ $f =$ ’) beside each node.

To solve this problem, we start by labelling $f[v_1, 0] = 0$. From this node, we then label $f[v_2, 0] = 1$, $f[v_3, 0] = 2$ and $f[v_4, 0] = 3$. The arc e_4 from v_4 to v_5 collects the 1st GPS datapoint g_1 , and so, assuming a collection value of $b_1(e_4) = 3$, we get $f[v_5, 1] = f[v_4, 0] + d(e) - b_1(e_4) = 1$, as well as $f[v_5, 0] = 4$. We then continue to label nodes v_6, v_7, \dots in both states $c = 0$ and $c = 1$; this is inefficient as these $c = 0$ labels do not lie on the optimal path, and so are never used. We also apply labels $f[v_4, 1] = 2$, $f[v_3, 1] = 3$, \dots in state $c = 1$, which again is inefficient as these labels are never used.

Consider first the labelling of v_5, v_6, v_7, \dots in both states $c = 0$ and $c = 1$. In this example, the state $[v_5, 1]$ clearly dominates $[v_5, 0]$ in the simplest since of its objective being better ($f[v_5, 1] < f[v_5, 0]$) because it has collected the 1st GPS point. In this case, there is no advantage in being at v_5 in the state $c = 0$ as we will never exploit the benefit that this state provides of being able to collect g_1 at some later stage. It is tempting to use dominance to say that the $[v_5, 0]$ label can be discarded as it is dominated by $[v_5, 1]$. However, under this approach, $[v_4, 1]$ would appear to dominate $[v_4, 0]$, and hence $[v_4, 0]$ would also be discarded. Unfortunately, deleting $[v_4, 0]$ is not appropriate as it does lie on the optimal path, and hence contains information (such as a predecessor arc) that is required to construct the optimal solution.

To address these difficulties, we note that the map-matched routes we are attempting to re-construct have been chosen by intelligent drivers, and thus are typically near-shortest paths. Therefore, we wish to restrict our solutions to near shortest paths. While there are many possible interpretations of a near-shortest path, our requirements here are (roughly speaking) that our map-matched route never ‘backtracks’. We say a route R starting at node v_{start} is non-backtracking if, for any pair of nodes $(v_{[i]}, v_{[j]})$ such that $v_{[j]}$ follows node $v_{[i]}$ in R , the shortest path distance from the start node v_{start} to node $v_{[j]}$, $d(v_{start}, v_{[j]})$, is greater than that for node $v_{[i]}$, i.e. $d(v_{start}, v_{[j]}) > d(v_{start}, v_{[i]})$. We can also describe our GPS datapoints g_1, g_2, \dots, g_n as being non-backtracking by which we mean the collection arcs for successive datapoints occur at increasing distance from v_{start} . More formally, given any two collection arcs $e_i : b_p(e_i) > 0$ and $e_j : b_q(e_j) > 0$ with $p < q$, we require $d(v_{start}, h(e_i)) < d(v_{start}, t(e_j))$. (We also require that the end node, v_{end} , is located at a greater distance than the collection points for g_n ; i.e. given any $e : b_n(e) > 0$,

$d(v_{start}, h(e)) < d(v_{start}, v_{end})$.) If the GPS datapoints are non-backtracking, then we expect our map-matched routes to also be non-backtracking. We will restrict ourselves to such cases.

Consider again our example of Figure 1. If the GPS datapoints are non-backtracking, then once we have collected GPS point g_1 (which occurred as we reached a distance of 4 units from the start node), our search for later GPS datapoints can ignore any of the nodes $v \in V$ at distances 4 or less from the start. Because Dijkstra's algorithm applies permanent labels in order of increasing distance, the set of nodes we wish to remove from the next phase of our search are simply those that have been permanently labeled (being in this case those nodes at distances of 4 or less). If our modified algorithm includes logic to prevent permanently labeled nodes from being re-labeled, the $c = 1$ label $[v_4, 1]$ will never be applied to node v_4 (and indeed no $c = 1$ labels will be applied to v_3 , v_2 , or v_1). This 'no-backtracking' rule has removed these unwanted labels. However, it also has the added advantage that the issue of which label(s) to keep does not now arise for these nodes, and so we can choose a dominance rule that does not need have to handle this case. Indeed, our simple objective-based dominance rule proposed above performed well for nodes v_5 , v_6 etc., and so we can adopt this rule. This means that (in general) each node can be labeled with just one distance label (or, more correctly, one 'distance less collection values' label), along with an associated state that is used to ensure GPS datapoints are collected no more than once and in the correct sequence. Thus, the non-backtracking solution assumption allows us to construct a tree on the road network instead of a more complicated solution structure, with the optimal map-matched route then occurring as a path in this tree. As we will see, we can do this in much the same way as a shortest path tree is constructed. (Note that our search does not build a true shortest path tree, but instead the tree's 'distances' are modified by the collection of GPS datapoints, giving a tree that is distorted around these datapoints. We require that the solution be non-backtracking with respect to a distance metric defined by the order in which nodes are permanently labeled.)

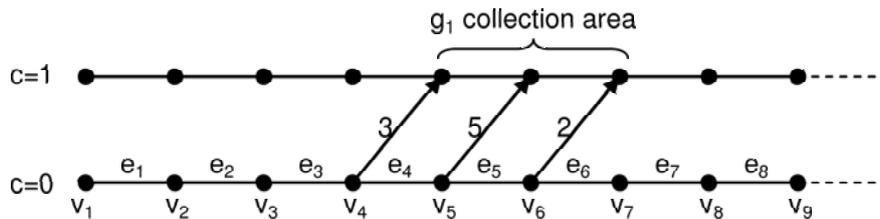


Figure 2: A simple 1D example of a state-expanded network with 3 alternative collection arcs, e_4 , e_5 and e_6 , for GPS datapoint g_1 . Each collection arc e realises a different value $b_1(e)$ as a result of the collection.

While keeping just a single 'distance, state' label on each node works for most of the network, it fails in the collection area around a GPS datapoint. Consider the example in Figure 2, where g_1 can be collected via e_4 , e_5 or e_6 . Of these 3 choices, the e_5 collection realises the greatest value of 5 units. Our approach must be capable of finding this best option while making sure it does not prematurely

collect the datapoint via e_4 (as would happen under our dominance rule); this requires the tracking of multiple states (in this case states $c = 0$ and $c = 1$) during exploration of the collection area surrounding the GPS datapoint. To ensure this state information is maintained, our simple objective dominance must be disabled for the nodes surrounding g_1 (in this case nodes v_5 and v_6). Once we have left the collection area, multiple states can be reduced to just one state using our objective based dominance rule. Thus, using our no-backtracking rule, and using dominance outside the collection area, the search space we actually explore is as depicted in Figure 3.

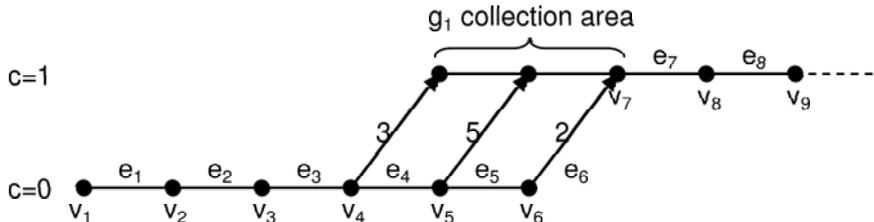


Figure 3: A depiction of the state-space as explored by our non-backtracking Dijkstra search using dominance. Multiple states need to be explored only in the collection area.

To summarise the model so far, by assuming a no-backtracking rule, we have simplified the state space so that, for most nodes, we can discard via dominance all states except that state with the smallest distance label (where this distance label records the total distance travelled less the accumulated value of any GPS datapoint collections made). Thus, each node needs to record a single state label and a single associated distance label. The only exception to this is for nodes located in the collection area around a GPS datapoint where multiple states must be maintained.

In our implementation of this algorithm, we can make a further refinement by searching for each successive GPS datapoint in turn without having to use states to track which particular datapoint has been collected at each node. Thus, our search becomes a series of staged search tree expansions, with a new expansion being started to find the next GPS datapoint each time the tree has been grown sufficiently to collect the previous GPS datapoint. This means that, with the exception of nodes in the datapoint collection areas, only distance (not state) labels must be maintained on the nodes. This gives the major advantage that state information can be discarded from (most) nodes, leaving an essentially Dijkstra-based search with minimal modifications.

At the cost of some possible loss of optimality, we can also remove the need for multiple states from the collection area nodes, giving a state-free algorithm. To achieve this, we again rely on near-shortest paths as follows. Let us assume that GPS datapoint g_{c-1} has been collected, and we are currently targeting the collection of datapoint g_c . We allow our non-backtracking Dijkstra search process to expand our search tree in the normal way. As this search tree grows, it will pass through the collection area for g_c , which we allow to happen as if the GPS datapoint was not present. No datapoint collection is made while the tree is being built through these collection arcs. Then, once the Dijkstra search frontier has gone

past these arcs (i.e. when the head nodes of all these arcs have been permanently labeled), the Dijkstra search is paused, and those paths in the search tree that include possible collection arcs are explored to determine which arc in the path gives the best datapoint collection value. Once the best datapoint collection arc has been found in a path, the distance label of the path's current end node is reduced by the value realised by this collection. When all possible collection paths have been checked (and updated) in this way, the Dijkstra search is then resumed with the next target collection datapoint being g_{c+1} . This search uses the new distance labels as updated with the datapoint collection values $b_c()$, and thus gives a search tree that is biased towards the GPS datapoint g_c as required. Note that the issue of using Dijkstra with negative arc lengths disappears as we are essentially re-starting Dijkstra with a set of initial node labels, some of which happen to be negative.

The following algorithms can be used to implement our search. Algorithm 1, `InitialiseSearch`, is used to initialise structures associated with the tree T that we will build. Algorithm 2, `ScanForNextDatapoint`, is called repeatedly to grow the tree T out to the next GPS datapoint. This algorithm is modified from the standard version so that no label updates are made for nodes with the tag 'permanent' (meaning that a permanent distance label has been applied). The algorithm returns after scanning all arcs that can collect the next GPS datapoint g_c . (To be more precise, the algorithm returns when the nodes at the heads of all possible collection arcs have been permanently labeled.) This `ScanForNextDatapoint` algorithm also constructs data structures that are needed for Algorithm 3, `RealiseCollection`. This `RealiseCollection` algorithm is called after each execution of `ScanForNextDatapoint` to explore those paths in the expanded tree that contain arcs that can be used to collect the GPS datapoint g_c . This algorithm scans back through each path from its leaf node v looking for the arc e that gives the best collection value $b_c(e)$, and then updates the distance label $d_T(v) := d_T(v) - b_c(e)$ on v . Each leaf node that has its distance label reduced is then re-inserted into the list of partially scanned nodes, ensuring that these nodes are re-scanned when growing of the search tree continues. These 3 algorithms are called from the final algorithm, Algorithm 4 'FastMapMatching'.

InitialiseSearch(T)

```

for all  $v \in V$ 
  set  $d_T(v) = \infty$ 
  set  $\text{CanCollect}_T(v) = \text{false}$ 
  set  $\text{Permanent}_T(v) = \text{false}$ 
let  $d_T(v_{start}) = 0$ 
let  $\text{pred}_T(v_{start}) = \text{NULL}$ 
let  $Q_T = \{v_{start}\}$ 

```

Algorithm 1: 'InitialiseSearch'. This algorithm initialises the data required for search tree T , being node distance labels $d_T(v)$, node predecessor arc labels $\text{pred}_T(v)$, partially scanned node list Q_T , and node tags $\text{CanCollect}_T(v)$ and $\text{Permanent}_T(v)$.

ScanForNextDatapoint(T, c)

Find our set of destination nodes, and check they are not already permanently labeled

let $H = \{h(e) : e \in E, b_c(e) > 0\}$

if $(\exists v \in H : \text{Permanent}_T(v) = \text{true})$ then

stop(“The datapoints fail the non-backtracking assumption”)

let $L_T = ()$

Scan repeatedly until all target nodes are permanently labeled

while $(\exists v \in H : \text{Permanent}_T(v) = \text{false})$ do

extract node v with smallest $d_T(v)$ from Q_T

let $\text{Permanent}_T(v) = \text{true}$

’ If the path to v can collect g_c , add v to L_T and tag v ‘CanCollect’

let $e_{pred} = \text{pred}_T(v)$

let $v_{pred} = t(e_{pred})$

if $(b_c(e_{pred}) > 0$ or $\text{CanCollect}_T(v_{pred}))$ then

$\text{CanCollect}_T(v) = \text{true}$

$L_T = L_T + v$

Scan all arcs leaving v , updating only non-permanently labeled nodes

for all arcs $e \in E : t(e) = v$ do

let $v_{head} = h(e)$

if $\text{Permanent}_T(v_{head}) = \text{false}$ then

let $d^* = d_T(v) + d(e)$

if $d^* < d_T(v_{head})$

let $d_T(v_{head}) = d^*$

let $\text{pred}_T(v_{head}) = e$

insert v_{head} into Q_T

Algorithm 2: ‘ScanForNextDatapoint’. Note that L_T is an ordered list associated with tree T , where $L_T = L_T + v$ means append node v to the end of L_T . The set Q_T contains the partially scanned nodes, and is often implemented as a priority list to enable rapid extraction of that node v with the smallest distance label $d_T(v)$.

4 Conclusions

We have presented a new algorithm for solving map-matching problems that reduces the computational effort required by exploiting the ‘non-backtracking’ property we expect to see in the resulting map-matched routes. This new algorithm will detect routes that contain back-tracking, for which we will need to use our earlier slower algorithm. Computational experiments are now being conducted to evaluate the effectiveness of this new approach. These will be reported shortly.

RealiseCollections(T, c)
Delete all non-leaf nodes from list L_T
for all $v \in L_T$
 delete $t(\text{pred}_T(v))$ from L_T
Scan all leaf nodes, tracing back from each to find the best collection to make
for all ($v \in L_T$) do
 $v' = v$
 $b = 0$
 repeat
 let $e_{pred} = \text{pred}_T(v')$
 let $b = \max(b, b_c(e_{pred}))$
 let $v' = t(e_{pred})$
 until $\text{CanCollect}_T(v') = \text{false}$
 'Accumulate the collection value in v 's distance label
 let $d_T(v) = d_T(v) - b$
 'Node v 's distance label has reduced; ensure it is re-scanned
 insert v into Q_T
Reset the CanCollect labels
for all $v \in V$
 let $\text{CanCollect}_T(v) = \text{false}$

Algorithm 3: 'RealiseCollections'. This algorithm is used to realise the collection of GPS datapoint g_c . On input, the node list L_T contains the nodes in each shortest path in the tree T that can potentially collect GPS datapoint g_c . This algorithm traces back from the last permanently-labeled node in each of these paths to find the best collection value for datapoint g_c , and updates $d_T()$ and Q_T ready for continued search.

FastMapMatching
InitialiseSearch(T)
for $c = 1, 2, \dots, n$ do
 ScanForNextDatapoint(T, c)
 RealiseCollections(T, c)
ScanForNode(T, v_{end})

Algorithm 4: 'FastMapMatching'. Note that $\text{ScanForNode}(T, v_{end})$ performs our no-backtracking Dijkstra search starting with the tree T and continuing until node v_{end} is labeled. After this is completed, the map-matched route can be found by tracing back from v_{end} in the normal way.

5 Acknowledgements

The author would like to acknowledge the collaborative work undertaken with colleagues from Optima (www.theoptimacorporation.com) while developing the first version of the map-matching algorithm.

References

- [1] S. Edelkamp, S. Jabbar, and T. Willhalm. Geometric travel planning. *IEEE Transactions on Intelligent Transportation Systems*, 6:5–16, 2005.
- [2] F. Marchal, J. Hackney, and K.W. Axhausen. *Efficient Map-Matching of Large GPS Data Sets - Tests on a Speed Monitoring Experiment in Zurich*, volume 244 of *Arbeitsbericht Verkehrs und Raumplanung*. ETH Zürich, Zürich, 2004.
- [3] Andrew J Mason. Emergency vehicle trip analysis using GPS AVL data: A dynamic program for map matching. In *Proceedings of the 40th Annual Conference of the Operational Research Society of New Zealand*, pages 295–304, December 2005.
- [4] Otto Anker Nielsen. Map-matching algorithms for GPS data - methodology and test on data from the AKTA roadpricing experiment in Copenhagen. Technical report, Centre for Traffic and Transport (CTT), Technical University of Denmark (DTU), 2004.
- [5] K.W. Axhausen and S. Schönfelder, J. Wolf, M. Oliveira, and U. Samaga. Eighty weeks of GPS traces: Approaches to enriching trip information. Technical report, IVT / ETH, CH 8093 Zürich, 2003. Presented at the 83rd Transportation Research Board Meeting, Washington, D.C., Jan. 11-15, 2004.
- [6] The Optima Corporation Ltd. Simulation for Improving Responses in Emergency Networks (Siren) Software, 2005. www.TheOptimaCorporation.com.

Dynamic Outer-Approximation Sampling Algorithm

Ziming Guan
Engineering Science Department
University of Auckland
New Zealand
z.guan@auckland.ac.nz

Abstract

We describe an algorithm called Dynamic Outer-Approximation Sampling Algorithm (DOASA) to solve multistage stochastic quadratic programming problems that have some random factors with inter-stage independence. The algorithm which is similar to those in the same class (AND, CUPPS, ReSa and SDDP) can be proved to converge almost surely, and performs well on instances with low state dimension. We describe the application of the algorithm to a production and sales planning problem for dairy products.

1 Introduction

In the real world, numerous problems can be formulated as multistage stochastic programs. Such problems have a multistage horizon, for example 12 months in a production season. At the beginning of each stage, some random quantities become realized and they are observable. In previous stages, however, these random quantities are unknown, but occur with probabilities which are known. In each stage, after observing the random quantities, an action can be taken. Such an action aims to minimize the total cost in that stage, which is composed of the deterministic cost in the current stage and the expected cost in future stages. The ultimate goal is to minimize the total cost of our problem by taking the best action at the beginning of the first stage.

Multistage stochastic problems are difficult to solve. The most successful and practicable algorithms is a class of sampling-based decomposition algorithms, which is called *Multi-stage Sampled Benders Decomposition (MSBD)* in Linowsky and Philpott [5]. This class of algorithms iteratively sample realizations of random quantities and use Benders cuts to approximate subproblems in stages. Stochastic Dual Dynamic Programming (SDDP) was developed by Pereira and Pinto [1]. Based on similar ideas, the Convergent Cutting-Plane and Partial-Sampling algorithm (CUPPS) was developed by Chen and Powell [2], the Abridged Nested

Decomposition (AND) by Donohue [3], and the Reduced Sampling method (ReSa) by Hindsberger and Philpott [4]. These algorithms have been successfully applied to linear problems. Based on the similar idea, we aim to develop an algorithm to solve quadratic problems and apply it to a production and sales planning problem for dairy products.

2 Multistage decomposition

In this paper, we focus on the multistage stochastic quadratic problems with the following five underlying assumptions:

1. Random quantities are interstage independent.
2. The sets of random quantities are discrete and finite in each stage.
3. The feasible region of the quadratic problem in each stage is non-empty and bounded.
4. Each stage problem has a quadratic objective, and linear constraints.
5. In the objective, there is no cross term of variables in the quadratic terms, and interstage dependent variables (state variables) appear only in the linear terms and constraints.

Under these assumptions, the multistage stochastic quadratic problem can be written in the following form.

For $t = 1$, $[QP_1]$ is the quadratic problem

$$\begin{aligned} Q_1 &= \min_{x_1} \frac{1}{2} x_1^T D_1 x_1 + g_1^T x_1 + \tilde{Q}_2(x_1) \\ \text{s.t. } A_1 x_1 &= b_1 \\ x_1 &\geq 0, \end{aligned}$$

and $[QP_t]$ is the quadratic problem

$$\begin{aligned} Q_t(x_{t-1}, w_t) &= \min_{x_t} \frac{1}{2} x_t^T D_t x_t + g_t^T x_t + \tilde{Q}_{t+1}(x_t) \\ \text{s.t. } A_t x_t &= w_t - B_{t-1} x_{t-1} \\ x_t &\geq 0, \end{aligned}$$

where for all $t = 2, \dots, T$,

$$\tilde{Q}_t(x_{t-1}) = \sum_{w_t \in W_t} p_t(w_t) Q_t(x_{t-1}, w_t),$$

and we set $\tilde{Q}_{T+1} \equiv 0$.

The problem $[QP_t]$ depends on the variable x_{t-1} , and the random quantity w_t . $p_t(w_t)$ is the probability of realization of $w_t \in W_t$. For simplicity, we have assumed that B_t , D_t and g_t do not depend on w_t , although this assumption can be relaxed.

The matrix D_t is a non-negative diagonal matrix with diagonal entries being positive for some decision variables and zeros for others including state variables. It

is positive semi-definite, so $\frac{1}{2}x_t^\top D_t x_t$ is convex. The term $g_t^\top x_t$ is linear and obviously it is convex. Since at any t , x_{t-1} is on the right hand side of the constraints in the problems in the next stage, we have that $Q_t(x_{t-1}, w_t)$ is convex for each w_t as long as \tilde{Q}_{t+1} is convex. But \tilde{Q}_{t+1} is a linear combination of Q_{t+1} and so we have \tilde{Q}_{t+1} being convex for every t by induction.

The function $\tilde{Q}_{t+1}(x_t)$ can be approximated by a collection of linear functions, which are called *cuts*, that gives an outer approximation to the problem. In each stage, $[QP_t]$ can be approximated by a sequence of approximate problems $[AP_t^k]$. $[AP_t^k]$ for iteration k is written as follows:

For $t = 1$, $[AP_1^k]$ is the quadratic problem

$$\begin{aligned} C_1^k &= \min_{x_1, \theta_2} \frac{1}{2}x_1^\top D_1 x_1 + g_1^\top x_1 + \theta_2 \\ \text{s.t. } A_1 x_1 &= b_1 \\ \theta_2 + (\beta_2^j)^\top x_1 &\geq \alpha_2^j \\ x_1 &\geq 0, \end{aligned} \quad (j = 0, \dots, k)$$

and for $t = 2, \dots, T-1$, $[AP_t^k]$ is the quadratic problem

$$\begin{aligned} C_t^k(x_{t-1}, w_t) &= \min_{x_t, \theta_{t+1}} \frac{1}{2}x_t^\top D_t x_t + g_t^\top x_t + \theta_{t+1} \\ \text{s.t. } A_t x_t &= w_t - B_{t-1} x_{t-1} \\ \theta_{t+1} + (\beta_{t+1}^j)^\top x_t &\geq \alpha_{t+1}^j \\ x_t &\geq 0, \end{aligned} \quad (j = 0, \dots, k)$$

and for $t = T$, $[AP_T] \equiv [QP_T]$

$$\begin{aligned} C_T(x_{T-1}, w_T) &= \min_{x_T} \frac{1}{2}x_T^\top D_T x_T + g_T^\top x_T \\ \text{s.t. } A_T x_T &= w_T - B_{T-1} x_{T-1} \\ x_T &\geq 0, \end{aligned}$$

where for all $t = 2, \dots, T$,

$$\begin{aligned} \beta_t^j &= \sum_{w_t \in W_t} p_t(w_t) \pi^j(x_{t-1}^j, w_t^j) \\ \alpha_t^j &= \sum_{w_t \in W_t} p_t(w_t) (C_t^j(x_{t-1}^j, w_t^j) - \pi^j(x_{t-1}^j, w_t^j)^\top x_{t-1}^j) \end{aligned} \quad (j = 0, \dots, k),$$

where $\pi^j(x_{t-1}^j, w_t^j)$ is a subgradient to $C_t^k(\bullet, w_t)$ at x_{t-1}^j . For all stages, the starting cut is set to a trivial cut $\theta_{t+1}^0 \geq -\infty$.

A QP satisfies Karush-Kuhn-Tucker conditions at its optimal solution. According to Wright [6], the KKT conditions of our convex QP can be stated as:

$$\begin{bmatrix} s \\ 0 \end{bmatrix} = \begin{bmatrix} I & -A_t^\top \\ A_t & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} + \begin{bmatrix} g_t \\ -w_t + B_{t-1} x_{t-1} \end{bmatrix}, \quad (x, s) \geq 0, \quad x^\top s = 0,$$

where s is a vector of slack variables, and λ is the Lagrangian multiplier of the KKT conditions. This enables our QP to be solved using primal-dual interior-point methods, for example the barrier optimizer in CPLEX. This yields a vector λ of Lagrangian multipliers at optimality. Then we have $\pi^k(x_{t-1}^k, w_t^k) = -(\lambda^k(x_{t-1}^k, w_t^k))^\top B_{t-1}$.

Since the $Q_t^k(x_{t-1}^k, w_t^k)$ is convex and $\pi_t^k(x_{t-1}^k, w_t^k)$ is a subgradient at x_{t-1}^k , the following property holds:

$$Q_t^k(x_{t-1}, w_t^k) \geq Q_t^k(x_{t-1}^k, w_t^k) + (\pi_t^k(x_{t-1}^k, w_t^k))^\top (x_{t-1} - x_{t-1}^k).$$

Then an expectation cut is constructed by taking expectation on $w_t^k \in W_t$ on both sides, which becomes

$$E [Q_t^k(x_{t-1}, w_t^k)] \geq E [Q_t^k(x_{t-1}^k, w_t^k) - (\pi_t^k(x_{t-1}^k, w_t^k))^\top x_{t-1}^k] + E [(\pi_t^k(x_{t-1}^k, w_t^k))^\top x_{t-1}],$$

and this yields the cut coefficients in $[AP_t^k]$ by replacing corresponding terms in $[AP_t^k]$.

Since cuts are added iteratively and no cut is taken out, $[AP_t^k]$ is a more restricted problem than $[AP_t^{k-1}]$, thus $C_t^k(x_{t-1}, w_t) \geq C_t^{k-1}(x_{t-1}, w_t)$, which means $C_t(x_{t-1}, w_t)$ is monotonic non-decreasing. It provides a lower bound of the true Q_t , and converges to Q_t .

3 DOASA

In this section, we present the core idea of our DOASA algorithm for solving $[QP_1]$ as below.

- Forward pass
 - Randomly sample random quantities w_t^k for each stage to construct a random sample path $\{w_t^k\}$ for $t = 2, \dots, T$.
 - Solve $[AP_t^k]$ in order of $t = 1, \dots, T - 1$, to obtain feasible solutions x_t^k in each stage.
- Backward pass/Cut generation
 - $[AP_t^k]$ is solved in order of $t = T, T - 1, \dots, 2$.
 - In each stage, $[AP_t^k]$ is solved for each possible realization of random quantities.
 - One cut is constructed with the cut coefficients α_t^k and β_t^k calculated in a way shown in the formulation in the previous section, and the cut is passed to the previous stage.
- Stopping criteria/Convergence checking
 - After $[AP_2^k]$ is solved in the backward pass, $[AP_1^k]$ is solved and C_1^k is an estimated lower bound of Q_1 . Check if $C_1^k - C_1^{k-1} \leq \tau C_1^{k-1}$, where τ is a pre-defined tolerance in percentage. This is to check the stability of the estimated lower bound.
 - We run the algorithm for N iterations.

- Within N iterations, if the stability check is passed, and this holds for a pre-defined number of iterations, then convergence is claimed and the algorithm stops. The total cost at the first stage problem is an estimated lower bound of Q_1 .
- Post-solution checking by simulation
 - A large number of random paths are sampled and their respective probabilities of realizations are recorded.
 - Problems for each sampled path are solved.
 - The average cost weighted by probability of sampled paths is an estimated upper bound of Q_1 . The standard error is also computed.
 - The estimated upper bound is compared to the estimated lower bound to see the goodness of the solution.

4 Convergence of DOASA

DOASA has the same structure as the algorithms in the class of MSBD. The convergence property of this class of algorithms is addressed in Linowsky and Philpott [5]. Two convergence properties are defined in the literature:

Definition 3 (Cut Sampling Property)

MSBD is said to fulfill the cut-sampling property (CSP) if for each stage t , $\{k \mid \Omega_t^k = \emptyset\}$ is finite.

Definition 4 (Sample Intersection Property)

MSBD is said to fulfill the sample-intersection property (SIP) if for each stage t , and every outcome $\omega_{ti} \in \Omega_t$, $\Pr[(\omega_{ti} \in \Omega_t^k) \cap (\omega_{ti}^k = \omega_{ti})] > 0$ for every k with $\Omega_t^k \neq \emptyset$.

The Cut Sampling Property states that there are limited iterations with empty samples in the backward pass. The Sample Intersection Property states that samples in the forward pass and the backward pass intersect in iterations with nonempty samples in the backward pass with a positive probability. If MSBD satisfies these two convergence properties then it converges with probability of 1. (Proofs are given in [5].) DOASA samples one random path in the forward pass, and constructs a cut by solving all subproblems in the full sample space. Thus DOASA meets those two convergence properties and converges almost surely.

The algorithms in MSBD have a multiple path scheme in the forward pass to obtain multiple feasible solutions and so generate multiple cuts in each iteration. DOASA samples only one random path in the forward pass, which is called a *single path scheme*, and it generates only one cut in each iteration. DOASA displays fast convergence with problems with low state dimension, which is shown in the next section.

5 Application of DOASA to a production and sales planning problem for dairy products

In this section, we aim to present a production and sales planning problem for dairy products and apply the DOASA.

Fonterra Co-operative Group Ltd is New Zealand's multinational dairy company. They have a dedicated system to manage their production and sales planning. One of their major concerns is the variation in milk supply during a 12-month season. They reforecast milk supply every month and make their production and sales planning for the rest of season. However, forecasting of milk supplies is never perfect due to random effects, for example, weather. More or less milk supply will force a change in a pre-designed plan and incurs cost. To resolve this problem, we formulate this problem as a multistage stochastic quadratic problem. Observing the variation of milk supply occurs mainly in month 6 to 12 in the historical data, we decide to formulate a stochastic problem from month 5 to 12.

A brief description of Fonterra's production and sales structure in our problem is as below, with information from [7]:

- In each month, milk is collected from farms in six dedicated regions. Milk volume varies in different regions and over different months.
- Milk is sent to five different types of factories to produce nine different products, which incur production costs.
- Products from plants are delivered to stores, which incurs inland transportation costs.
- Stores carry products from the previous month.
- Products in stores are
 - either sold to four international markets to generate revenue while incurs overseas transportation costs,
 - or stored to be sold in later months, which incurs storage costs. Products in stores are called *Inventory*.
- Inventory at the end of season can be carried over to the next season to generate revenue. In our problem, we give them some multipliers to present the value from carry-over.
- Decisions must be made on the allocation of milk to the factories and control of inventory and sales.
- The objective is to minimize the seasonal cost, which is the costs in production, storage and transportation less the revenue from sale.

The formulation of stage problems is given as below.

$$\begin{aligned}
Q_t(y_{t-1}, v_{t-1}, w_t) &= \min_{x_t, y_t, z_t} -x_t^\top F_t(x_t) + c_{x_t}^\top x_t + c_{y_t}^\top y_t + c_{z_t}^\top G_t(z_t) + \tilde{Q}_{t+1}(y_t, v_t) \\
\text{s.t.} \quad x_t + y_t - G_t(z_t) &= y_{t-1} \\
s_t &= R_t(v_t) \\
v_t &= \rho v_{t-1} + w_t \\
w_t &\sim \text{Normal}(0, \Sigma) \\
z_t &\leq s_t \\
x_t &\leq b_{x_t} \\
y_t &\leq b_{y_t} \\
z_t &\leq b_{z_t} \\
x_t, y_t, z_t, F_t(x_t), G_t(z_t) &\geq 0
\end{aligned}$$

where x_t , y_t and z_t are variables of sales, inventory and input to factories respectively, of which y_t is a state variable; $F_t(x_t)$ is linear price-demand function of x_t , and $G_t(z_t)$ is production from factories which is a linear transformation of input z_t ; s_t is milk supply, v_t is the second state variable, w_t is a random quantity vector with zero mean and covariance matrix Σ , and $R_t(v_t)$ is a linear transformation of v_t ; c_{x_t} , c_{y_t} and c_{z_t} are transportation, storage and production costs; b_{x_t} , b_{y_t} and b_{z_t} are bounds on x_t , y_t and z_t .

Constraint 2 states production, sales and inventory must be balanced. Constraints 3 to 5 state milk supply is determined by a state variable and a random quantity with interstage independence. Constraint 6 states milk supply provides an upper bound on input to factories. Constraints 7 to 9 states some upper bounds on sales, inventory and input. And Constraint 10 states that sales, inventory, input, price and production are non-negative.

DOASA solves the problem with solver AMPL/CPLEX 10.0. With 9 state variables in y_t plus 6 state variables in v_t , and 121 realizations for w_t in each stage, DOASA converges within 13 iterations in 13 minutes with thresholds in stopping criteria being 0.1% of deviation for stability of the estimated lower bound. A post-solution simulation shows the estimated lower bound and upper bound are close with an acceptable deviation. For comparison, a deterministic problem with the same production and sales structure but with one determined path is solved, where the path is the expectation of random quantities. Its solution is tested against the one from DOASA in post-solution simulation. DOASA generates less costs on average and out-performs the deterministic method.

6 Conclusion

DOASA is an algorithm to solve multistage stochastic quadratic problems. It has a similar structure to the algorithms in the MSBD class, which includes AND, CUPPS, ReSa and SDDP. It approximates the quadratic problems in each stage as a sequence of approximate problems using Benders cuts and converges almost surely. We have applied DOASA to a production and sales planning problem for dairy products. It displays fast convergence, and out-performs the deterministic method in post-solution simulation.

7 Acknowledgments

I would like to express great thanks to my supervisor Professor Andrew Philpott for his supervision on my work. I would like to thank Fonterra Co-operative Group Ltd for providing information data on production and sale planning. I also would like to thank Dr. Geoffrey Pritchard for his assistance and advice in statistical analysis on the data.

References

- [1] M.V.F. Pereira, L.M.V.G. Pinto, Multi-Stage Stochastic Optimization Applied to Energy Planning, *Mathematical Programming* 52, 359-375, 1991.
- [2] Z. L. Chen, W. B. Powell, Convergent Cutting Plane and Partial-Sampling Algorithm for Multistage Stochastic Linear Programs with Recourse, *Journal of Optimization Theory and Applications*, Vol. 102, 497-524, 1999.
- [3] C. J. Donohue, Stochastic Network Programming and the Dynamic Vehicle Allocation Problem, *Ph.D. Dissertation, University of Michigan*, 1996.
- [4] M. Hindsberger and A.B. Philpott, ReSa: A Method for Solving Multi-stage Stochastic Linear Programs, working paper, presented at the conference "Stochastic Programming '01", Berlin, Germany, August 2001.
- [5] K. Linowsky and A.B. Philpott, On the Convergence of Sampling Based Decomposition Algorithms for Multistage Stochastic Programs, *Journal of optimization theory and applications*, Vol. 125, No.2, 349-366, New York, 2005.
- [6] S.J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, 1997.
- [7] Fonterra Co-operative Group Ltd, personal communication.

How to set optimal line tariffs

Golbon Zakeri
University of Auckland
g.zakeri@auckland.ac.nz

Abstract

We will discuss the problem faced by lines companies of setting optimal values for their line tariffs. We model this problem as a Stackelberg game in which the lines company is a leader who sets the tariffs. One or more major consumers are modelled as followers. The consumers observe tariffs and respond to them (as well as to other consumer demand patterns).

Farthest Insertion Heuristics for the Freeze-Tag Problem

Sarah E. Marshall

School of Mathematics, Statistics and Computer Science
Victoria University of Wellington, New Zealand
Sarah.Marshall@mcs.vuw.ac.nz

Abstract

The Freeze-Tag Problem is a combinatorial optimisation problem that begins with a set of robots situated across a Euclidean plane. All robots except one (the *root*) are asleep. A robot is awakened when it is reached by another robot. Once awakened, a robot starts moving and can wake up other robots. The objective is to determine the optimal schedule for awakening all robots, i.e., the schedule that minimizes the time until the final robot is awakened (the *makespan*). The solution structure is a degree-constrained tree (the *wakeup tree*).

We investigate the use of farthest insertion heuristics in finding approximate solutions to this problem. Existing greedy-based heuristics tend to construct a wakeup tree from the “inside outwards”, i.e. from the root to the furthest leaves. Farthest insertion tends to construct a wakeup tree from the “outside inwards”. Computational results suggest that heuristics based on farthest insertion are not competitive with existing greedy-based heuristics, but that heuristics based on best insertion perform well.

1 Introduction

The Freeze-Tag Problem (FTP) is a combinatorial optimisation problem that begins with a set V of n robots situated at various locations across a Euclidean plane. Initially all robots, with the exception of the *root* robot, are asleep. A robot is awakened when it is reached by another robot and, once awakened, that robot starts moving and can wake up other robots. The aim of the FTP is to determine the optimal schedule for awakening all robots, such that the time until the final robot is awakened (the *makespan*) is as short as possible (Arkin et al. 2006). This problem was dubbed the Freeze-Tag Problem by Arkin et al. (2006) due to similarities to the children’s tag game.

The FTP begins with the root robot r moving to the position of some other robot i . When robot r reaches robot i , robot i is woken, and from the position of robot i both robots r and i can (but do not have to) move towards other sleeping robots. This process of waking robots continues until the final robot has been woken.

A solution to the FTP can be modelled by a tree structure, within which each robot can be represented by a node. If node i “wakes” another node j (i.e. there is a directed edge from i to j), then node i is said to be the *parent* of node j and node j is said to be the *child* of node i . In this model, the root node r must have one child (i.e. out-degree 1), and all other nodes $j \in V - \{r\}$ must have exactly one parent and can have at most two children (i.e. in-degree 1 and out-degree ≤ 2). The distance d_{ij} denotes the distance between nodes i and j . It is assumed that these distances are symmetric, i.e., $d_{ij} = d_{ji}$, and that for two nodes i and j , $j \neq i$ that $d_{ij} < \infty$. This implies that the set of nodes V is part of a complete weighted graph (Arkin et al. 2006).

Under this tree structure, obtaining a solution to the FTP requires finding a rooted binary spanning tree in a complete weighted graph, with the added restriction that the root node must have out-degree 1 (Arkin et al. 2006). The resulting tree determines the awakening schedule and is therefore called a *wakeup tree*. The objective of the FTP is to find a wakeup tree, such that the *makespan* (i.e. the length of the path from the root node to the furthest leaf) is as short as possible (Bucatuschi 2004; Hoffman 2004).

The FTP was determined to be NP-hard by Arkin et al. (2006). Due to the complexity of the FTP, previous research has focused on developing heuristics to provide good solutions rather than developing algorithms to provide optimal solutions. Several construction heuristics have been suggested for solving the FTP, including greedy, sector (Arkin et al. 2006; Sztainberg et al. 2004) and cheapest insertion (Johnston and Upton 2006). Improvement heuristics include ant colony (Bucatuschi 2004), genetic algorithms (Hoffman 2004) and local search (Johnston and Upton 2006).

The remainder of this paper is structured as follows. Section 2 looks at greedy construction heuristics and visually identifies some shortcomings. As a potential remedy, Section 3 proposes a number of new construction heuristics based on the idea of farthest-insertion. Section 4 conducts a computational experiment to compare the strategies proposed. Finally Section 5 offers some conclusions and recommendations for future research.

2 Greedy Construction Heuristics

The greedy-based heuristics involve the awakened robots moving to the nearest sleeping robot in order to wake it up. Two types of greedy heuristics are described below and their performance is discussed.

2.1 Greedy Claim and Greedy Delay Heuristics

The Greedy Claim heuristic constructs a wakeup tree by appending nodes which are close to the root and gradually moving outwards, i.e., it grows from the inside-out. At each iteration of the Greedy Claim heuristic the nearest asleep neighbour j of node i (the non-full node with the earliest arrival time) is appended, with ties being broken arbitrarily (Arkin et al. 2006).

The Greedy Delay heuristic also grows the wakeup tree from the inside-out. However this heuristic appends nodes using the *effect on the makespan* as the append criterion. At each iteration of the Greedy Delay heuristic, all non-full nodes in the

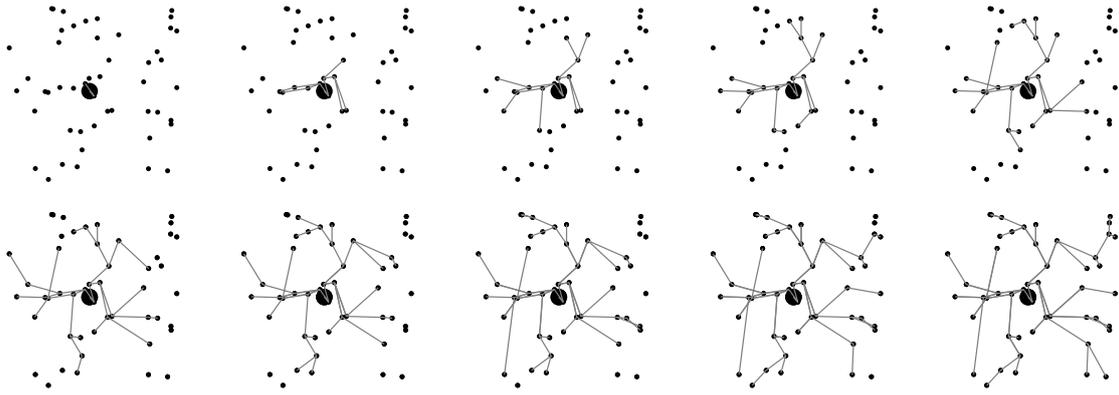


Figure 1: Snapshots of the inside-out growth of the Greedy Delay heuristic for problem instance \mathcal{S} with $n = 50$ robots.

partial wakeup tree (potential parents) are identified. For each potential parent i , the nearest asleep neighbour j (potential child) is identified. For each parent-child pair (i, j) , the distance d_{r+ij} from the root via the parent i to the child j is calculated. The parent-child pair which has the minimum distance d_{r+ij} is chosen, and node j is appended to node i (Arkin et al. 2006). Sztainberg et al. (2004) and Johnston and Upton (2006) both agree that the Greedy Delay heuristic is a consistently good construction heuristic.

Example. The Greedy heuristics construct a wakeup tree from the inside-out, i.e., they start from the root, gradually moving outwards, reaching the furthest nodes last. This growth process is shown for the Greedy Delay heuristic on problem instance \mathcal{S} , with $n = 50$ robots, in Figure 1. The root node has been placed in the centre of the Euclidean plane so the growth can be shown more clearly. As shown in Figure 1, the critical path is likely to be “completed” near the end of the heuristic.

2.2 Sensitivity to the Location of the Root Node

The greedy heuristics construct wakeup trees from the inside-out. They begin by waking the node that is closest to the root. Because of this, the location of the root in relation to the other nodes is likely to have a significant impact on the makespan of the wakeup tree.

In order to test this, the Greedy Delay heuristic was run multiple times for problem instance \mathcal{S} with $n = 50$ nodes, with only the root node having a different location in each instance. Following Jones (1996, 1997), an image was created using the value of the makespan for each location of the root node. This image is shown in Figure 2 and depicts the relationship between the location of the root node and the makespan. The location of nodes $i = 2, 3, \dots, n$ are indicated by white crosses. The shade at a particular point (x, y) indicates the value of the makespan if the root node had been located at (x, y) .

The Greedy Delay heuristic causes the root node to wake its nearest neighbour first; consequently this image is a Voronoi diagram (O’Rourke 1994). Notice that there are boundaries between very dark and very light regions, indicating high sensitivity to the root location. This image shows that the near-root decisions of who

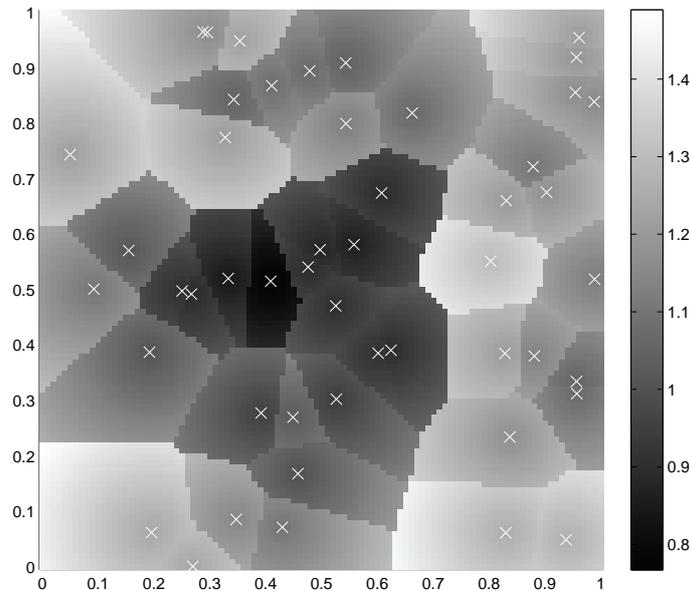


Figure 2: Sensitivity of the Greedy Delay Heuristic to the location of the root node for problem instance \mathcal{S} with $n = 50$ robots.

to wake are *vital* and should not necessarily be driven by the node nearest to the root.

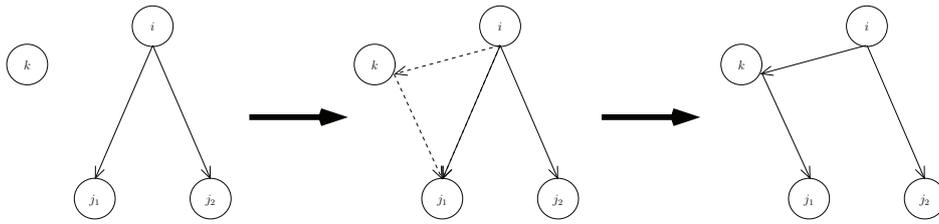
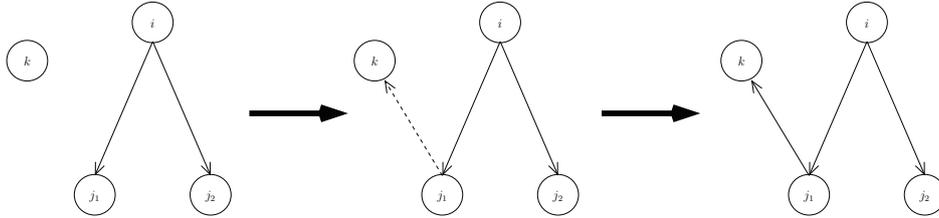
2.3 Research Aim

Since the greedy-based heuristics begin by waking the node which is closest to the root, initially nodes are appended at a relatively low cost. This means that the potentially *troublesome* outer nodes, which are furthest from the root, are left until last and hence the makespan tends to increase significantly in the final stages of the heuristics.

An alternative to growing a tree from the inside-out, would be to grow a tree from the outside-in. If a wakeup tree was constructed by including the potentially troublesome outer nodes first, then we might be able to avoid the significant increase of the makespan in the final stages of the heuristics. Growing from the outside-in may also help to prevent the development of the tight core of edges around the root, which, as shown in Figure 1, can significantly affect the makespan. New heuristics could be developed in order to force the wakeup tree to grow from the outside-in. These heuristics could apply a farthest insertion framework which has proved successful for the Travelling Salesman Problem (Rosenkrantz, Stearns, and Lewis 1977). Hence the primary aim of this paper is to investigate the performance of new insertion-based heuristics compared with the existing greedy-based heuristics.

3 Farthest Insertion Construction Heuristics

To determine whether a farthest insertion heuristic could be competitive against the existing Greedy Delay and Greedy Claim heuristics, a variety of heuristics need to be developed. In the development of heuristics, three main components are considered as follows.

Figure 3: Insertion of node k into a partial wakeup tree.Figure 4: Appending of node k into a partial wakeup tree.

3.1 Initialisation

To form an initial wakeup tree, choose the two nodes i and j which are farthest from the root r . The final wakeup tree must certainly contain nodes i and j , so we seek to find a partial wakeup tree containing at least these three nodes, which has the shortest possible makespan. To accomplish this, select an intermediate node k such that the partial wakeup tree in which k is the child of the root r and the parent of i and of j , has the shortest possible makespan. That is,

$$k^* = \arg_k \min[\max[d_{rk} + d_{ki}, d_{rk} + d_{kj}]].$$

3.2 Operators

A partial wakeup tree can be augmented using a variety of different operations, for example, *unassigned* nodes (nodes not yet in the wake-up tree) could be *appended* or *inserted*.

Insert. As shown in Figure 3, node k could be inserted into the wakeup tree between *parent* i and *child* j_1 .

Append. As shown in Figure 4, node k could be appended to the wakeup tree from a *non-full* node j_1 .

3.3 Overall Strategies

At each iteration of a heuristic, a node can either be inserted or appended. However a *strategy* is required to determine which operation should be performed and on which nodes. Three possible strategies are:

1. **Best Insertion/Append.** Find the best insertion candidate. Find the best append candidate. Of these, choose the candidate and operation which will create the wakeup tree with the shortest makespan.

Table 1: Summary of Heuristics

Heuristic	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Strategy	Greedy	Greedy	1	1	1	1	2	2	2	2	3	3	3	3
Insert Criterion	-	-	1	1	2	2	1	1	2	2	1	1	2	2
Append Criterion	Claim	Delay	1	2	1	2	1	2	1	2	1	2	1	2

2. **Farthest Insertion/Append.** Find the best way to insert each unassigned node k . Find the best way to append each unassigned node k . Of these, find the k^* which produces the longest makespan. For the chosen k^* choose the operation which will create the wakeup tree with the shortest makespan.
3. **Alternating Insertion/Append.** Insert the best insertion candidate. Append the best append candidate.

It remains to define what is meant by “best” operation. The *best* operation can be defined using one of the following criteria.

Criterion 1: The operation which has the least operation cost, with ties broken using the least effect on the makespan.

Criterion 2: The operation which has the least effect on the makespan, with ties broken using the least operation cost.

Thus there are two ways to define *best insert* and two ways to define *best append*.

3.4 Summary of Heuristics

For each of the three strategies outlined above, there are the two ways to define *best insert* and two ways to define *best append*. Combining all strategies with each type of *best* operation yields 12 methods. Including the Greedy Claim and Greedy Delay, 14 heuristics have been tested here. For convenience a numbering system is used (see Table 1), e.g., Heuristic 4 uses strategy 1, and defines *best insert* using criterion 1 and *best append* using criterion 2.

4 Computational Experiments

To test the 14 heuristics outlined above, a set of 1000 problem instances was developed. Each problem instance had 50 robots which were situated at random locations within the unit square. Every heuristic was tested for all problem instances and the makespan in each case was recorded.

The aim was to investigate the performance of the 12 new heuristics compared with the existing greedy strategies. The proportion of problem instances for which a particular heuristic had a makespan which was equal to the shortest makespan, or less than or equal to the third shortest makespan, was calculated (see Table 2). The performance of the heuristics can also be compared by the boxplots in Figure 5.

Table 2: Results from 1000 problem instances with $n = 50$ robots.

Heuristic	Mean Make-span	Proportion Shortest	Proportion Shortest 3
1	1.613	0.000	0.020
2	1.195	0.434	0.884
3	1.662	0.013	0.049
4	1.216	0.343	0.891
5	1.442	0.022	0.145
6	1.215	0.388	0.895
7	2.474	0.000	0.000
8	1.875	0.000	0.004
9	2.324	0.000	0.000
10	1.840	0.000	0.005
11	2.220	0.000	0.000
12	1.915	0.000	0.002
13	1.757	0.000	0.009
14	1.431	0.015	0.148

Effect of the Operation Criteria. All even-numbered heuristics append using the *least effect on the makespan* as the criterion. For a particular heuristic i , $i = 2, 4, \dots, 14$, the heuristic $i - 1$ uses the same strategy and the same insert criterion as heuristic i , i.e., heuristics i and $i - 1$ differ only on the append criterion. As shown in Figure 5, the box plots for even-numbered heuristics are all better than the odd-numbered heuristic immediately before them.

Similar comparisons can be made for the insertion criterion. The heuristic pairs (3,5), (4,6), (7,9) and (8,10) correspond to the first two strategies. The heuristics within each pair differ only on the insertion criterion used. As shown in Figure 5, the boxplots corresponding to these pairs show a similar distribution.

For strategies 1 and 2, it seems that the append criterion used has a greater affect on the makespan than does the insertion criterion. This trend does not seem to hold for the heuristics which used strategy 3.

Effect of the Strategy. Heuristics 1 and 2 correspond to the existing greedy-based strategies, while heuristics 3–6, 7–10 and 11–14 correspond to strategies 1, 2 and 3 respectively. In Figure 5 we notice roughly four clusters of heuristics based on effectiveness. The first cluster consists of heuristics 2, 4 and 6, the second consists of heuristics 1, 5 and 14, the third of heuristics 3, 8, 10, 12, 13 and the final cluster consists of heuristics 7, 9, 11. Overall the two greedy heuristics and those based on strategy 1 seemed to provide short makespans most often.

Behaviour of the heuristics. The results provide an overall impression of the performance of the heuristics but do not provide much information about why particular heuristics are more effective than others. The growth of all heuristics was examined using growth plots similar to Figure 1. A sample of these plots are shown here for heuristics 6, 10 and 14 in Figures 6, 7 and 8 respectively.

Strategy 1 (heuristics 3–6) performed the best append or insertion at each iteration. Heuristics 3 and 5, which used minimum cost as the append criterion, do not

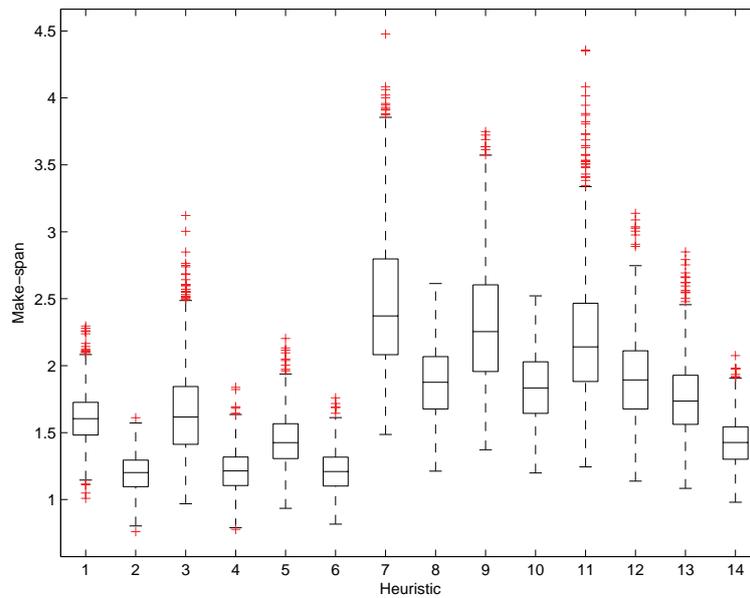


Figure 5: Boxplot of makespan from 1000 problem instances with $n = 50$ robots.

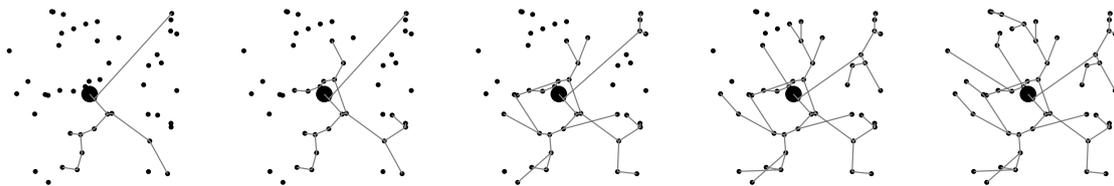


Figure 6: Snapshots of the growth of Heuristic 6 for problem instance \mathcal{S} with $n = 50$ robots.

branch as much as Heuristics 4 and 6, which used the affect on the makespan as the append criterion.

Heuristics (7–10) that use Strategy 2 all developed a *boxy* frame, as shown in Figure 7. This frame collapses inwards during the latter stages of construction. None of these heuristics showed much branching.

Strategy 3 (heuristics 11–14) forced the wakeup tree to insert and append on alternate iterations. Despite the compulsory appending at every alternate iteration, heuristics 11 and 12 did not perform much branching. This feature was less evident for heuristics 13 and 14, which produced bushier wakeup trees.

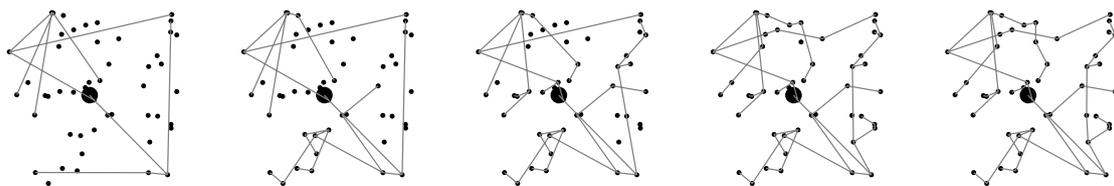


Figure 7: Snapshots of the growth of Heuristic 10 for problem instance \mathcal{S} with $n = 50$ robots.

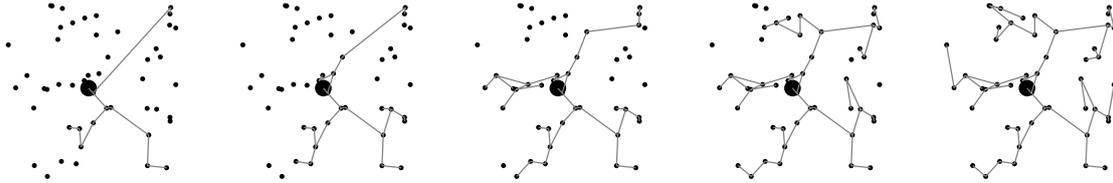


Figure 8: Snapshots of the growth of Heuristic 14 for problem instance \mathcal{S} with $n = 50$ robots.

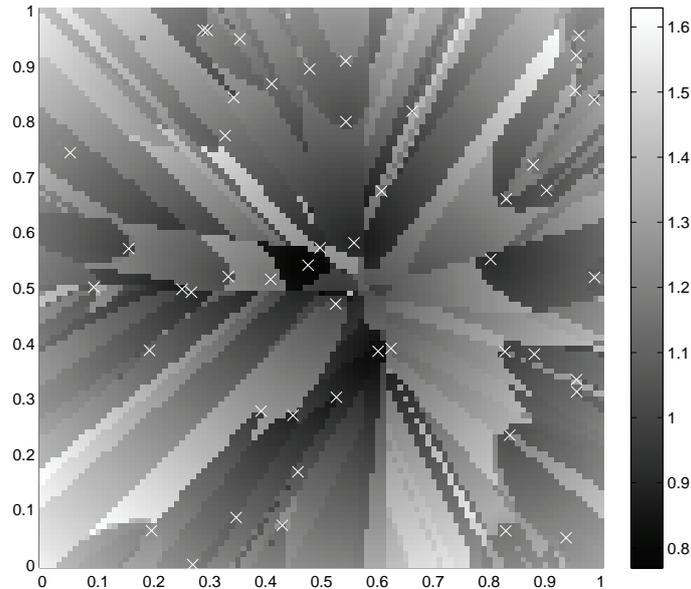


Figure 9: Sensitivity of Heuristic 6 to the location of the root node for problem instance \mathcal{S} with $n = 50$ robots.

Sensitivity to the Location of the Root Node. Figure 9 gives the sensitivity of the makespan to the location of the root node using Heuristic 6 (compare with Figure 2). A *radial* structure is apparent in Figure 9, with shading indicating that the initial movement of the root robot is to wakeup a central robot rather than one necessarily nearby (compare to the Voronoi structure in Figure 2). Notice that there are still boundaries between very dark and very light regions, so Heuristic 6 is no less sensitive to the root location than Greedy Delay.

5 Conclusion

This research has shown that the farthest insertion heuristics tested here (i.e. heuristics 7–10) are not competitive with the existing greedy heuristics for finding solutions to the FTP. However it has also highlighted some interesting features of insertion heuristics when applied to the FTP.

Overall, strategy 1 seemed to provide the most efficient makespans most often. Strategies 2 and 3 were less effective, but could however be refined (perhaps by encouraging more branching) to produce better solutions. A refinement of the logic used here in creating new strategies could be a potential area for future research. A comparison of the 14 different heuristics suggests that the append criterion used

seems to have a greater affect on the makespan than the insertion criterion.

Variations of the FTP are also possible and as yet have not been studied, so would also provide an area for future research. For example, suppose that the waking of each robot generated a particular reward (perhaps time-dependent rewards). The addition of rewards means that the objective of the problem will be to maximise the rewards *and* minimise the makespan. Another variation of the FTP would be if there is a cost associated with total distance travel (perhaps due to fuel usage). In this case, the objective of the problem will be to minimise the makespan, and the total distance travelled.

Acknowledgements

I would like to thank my supervisor Mark Johnston for his support and encouragement throughout this research project.

References

- Arkin, E.M., M.A. Bender, S.P. Fekete, J.S. B. Mitchell, and M. Skutella. 2006. "The Freeze-Tag Problem: How to Wake Up a Swarm of Robots." *Algorithmica* 46:193–221. A preliminary version appears in *Proceedings of the 13th annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 568–577, 2002.
- Bucatanachi, D.G. 2004. "Ant colony system for the freeze tag problem." *Midstates Conference for Undergraduate Research in Computer Science and Mathematics, Denison University*.
- Hoffman, B. 2004. "Genetic algorithms applied to the Freeze Tag Problem." *Midstates Conference for Undergraduate Research in Computer Science and Mathematics, Denison University*.
- Johnston, M.R., and G.J.G. Upton. 2006. "Improvement heuristics for the Freeze-Tag Problem." Submitted to *Journal of Heuristics*.
- Jones, C.V. 1996. *Visualization and Optimization*. Boston: Kluwer.
- . 1997. "The Stability of Solutions to the Euclidean Traveling Salesman Problem. Part I: Experimental Results." Unpublished working paper, University of Washington.
- O'Rourke, J. 1994. *Computational Geometry in C*. Cambridge: Cambridge University Press.
- Rosenkrantz, D.J., R.E. Stearns, and P.M. Lewis. 1977. "An Analysis of Several Heuristics for the traveling salesman problem." *SIAM Journal on Computing* 6:562–581.
- Sztainberg, M.O., E.M. Arkin, M.A. Bender, and J.S.B. Mitchell. 2004. "Theoretical and Experimental Analysis of Heuristics for the "Freeze-Tag" Robot Awakening Problem." *IEEE Transactions on Robotics and Automation* 20:691–701. A preliminary version appears in *Proceedings of the 8th Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 270–279, 2002.

Routing Trains Through Railway Junctions: A New Set Packing Approach

R. Lusby

Department of Engineering Science
University of Auckland
r.lusby@auckland.ac.nz

Abstract

Arguably the most important decisions facing railway companies today concern the effective utilization of available resources. One such problem, the motivation for this paper, entails allocating the track capacity of a junction to a timetabled set of trains so as to ensure quality routings are obtained. Large junctions are highly interconnected networks of track where multiple railway lines meet, intersect and split. The huge number of routings possible makes this a very complicated problem. We show how this problem can be logically formulated as a set packing model and solved efficiently with a pricing routine in which the columns of the constraint matrix are represented using tree structures. Such a structure is effective as dual variable values can be accumulated at nodes in the tree, and hence the individual pricing of variables can be avoided. A discussion of the variable generation phase (train paths), which takes into account the dynamics of the trains, is also included. The decision support system currently being developed will enable planners to solve strategic, tactical, and operational variants of the problem. An example junction is used to illustrate the proposed methodology.

1 Introduction

The railway industry is rich in problems that can be modelled and solved using Operations Research techniques. In this day and age, arguably the most important of these are the ones that concern the effective allocation and utilization of available resources. One such problem, the focus of this paper, entails allocating the track capacity of a junction to a timetabled set of trains to ensure quality routings are obtained whilst adhering to a variety of operational constraints.

The diverse range of problems facing railway companies are typically categorized according to the required planning horizon. This leads to the customary three stage approach with problems being defined as strategic, tactical, or operational in nature. The problem of routing trains through railway junctions arises at each of the levels. A short description of each variant is given next.

Problems at the strategic level are characterized by lengthy time horizons and typically involve resource acquisition. Viewed in this context the problem considered

here appears in the form of capacity assessment. Railway management often face the task of deciding between a number of possible investment alternatives concerning proposed infrastructure modifications to junctions. Perhaps the most influential factor in making the final decision is capacity. Railway management are very interested in knowing, with precision, what level of rail traffic the modified infrastructure would cater for. This involves determining the maximum number of trains that could be routed through the junction within a given time horizon. The construction or modification of infrastructure involves high capital investment and has long lasting ramifications. It is essential that one has proper tools to assist in this process.

Tactical level problems focus more on allocating resources on an infrastructure which is assumed to be fixed. Such problems normally have a mid-term planning horizon. On this level the problem considered here answers the question of timetable feasibility. In most countries it is not uncommon for the railway system to be divided into two main areas; those responsible for the infrastructure, and those responsible for the rolling stock (train operating companies). The train operating companies each submit a preferred timetable, and the infrastructure managers determine if there exists a conflict free routing for all the trains in the amalgamated timetable. There is no clear objective at this stage, however, one typically aims to schedule the maximum number of trains while considering the route preferences of the trains.

Operational problems are defined to be those that occur on a day-to-day operational basis when pre-determined operating policies need to be adjusted due to unforeseen disturbances. The dynamic environment in which these problems occur necessitates real time resolution. The impact of late train arrival, track maintenance, or even accidents will propagate through the timetable with varying degrees of severity and quite possibly result in the pre-determined operating policy becoming infeasible. The variant of the problem occurring here entails reassigning train routes so as to return to the original schedule with minimal required disruption.

Despite its apparent potential benefits, the application of Operations Research techniques to the problem of routing trains through railway junctions has been surprisingly limited with manual approaches still widely employed. The purpose of this paper is to present a set packing model applicable to all instances of the problem outlined above. We demonstrate how the train routing problem can be logically formulated as a set packing model, and discuss its flexibility from a modelling perspective. While being perhaps the most logical formulation of the problem, its dimensions do impose some restrictions from a computational point of view. However, we show that the dual of this formulation does possess a number of nice properties that one can exploit in solving the problem, and present a solution procedure which entails solving the dual through the dynamic addition of violated cuts (primal variables). The primal variables are train paths and are constructed using a generator which takes into account the dynamics of the trains. An efficient pricing routine in which the primal variables are represented by several tree structures is also described. We illustrate the proposed methodology with the aid of an example junction.

This paper is organized as follows. Section two gives a detailed description of the problem, introduces some definitions and previous work. Section three describes the set packing model and its dual as well as our path generator, while section four details the solution approach and also the primal variable tree structures. We conclude with an example in section five, and conclusions are given in section six.

2 Problem Definition

A *junction* is a highly interconnected network of track where multiple railway lines meet, intersect, and split. Quite typically it includes a station, although this is not assumed in the definition. The network of track comprising the junction can be divided into a number of *track sections*. These are essentially segments of track on which for safety reasons there can be at most one train at any given time. Some examples include switches, crossings and platform track sections. For completeness, and for the purpose of remaining consistent with the previous literature, we define the perimeter of the junction to consist of a number of *entering points* and *leaving points*. These are rather self-explanatory and just indicate the points at which it is possible for trains to enter and leave the junction. Furthermore, they also identify the scope of the infrastructure concerned. A *route* through the junction is thus a sequence of track sections connecting an entering point to a leaving point. The route may or may not involve stopping at an available platform. Depending on the number of such points as well as the number of switches in the junction, there may be a significant number of routes possible. We distinguish this from a *train path* which refers to a possible traversal of a given route in time. Figure 1 below illustrates the typical characteristics of a junction as well as a possible route from entering point A to leaving point E.

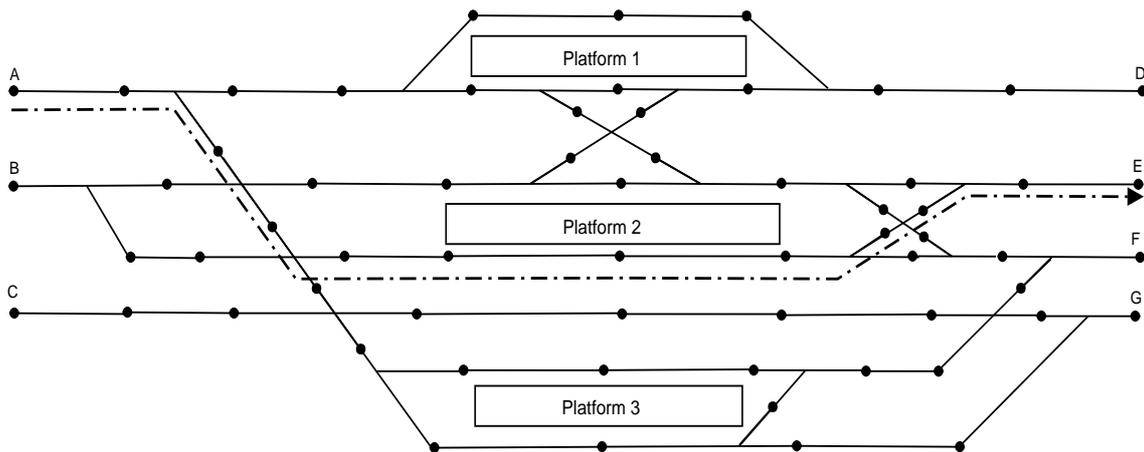


Figure 1: A possible junction showing a possible route

A formal definition of the problem we are addressing is as follows. Given the detailed track layout of the junction (defined by the entering and leaving points) as well as a proposed timetable, i.e. the respective arrival and departure times for a set of trains, what is the maximum number of trains that can be assigned a route through the junction? This is what is referred to as the feasibility problem. Other objectives are of course possible, however this one has been chosen for expository purposes. This simple objective function does illustrate the subtle difference between the strategic and tactical level variants of the problem. If the solution to this problem is less than the cardinality of the train set, the proposed timetable is infeasible, and a saturated solution (for this timetable only) is given by the solution to the optimization problem. On the other hand, if the solution to this problem is equal to the cardinality of the train set (i.e. a conflict free routing has been found for all trains), the proposed timetable may or may not be saturated. One could look at introducing saturating trains in such a situation. Irrespective of which problem variant we are concerned

with, the chosen routes must satisfy a number of constraints.

The most important requirement for the selected routes is that no two trains share any part of the junction infrastructure at the same time. Routes that do not satisfy this criterion are said to be in conflict, and obviously cannot be assigned simultaneously. To prevent trains from getting too close to one another within a junction railway companies tend to implement one of two (or possibly both) systems. The *route locking and sectional release system* enforced by the Dutch (among others) stipulates that trains must lock a sequence of track sections prior to using them. For instance, when a train arrives at an entering point to the junction it must lock all the track sections it is going to use in reaching its designated platform, and then prior to departure, lock all the track sections it is going to use in leaving the junction. Trains successively release each of the locked track sections after traversing them. Such a system allows trains to proceed without interruption within a junction as no track section can be simultaneously locked by two or more trains. Some buffer time is usually incorporated into the release time of the individual track sections to build some robustness into the route. The *block signalling* system implemented by the French and German railway companies is similar. Essentially the railway network is divided into a number of blocks delimited by signals, where each block may contain one or more track sections. On entry to a block, all track sections are simultaneously locked. These are released when the tail of the train has exited the block and some additional clearing time has elapsed. Any realistic model must accurately incorporate these features.

Zwaneveld et al. (1996) detail a number of other constraints pertaining to customer considerations as well as train connections. For instance, it is often beneficial to have all trains travelling in a similar direction leave from platforms that are close to each other in proximity. It is also necessary to account for trains that must be coupled and decoupled. There may of course be other constraints not listed here that refer to particular operating policies implemented by individual railway companies.

The problem of routing trains through railway junctions has received relatively limited attention in the literature. Zwaneveld et al. (1996) and Zwaneveld, Kroon, and van Hoesel (2001) propose node packing formulations and solve it exactly. Delorme (2003), Delorme, Gandibleux, and Rodriguez (2004), and Gandibleux et al. (2005) also adopt node packing formulations but elect to solve them via metaheuristics. Delorme, Rodriguez, and Gandibleux (2001), and Rodriguez (2002) present constraint programming approaches, while Carey and Carville (2003) propose a simple greedy heuristic.

3 The Set Packing Model and its Dual

The problem of routing trains through railway junctions is a natural application of the set packing model. Recall that to provide a conflict free routing through the junction for a set of trains, one must ensure that at most one train is locking any track section at any given time. To capture both the spatial and temporal components inherent in such a restriction, one needs to identify a constraint for each track section in a sequence of uniform time intervals. This is analogous to taking a snapshot of the junction in each time interval. The time interval used is obtained through a discretization of the timetable period. This is consistent with what is done in planning in practice although the duration of the time interval is railway

company dependent. A typical value might be about 15 seconds. This modelling approach allows one to represent a path through the junction for a particular train as a column of zeros and ones. A one in a particular row indicates that the train is claiming that particular track section at that particular time, while a zero indicates otherwise. As was mentioned earlier, trains may have a number of possible routes (each with many paths) through the junction, including of course the *null route* which pertains to the train not being routed at all. We therefore, in addition to the set packing constraints discussed above, generalise the model through the inclusion of a generalized upper bound (GUB) constraint for each train, thus ensuring we pick one and only one of the possible paths for each train. Our complete model is given below. We will refer to this model as our primal problem.

$$\begin{aligned}
 & \text{Maximize} && \sum_{i=1}^t \sum_{j=1}^{n_i} \rho_{ij} x_{ij} \\
 & \text{subject to} && \begin{bmatrix} T_1 & T_2 & \dots & T_t \\ R_1 & R_2 & \dots & R_t \end{bmatrix} \mathbf{x} \leq \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \\
 & && x_{ij} \in \{0, 1\} \quad i = 1, 2, \dots, t, \forall j \in P_i,
 \end{aligned} \tag{1}$$

where t is the number of trains, b_1 and b_2 are vectors of ones, P_i is set of paths for train i (indexed 1 to n_i), ρ_{ij} is the benefit received in assigning train i path $j \in P_i$. π_1 and π_2 are the dual vectors corresponding to the GUB constraints, and the time period track section constraints respectively. The constraint matrix consists of submatrices T_i and R_i ($i = 1, 2, \dots, t$). These are defined as follows.

$$T_i = e_i e^T \text{ with } e_i \text{ the } i\text{th unit vector and } e^T = [1 \ 1 \ \dots \ 1]$$

$$R_i = (r_{l,k}) = \begin{cases} 1 & \text{if the } k\text{th path for train } i \text{ uses time interval track section pair } l \\ 0 & \text{otherwise.} \end{cases}$$

Our decision variables are given by the binary variables

$$x_{ij} = \begin{cases} 1 & \text{if train } i \text{ is assigned path } j \in P_i \\ 0 & \text{otherwise.} \end{cases}$$

Unlike the node packing models proposed by Zwaneveld et al. (1996) and Zwaneveld, Kroon, and van Hoesel (2001), this particular approach implicitly deals with the train routes as it attempts to identify and resolve conflicts between trains during the solution procedure rather than having to generate them all a priori and explicitly represent the infeasibilities. This being the case, we can easily consider additional trains, additional routes, and delayed trains. Identifying and resolving conflicts is embedded in the optimization and there would be a limited amount of additional effort required to include such variables. Providing that they represent feasible paths through the junction, they are nothing more than extra columns in the model.

The above model is very similar in structure to a well known optimization model known as the crew rostering model developed by Ryan (1992). Indeed, one could regard a train as being a crew member and the sequence of time period track section pairs it claims as its sequence of duties. Thus the columns defining routes for trains

are analogous to the columns defining lines of work for crew members. From a modelling perspective the only noticeable difference between the two models appears in the right hand side. In the crew rostering model this is not restricted to being of unit value, nor packing for that matter. In terms of the dimensions of the two models there is a major difference. The crew rostering model is characterized by a small row dimension with many variables. It is not uncommon for crew members to have hundreds of thousands of possible lines of work, however there are only relatively few constraints corresponding to the duties that must be allocated. The set packing model above, on the other hand, is characterized by a constraint matrix with many constraints and relatively few variables. Any realistic problem will have thousands of time period track section constraints. In contrast, trains only have a limited number of possible routes through the junction.

To determine the best solution approach to adopt, one should consider what properties an optimal solution to the primal problem (LP) would have. We would expect there to be, comparatively speaking, few variables. We also believe that there would be a significant number of inactive constraints. It seems highly unlikely that all track sections will be locked in all time intervals. The dimensions of the primal problem naturally suggest one should consider the dual formulation. In the dual setting at optimality these observations would be mirrored in the form of very few active constraints, and most of the dual variables being non-basic at value zero. The dual formulation has a similar dimension to the crew rostering model. Experience tells us that such models can be solved efficiently. The solution approach that we have adopted, which is explained in detail in Section 4, hence focuses on the dual and attempts to exploit its small row dimension as much as possible. For completeness, the dual formulation of (1) is given below (all notation is as previously defined).

$$\begin{aligned} \text{Minimize} \quad & b_1^T \pi_1 + b_2^T \pi_2 \\ \text{subject to} \quad & A^T \begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix} \geq \rho \\ & \pi_2 \geq 0 \end{aligned} \tag{2}$$

4 Solution Approach

The solution approach we propose consists of two main steps. The first step involves finding all possible train paths (variable generation), while the second entails solving the problem. A detailed description of each is given next.

4.1 Variable Generation

In our modelling we assume that all train routes are generated a priori, but not train paths. Modelling train paths essentially involves determining the distribution of zeros and ones in a particular column. In other words, indicating when and at what time a train will lock particular track sections during its passage through the junction. This depends on two key factors. The safety system enforced by the rail company stipulates how many additional track sections will need to be locked, while the dynamics of the trains (the acceleration and deceleration capabilities) will determine the exact running time of a train on a given track section.

In modelling train dynamics we make several assumptions. Whenever a train enters a new track section on its route, it is limited to doing the following:(1) Proceeding with constant velocity,(2) Accelerating at a constant rate, and (3) Decelerating at a constant rate. This approach of constant acceleration and deceleration rates is consistent with previous work on modelling train dynamics; Zwaneveld et al. (1996) and Lu, Dessouky, and Leachman (2004) both adopt a similar approach. We believe it to be a realistic approach in that it generates only practically feasible paths. Extra restrictions are also included to ensure a train will not accelerate immediately after decelerating and vica versa. This ensures only smooth train paths are considered.

We assume that for each of a train's possible paths, the entering track section, entering time, and entering speed of the train is known. We also make the assumption that a train will not be accelerating/decelerating on entry to a junction. This defines the initial point of the train path, and is represented as a time space node which stores this information. Similarly, a time window in which the train may depart the junction, as well as the leaving track section are also known. In addition to this each train is assigned a maximum permitted amount of time it can be in the junction. The first time space node is then extended based on the dynamics of the train given the possibilities outlined above. A new time space node will be generated each time a train enters a new track section on its route. This process continues until all possible train paths have been enumerated. Train paths will terminate as feasible if they leave the junction in the designated time window. However, they will terminate as infeasible if this is not the case, or if the duration of the train run through the station exceeds the maximal tolerance. A diagram illustrating this process is given below.

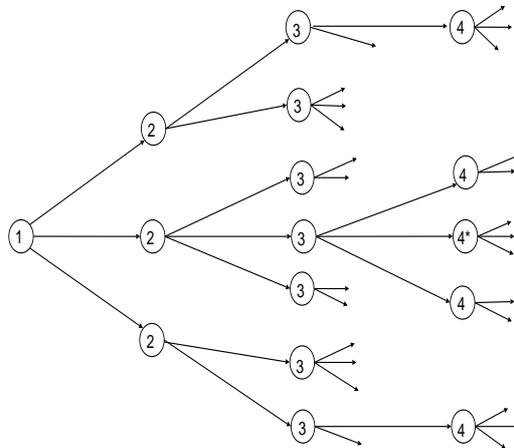


Figure 2: Modelling Train Paths.

The figure above represents the possible ways a train may traverse the first three track sections of its route. The number of the track section it is entering is given on the node. The successor nodes are generated based on the dynamics of the train. The arrows emanating from nodes correspond to the different possible kinematic options available. Each track section is assumed to have both a minimum and maximum permitted velocity. This could be an infrastructure restriction (i.e. speed limit), or the maximum permitted speed of the train. As a result of this, one needs to be careful when extending nodes. To ensure feasible extensions are guaranteed a backwards preprocessing step is used to adjust the maximum and minimum velocities

on a given track section. This prevents a train from reaching a situation in which it will be unable to comply with the speed requirements of the next track section.

As can be seen from the diagram, the first three track sections can be traversed via a number of different acceleration and deceleration patterns. The fact that we consider only constant rates of acceleration and deceleration, and that different track sections often have different lengths, means it is extremely unlikely that duplicate time space nodes will be generated via two different traversal patterns. On the rare occasion that this does occur, removing one of the labels through domination would also be unlikely. This is primarily because each time space node is defined by an entering time, entering track section, entering speed and entering acceleration. All of which would need to be equal for domination to occur. This means that the modelling described above in fact builds a tree structure. For each route for each train there will be a different tree. On traversing any tree one would obtain all possible ways to traverse the corresponding route.

4.2 Primal Variables as Tree Structures

The representation of the train routes (primal variables) as tree structures is advantageous from a computational point of view. It allows one to price the paths described by the tree very efficiently. Using Figure 2 as an example once more, the arcs indicate the length of time the train spends on the track section associated with the node it emanates from. Hence telling us which primal constraints all train paths containing this arc would cover. We can easily store an accumulated dual value at each node by summing the appropriate dual variables along the arc leading to it. Through assigning each node an accumulated dual value one has effectively priced out a component of the reduced cost for all train paths using this arc. For example, all subsequent nodes of 4* would have the accumulated dual variables to that point in common. This is much more effective than pricing individual variables as one only ever needs to back track to the first arc of difference on pricing a different train path; equivalent to a pre-order walk of the tree. If the accumulated dual of a leaf node (which represents a complete path) is favourable, the variable it represents is easily obtained by tracing back through each predecessor node.

4.3 Solution Method

To solve the set packing model above we propose an approach which attempts to exploit the small row dimension of the dual formulation as much as possible. Essentially the method involves dynamically updating the dual problem through the addition of entering variables as violated constraints.

We begin with an initial dual problem which has just one constraint for each train. We elect to assign each train its null route, although one could easily assign each train any of its possible routes. This is done because we only ever need enough constraints (primal variables) in the dual to ensure we have a primal basic feasible solution. This is analogous to having a restricted master problem in the primal setting. The solution to the dual problem gives us the dual variables for the primal set packing LP. This solution vector can easily be used to price out the tree structures for favourable train paths. A primal variable (cut) pool is also set up (initially null) which will store a subset of the paths. This pool is priced prior to the trees.

In our pricing of the tree structures we implement a form of partial pricing. We

only ever call the pricing routine for a particular train and a particular route, and thus initially focus on one tree. If a favourable train path is found in this tree it is returned immediately to the optimization in the form of a violated constraint. The reduced cost of the primal entering variable is the extent to which the corresponding dual constraint is violated. Hence, on appending the constraint to the dual problem, an artificial variable with an initial value equal to the reduced cost is included to ensure we have a starting basis for the next iteration. The dual problem is then reoptimized to obtain updated dual variables, and the process repeats itself. In effect, we equate an iteration of the primal simplex with the addition of a dual constraint. If, on the other hand, a favourable train path is not found in the tree called for, another tree (for a different train and route) is examined. This continues until either a primal entering variable has been found, or we have priced all trees. In the latter case, we would declare optimality with the optimal solution to the primal problem being the dual vector of the optimal solution to the dual problem.

In an effort to maintain a small basis throughout the solve an inactive constraint removal routine is also implemented. If at a particular solution to the dual an inactive constraint appears, it is removed. Inactive constraints in the dual indicate that the corresponding primal variable is at value zero and can be removed.

5 Example Junction

To demonstrate our model and solution approach we test it on the junction given in Figure 4. While this junction is fictional and purely for expository purposes, it has been created based on the Pierrefitte-Gonesse junction which is used in the test cases of Delorme, Rodriguez, and Gandibleux (2001), Delorme, Gandibleux, and Rodriguez (2004), Delorme (2003), Gandibleux et al. (2005), and Rodriguez (2002). The purpose of this was to assimilate a real life situation as much as possible. The test junction has 50 track sections. Each of which has an upper and lower speed limit and is either unidirectional (indicated by an arrow) or bidirectional. The four different directions entering/leaving the junction have been labelled A,B,C, and D.

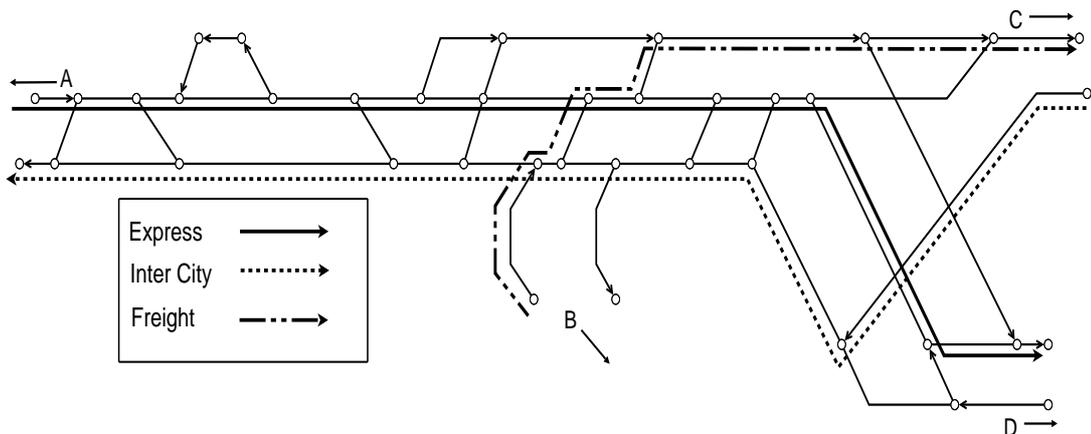


Figure 3: Test Junction.

The instance we consider consists of eight trains. All trains enter the junction within 570 seconds of each other and the timetable period is 1100 seconds. The trains are one of three types; Express, Inter City, or Freight. The composition is as follows:

- 3 Express trains between A and D
- 1 Express train and 2 Inter City Trains between A and C
- 2 Freight trains between B and C

The difference in train category is reflected in the acceleration and deceleration capabilities of each. The values used are consistent with those of Danish trains; again in order to achieve a real life situation. In this particular case trains have between one and three possible routes (an example of each is given in the diagram).

As was explained in Section 4, the first step in the solution process is to generate the necessary tree structures. For this problem 17 such trees are required. The largest of which contains approximately 500,000 nodes representing about 260,000 possible paths, while the smallest consists of 77 nodes representing 36 paths. In total, the 17 tree structures contain approximately 800,000 possible paths. In reality we believe it is unlikely that there would be so many possible ways to traverse the routes. However, by considering trees that are bigger than we would expect, we can gain an insight into the time required to initialise trees in general. For the problem at hand all 17 tree structures are initialized in less than two seconds.

The pricing of the trees is also very quick. To price all the paths in a particular tree involves a pre-order traversal of the tree and one would expect this to be on par with, if not faster than, the time required to generate the tree itself. Indeed this is the case. Small trees can be priced in fractions of a second, while the larger ones may take slightly longer. The largest tree in the problem considered here takes no more than half a second to price. The traversal code used to price the trees has been stress tested on a tree containing 1.7 million nodes. The 1 million variables that this tree represented took between two and three seconds to price. We believe this to be very encouraging as trees are not likely to ever contain this many possible paths.

With our approach the problem took less than five seconds to solve. This included the preprocessing time involved in generating all paths. The following statistics are interesting to note. During the solve 34 constraints (train paths) were added to the dual, 29 were removed, and there were never more than 17 constraints in a dual basis. This illustrates how we can maintain a very small basis throughout the solve.

The fractional solution obtained by the solver indicated that it is possible to achieve an objective of eight (equivalent to routing all trains) by fractionally assigning routes to trains. By recording integer solutions found by the dual during the execution of the solver a lower bound on the number of trains that can be assigned a route can be obtained. In this case, a lower bound of 7 was found. From the optimal solution we can easily identify where a conflict occurs, and which trains are involved. Although no specific branch-and-bound routine is included in the software at this stage, to ascertain whether or not an integer solution routing all trains exists, a form of primal constraint branching was implemented. Conflicts were chosen arbitrarily from the optimal solution and a decision as to which train would use the particular time period track section was made (again arbitrarily, although in reality one of the two trains may have a higher priority than the other). As a result of this certain variables were no longer considered for each of the trains. The problem was then resolved with a smaller set of variables. Two such branches were required to show there did exist an integer solution which routed all trains. This shows that this problem lends itself to the constraint branching framework. This is the topic of future research and will be implemented in later work.

If the objective value is ever less than the number of trains being considered, the proposed timetable would be infeasible and the only way of achieving feasibility would entail changing the timetable, or considering alternative routes for trains.

6 Conclusions

We have shown that the problem of routing trains through railway junctions can be formulated as a set packing problem and solved efficiently via a procedure that dynamically updates the dual problem through the addition of violated cuts. We have also demonstrated that by representing train routes as tree structures we can price a significant number of train paths very efficiently. We believe that both the modelling technique and solution approach described herein to be very promising.

References

- Carey, M., and M. Carville. 2003. "Scheduling and platforming trains at busy complex stations." *Transportation Research, Part A* 37A (3): 195 – 224.
- Delorme, X. 2003. "Modélisation et résolution de problèmes liés à l'exploitation d'infrastructures ferroviaires." Ph.D. diss., Université de Valenciennes et du Hainaut Cambrésis.
- Delorme, X., X. Gandibleux, and J. Rodriguez. 2004. "GRASP for set packing problems." *European Journal of Operational Research* 153 (3): 564 – 580.
- Delorme, X., J. Rodriguez, and X. Gandibleux. 2001. "Heuristics for railway infrastructure saturation." *Electronic Notes in Theoretical Computer Science* 50 (1): 39 – 53.
- Gandibleux, X., J. Jorge, S. Angibaud, X. Delorme, and J. Rodriguez. 2005. "An ant colony optimization inspired algorithm for the set packing problem with application to railway infrastructure." *Proceedings of the Sixth Metaheuristics International Conference (MIC2005)*.
- Lu, Q., M. Dessouky, and R.C. Leachman. 2004. "Modeling Train Movements Through Complex Rail Networks." *ACM Transactions on Modeling and Computer Simulation* 14 (1): 48 – 75.
- Rodriguez, J. 2002, October. Constraint programming for real-time train circulation management in railway nodes. Presentation from the 3rd AMORE Research Seminar.
- Ryan, D.M. 1992. "The solution of massive generalized set partitioning problems in aircrew rostering." *Journal of the Operational Research Society* 43 (5): 459 – 467.
- Zwaneveld, P.J., L.G. Kroon, H.E. Romeijn, M. Salomon, S. Dauzere-Peres, S.P.M. van Hoesel, and H.W. Ambergen. 1996. "Routing Trains Through Railway Stations: Model Formulation and Algorithms." *Transportation Science* 30 (3): 181 – 194.
- Zwaneveld, P.J., L.G. Kroon, and S.P.M. van Hoesel. 2001. "Routing trains through a railway station based on a node packing model." *European Journal of Operational Research* 128:14 – 33.

On the Mean Cumulative Function of Censored Warranty Data

Bronwyn R. Erasmuson

School of Mathematics, Statistics and Computer Science

Victoria University of Wellington

bronwyn.erasmuson@mcs.vuw.ac.nz

Abstract

Analysis of the Mean Cumulative Function (MCF) provides insight into many features of warranty data. Empirical data is used to calculate the MCF, so the method used here is particularly useful when the life distribution of the product is not known. Plotted functions yield information that might not otherwise have been found by purely computational methods. In warranty analysis of recurrence data, studies of the MCF provide information related to the number and cost of recurrences. This is of interest to the manufacturer, so liability can be estimated, and to engineers, as durability of the product is an area of scrutiny.

Here we will investigate the nonparametric approach to recurrent events data, and use a real data set from the automotive industry to derive the mean number and cost of events at any given time, and confidence intervals for these estimates.

1 Recurrent Events Data

Events which occur repetitively throughout the sample unit's life are called *recurrent events*.

Many applications involve repeated events data where a sample unit may accumulate any number of events over time. In the area studied in this report, product reliability, recurrent events are common. They occur in the areas of electronics, appliances, aviation, medical equipment observed, and transportation. We illustrate our findings by using automotive data from the General Motors warranty database. Recurrent events data also occurs in other everyday life, such as medicine, economics, criminology, the social sciences, and in business and marketing.

2 The Nonparametric Approach

2.1 Limits of the simple renewal process

The simple renewal process uses only the count to yield information about the sample, there is no interpretation in relation to cost, or other measurements related

to each recurrence. These recurrences themselves may be neither independent nor identically distributed, making many models currently used invalid. This very limited interpretation of the data provides minimal, and often invalid results. The use of nonparametric methods here overcomes these obstacles.

2.2 Nonparametric Problems

In many real life situations, the study of the uncertainty related to these situations requires the knowledge of certain probability distributions. Sometimes the general form of the distribution is assumed to be known, i.e., the distribution belongs to a certain family of distributions, then the problem reduces to the estimation of the unknown parameters. These are so called parametric models. In the contrary, if the distribution can not be identified as a member of a certain family of distributions, then nonparametric models are more useful.

There are many parametric models available for recurrence data, some of which are mentioned in Nelson (2003). The simplest model is the Poisson process, which is often used for product repairs. All parametric models have some degree of error in them. This can sometimes be negligible, but when it becomes sufficiently significant, it may be preferable to use a nonparametric approach.

2.3 The Nonparametric Approach To Recurrent Events Data

The nonparametric model for a population of units is a population of curves, each representing the cumulative history function for one unit. For discrete events, this is a population of staircase functions. Assume that the number (or cost) of events at any time t for a population has mean denoted by $M(t)$. The nonparametric MCF, or $M^*(t)$, is an estimate of the population $M(t)$ that involves no assumptions about $M(t)$.

The nonparametric method requires only very simple assumptions. The target population must be clearly specified, and a random, unbiased sample taken, with random censoring. The censoring is random if the cumulative history functions, that is the distribution of recurrences (or cost), of all the sample units are independent from their censoring ages. To simplify the theory, ties are assumed to not exist. In this paper, ties are placed in random order, as the MCF estimate depends on their order in calculations. These assumptions involve only the collection of a sample, not aspects of $M(t)$ itself.

The nonparametric MCF, and its confidence intervals, are calculated by looking at the cumulative number of recurrences, or cost, at any given time, in comparison to the number of units at risk. For right censored data, the number of units at risk is the number of uncensored, i.e. "alive" units. Censoring times are used to calculate the number of units at risk, so the procedure holds when right and left censoring, and even gaps in the data, are present.

2.4 Gaps and Censoring

In practical circumstances, a collection of data may have many periods over which part of the data is missing. When the records of a sample unit end at a certain

point, we call this right censoring. Similarly, when they begin at a certain time, so there is no records of the beginning of the products life, we call it left censoring. Gaps in the records also frequently occur, for example records may be lost, or not taken if the sample unit is not available at the usual sampling time.

Figure 1, below shows a sample of right censored data. Each horizontal line represents a vehicles lifespan, the crosses represent claims. Many vehicles in the sample had no claims. These must still be included as they are used in the calculations of number of units at risk.

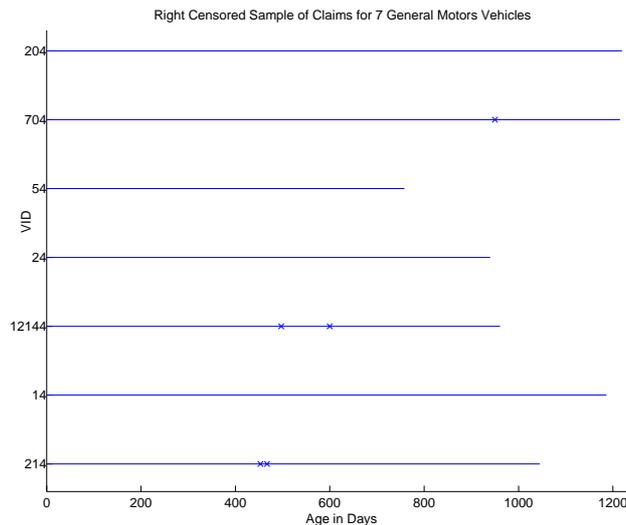


Figure 1: Sample of General Motors Data

The distribution of recurrences after the censoring time is unknown. This needs to be accounted for in the nonparametric model. The MCF and confidence intervals are calculated by taking into account the number of units *at risk* at that time, hence we just increment or decrement this number according to what data we have at different points in time.

2.5 Interval Age Data

In some cases, the exact event ages and censoring times are unknown. This often occurs with large samples of data, as these are collated for simplicity. The age scale is partitioned, and a tally of the number of events and censoring times for each interval is recorded. The data used in this paper is recorded in intervals of a day.

3 Cumulative History Functions and the MCF

3.1 Cumulative History Functions

Cumulative history functions are time-event plots. They are an alternate depiction of recurrence data (Nelson, 2003). The cumulative number of recurrences, or cost, is shown as a function of time. For discrete data, this function is a staircase plot, as in Figure 2 below. Unless recurrences occur simultaneously, a cumulative history function for the number of recurrences will have steps of equal height, while those in the cost function vary.

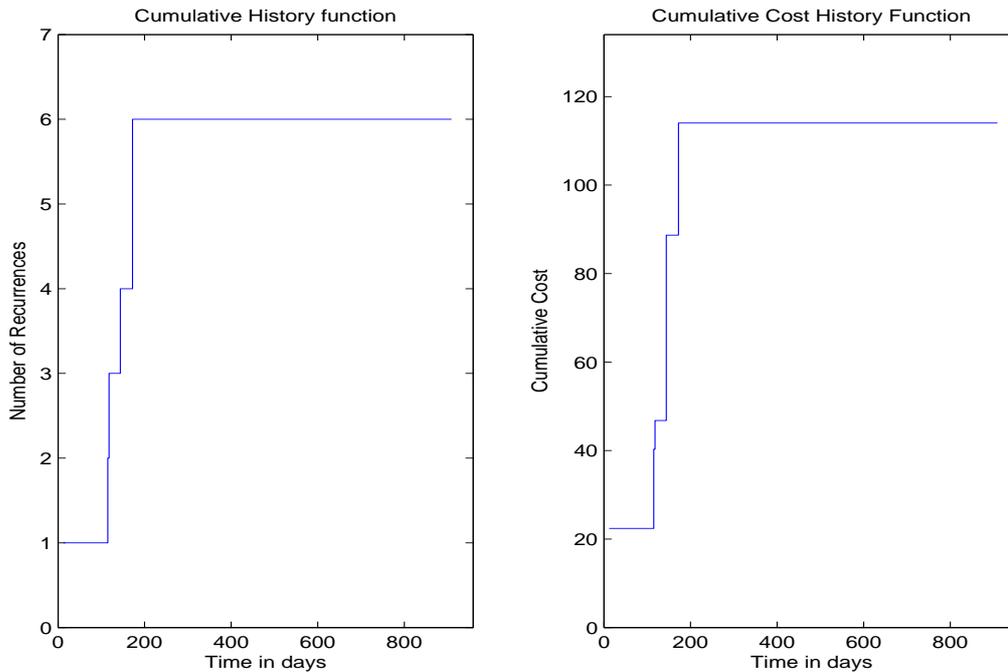


Figure 2: Lifespan of a sample unit from the General Motors warranty database.

Figure 2 presents a cumulative history functions for just one unit in the sample. The graph on the left shows the cumulative history function of the number of recurrences, while the graph on the right shows the cumulative history function of the cost of recurrences. Graphs of more than one function involve multiple plots on one graph. In this way, the cumulative history function of a population can be displayed. For a population, the cumulative history function for a unit is regarded as uncensored.

3.2 The Mean Cumulative Function (MCF)

At any time t_1 , the distribution of the data has a corresponding mean. This mean, as a function of time, is called the *mean cumulative function* (MCF) (Nelson, 2003). The MCF tells us the average number, or cost, of recurrences for a unit, up to time t . The curve $M(t)$ is a curve displaying the MCF for every point in time, t , and is generally regarded as continuous. The MCF is a summary of the population cumulative history functions.

3.3 Finding the MCF for Exact Age Data with censoring

Following Nelson (2003), we describe a method of calculating the MCF.

To find the MCF, a sample of units with their cumulative history functions is required. For right censored data, the number of units in the sample varies, decreasing for every censored sample unit. The number of uncensored, or "at risk" units must be recorded. The MCF can be worked out cumulatively on a claim by claim process. Between claims, the MCF is constant.

Nelson's step-by-step method, to find the MCF is as follows:

- (i) In a table, list all sample recurrence and censoring ages in order from smallest to largest.

- (ii) Record the number at risk for each row of the table in the second column. For rows containing a claim, the number at risk remains constant. For a censoring row, the number at risk decreases by one. For the first column, suppose there is a row zero, with number at risk equal to the sample size.
- (iii) For each claim, record the observed incremental mean number of recurrences, (or mean cost) per unit. For recurrences, this is $1/n$, where n is the number of units at risk. For costs, this is c/n , where c is the cost of the recurrence.
- (iv) The MCF is the cumulative mean number of recurrences (or mean cost). To find the MCF at each claim, add the incremental mean number of recurrences (or mean cost) to the MCF at the previous claim. The first claim has MCF equal to its mean number of recurrences (or mean cost).

This method produces a nonparametric, staircase estimate for the MCF, and does not require any assumptions about the distribution of the population. As the sample size increases, the estimate will approach the true MCF curve. An example of this method implemented is given in section 5.2.

3.4 MCF for Left Censored and Gap Recurrent Data

The method in section 3.3, can easily be adapted to left censored and gap data. In the second step, record the number at risk as previously done, but this time, as some units may be entering the sample as well as leaving, increment the number at risk for every left-censoring time, or at the end of a gap. In this way, the number at risk still reflects how many sample units are being observed at any given time. The rest of the steps are carried out as before.

4 MCF Confidence Intervals for Right Censored Data

There are two alternative methods for calculating the confidence limits for the MCF. The first type of limits, *Naive Limits*, are also known as *Poisson Limits*. The Naive limits are restricted to only MCF for the number of recurrences, and require additional assumptions. They are however, easily calculated, and the only option if the individual history of each sample unit is unknown.

The other type of limits, called here *Nelson Confidence Limits* are described below. The difficulty in calculating these lies in finding the variance. This must be found separately for each individual claim, and involves lengthy summation of various counts and costs multiplied together. This is described in full in section 4.2.

4.1 Naive (Poisson) Confidence Limits

Naive Confidence Limits are calculatable only for the number of discrete recurrences (not cost) in exact age data with right censoring. They can be extended to exact age data with left censoring and gaps, as well as interval age data, however this is not discussed in this paper.

Naive confidence intervals require the assumption that the failure process forms a non-homogeneous Poisson process. This assumption implies the increments of mean

number of recurrences are statistically independent. The MCF is the sum of these increments, and as they are independent, its variance is the sum of each increments individual variance. Let m_i be the i^{th} increment of the mean number of recurrences, and M_i is the MCF at increment i . Let r_i be the number of units at risk in the i^{th} increment, and let $M_0 = 0$. Then:

$$V(M_i^*) = \frac{M_i - M_{i-1}}{r_i}$$

as the failure process is assumed to be a Non-Homogeneous Poisson Process.

Therefore:

$$\begin{aligned} V(M_n^*) &= V(M_1 + M_2 + \dots + M_n) \\ &= V(M_1) + V(M_2) + \dots + V(M_n) \\ &= \frac{M_1}{r_1} + \frac{M_2 - M_1}{r_2} + \dots + \frac{M_n - M_{n-1}}{r_n}, \text{ and} \\ v(M_i) &= \frac{m_1}{r_1} + \frac{m_2}{r_2} + \dots + \frac{m_n}{r_n} \\ &= \frac{1}{r_1^2} + \frac{1}{r_2^2} + \dots + \frac{1}{r_n^2} \end{aligned}$$

The last line follows as $m_i = \frac{1}{r_i}$. This can be substituted into the 5% significance level confidence limit formula as follows:

$$M_i^* \pm 1.96\sqrt{V(M_i^*)}$$

These limits are often shorter than Nelson's limits, described in section 4.2. This can give a false impression of the accuracy of the MCF estimate, and for this reason, and that of the somewhat dubious non-homogeneous Poisson assumption that is required, these limits are not very reliable.

4.2 Nelson's Confidence Limits

Nelson's more accurate limits have the added complexity of requiring the knowledge of every sample units individual history. They are, however, appropriate for cost data, and do not require any particular assumption on the failure process. The formula for the 5% significance level confidence limits is:

$$M^*(t) \pm 1.96\sqrt{V(M^*(t))} \tag{1}$$

where $M^*(t)$ is the MCF at time t , and $V(M^*(t))$ is the variance of the MCF at time t . The MCF can be found by the method in section 3.3, and the variance can be found as described below, again, following Nelson (2003).

Let $Y_{i,j}$ be the number of recurrences (or cost if MCF is for cost) in interval i for unit j .

- (i) List all the censoring times from earliest to latest. Label the unit with the first censoring time, N , where N is the number of units, and label the next $N - 1$, and so on.

- (ii) Divide each units cost history into intervals, where each new interval begins at the next censoring time.
- (iii) Let I be the interval that the time t falls into. The variance is a sum involving all the intervals up to and including interval I . The formula is as follows:

$$V(M^*(t)) = \frac{1}{N}V(Y_{N,n}) + \frac{1}{N-1}V(Y_{N-1,n}) + \dots + \frac{1}{I}V(Y_{I,n}) \\ + \sum_{i=I+1}^N \left(\frac{2}{i} \sum_{j=I}^i Cov(Y_{i,n}, Y_{j,n}) \right)$$

where:

$$V(Y_{k,n}) = \sum_{n=1}^k \frac{(Y_{k,n} - \bar{Y}_{k,\cdot})^2}{k-1} \\ \bar{Y}_{k,\cdot} = \frac{Y_{k,k} + Y_{k,k-1} + \dots + Y_{k,1}}{k} \quad \text{mean } Y \text{ for interval } k \\ Cov(Y_{k,n}, Y_{k',n}) = \sum_{n=1}^{k'} \frac{(Y_{k,n} - \bar{Y}_{k,\cdot})(Y_{k',n} - \bar{Y}_{k',\cdot})}{k'-1} \quad \text{for } k' < k.$$

For interval I (the interval containing t), the calculations are made only up to time t , as opposed to the censoring time at the end of the interval.

5 MCF and Confidence Limits for Data from the General Motors Warranty Database

5.1 Description of the Data

The data in this paper consists of 1596 warranty claims, from a total of 44890 vehicles, 43581 of which had no claims. The data was provided by the North American Operation of General Motors, for the model year beginning 2000.

The size of the dataset was too large to use, when calculating the Nelson Confidence Limits, so a random sample was taken. This was taken by generating random numbers in MatLab between 0 and 44890, rounding up, and taking the correspondingly numbered unit as part of the sample. To insure no repetitions, the units chosen for the sample were given a flag. If a flagged unit was rechosen, the loop repeated itself until an unflagged unit was chosen. The claims for the chosen vehicles were then matched, using the vehicle identification number (vid).

5.2 MCF

Using MatLab, the process described in section 3.3 was coded, and the MCF with corresponding confidence limits was computed. Figure 3, shows the MCF, as well as the associated confidence limits.

5.3 Confidence Limits

Calculating the Naive confidence limits in Matlab was a simple task, and with such short computational time, it was possible to find the confidence limits for the entire dataset.

The Nelson confidence limits required more computation time. To minimize the time taken, the program created was designed to minimize repeated computation. However, the matrix designed to hold the values of $Y_{i,n} - \bar{Y}_{i,\cdot}$ was too large for the version of MatLab used. With such size restrictions, a sample of the original data was taken instead, as described in 5.1. Once the variance was computed, it was placed into the formula in 4.2 for each associated claim MCF, and the confidence limits were outputted. The results are shown in Figure 3 and Figure 4.

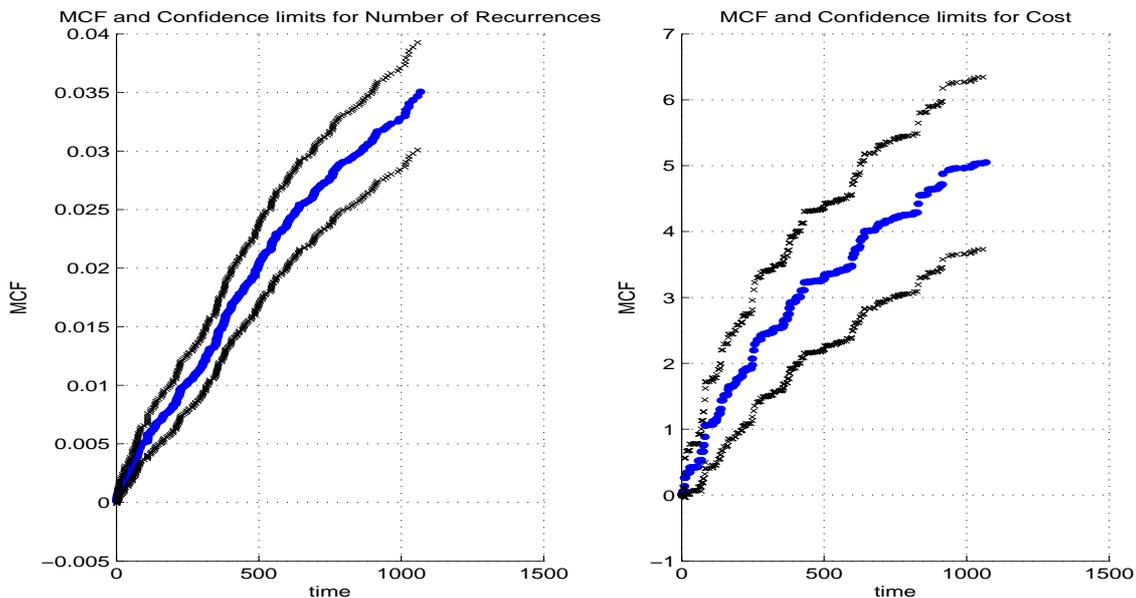


Figure 3: MCF and Nelson's Confidence Limits for a sample of 10,000 vehicles from General Motors Data, left for number of recurrences, right for cost.

Part of the Matlab code for computation of the MCF and its confidence limits is provided in the Appendix.

5.4 Description of Results

Figure 3 and Figure 4 display the MCF and confidence limits for General Motors data. The confidence limits for cost could only be found by Nelson's method. Due to the size of the data, and size restriction in Matlab, a sample of 10,000 vehicles was taken. Within this sample there were approximately 500 claims. The MCF for cost curve in Figure 3 is not smooth. This might be due to multiple claims for a single vehicle occurring within a small amount of time, incurring a high cost. The MCF curve for number of recurrences appears smooth, so it is more likely the bumps in the cost graph are due to fluctuations in cost.

Figure 4 shows the MCF for the entire sample, which shows a near straight line, with slight concavity. This implies the cost of repairs per unit is higher near the beginning of the lifespan, as is common with many products, often due to manufacturing

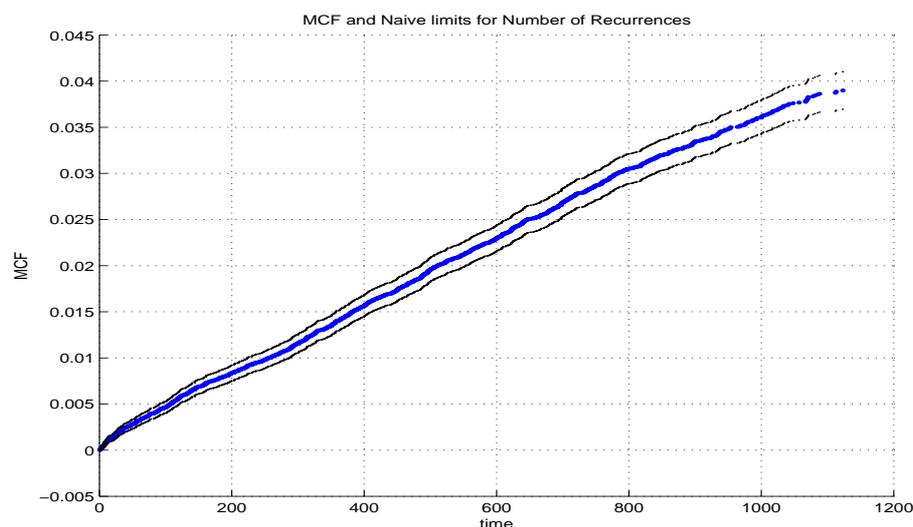


Figure 4: MCF and Naive Confidence Limits, for Number of Recurrences, for all 44890 vehicles from General Motors Data

faults. The Naive confidence limits give a much narrower range than the Nelson confidence limits. The Nelson confidence limits are more appropriate, however are over a smaller sample, so once again give less accurate results.

The graphs for the entire sample show a smooth curve for the MCF of the number of recurrences, and a fairly even curve for the MCF for cost. These are in reality staircase functions, however due to the large size of the sample, the graphs tend towards a continuous curve, and appear that way here.

6 Summary

The nonparametric estimates of $M(t)$ produced here are easily calculated in any spreadsheet type program. When data with no perceivable distribution is in debate, the nonparametric MCF is an ideal starting point, to learn more about the data. The lack of assumptions provides results that are accurate, and the confidence limits give bounds on this accuracy. This is often preferable to many parametric estimates, where if the assumptions are not met, the estimates are then invalid. In these cases the confidence limits are meaningless, and a false impression of the data results.

The MCF is of interest to the financial department of General Motors, to estimate the average warranty expenses per car for certain period of time which is valuable information in maintaining the warranty expected in general. Also, the MCF provides the engineers with valuable information on the failure rate of the automobiles as they age.

Appendix

Next we provide the code to find the MCF for cost, written in Matlab, using the data described in section 5.1:

```
function mt = M_tCost(data,cen) %cen is the sample size
```

```

%The data brought in is unsorted:
sortedData = sortrows(data, 2); %sorts by claim/censoring age.
dimsData = size(sortedData); %number of rows and columns
numRows = dimsData(1);
r = numRows-cen; %r is the number of claims
numRisk = zeros(numRows,1); %counts the #units still at risk

%set first value of # at risk and find row containing the first claim.
if sortedData(1,3)==1
    numRisk(1)=cen;
    firstClaim = 1;
else numRisk(1)=cen-1;
    found=0;
    n=2; %counter
    while(~found)
        if sortedData(n,3)==1
            found=1;
            firstClaim=n; %row n contains the first claim
        end
        n=n+1;
    end
end
%set all other values of numRisk
for n = 2:numRows
    if sortedData(n,3)==0 %if it is a censoring time
        numRisk(n)=numRisk(n-1)-1;
    else
        numRisk(n)=numRisk(n-1);
    end
end

finalData = [zeros(r,1) zeros(r,1)]; %matrix of data: claim time, MCF
meanCost = zeros(r,1); %mean Number of claims. Other rows have a zero placeholder.

%set first non-zero values
meanCost(1)=sortedData(firstClaim,4)/numRisk(firstClaim);
finalData(1,1)= sortedData(firstClaim,2);
finalData(1,2)= meanCost(1);
claimNum = 2; %set counter
for n=(firstClaim+1):numRows
    if sortedData(n,3)==1
        finalData(claimNum,1) = sortedData(n,2); %claim time
        meanCost(claimNum)=sortedData(n,4)/numRisk(n);
        finalData(claimNum,2) = finalData(claimNum-1,2) + meanCost(claimNum); %MCF
        claimNum=claimNum+1; %increment the counter
    end
end
mt = finalData;

```

For access to the rest of the code used to write this paper, please email brwyn.erasmuson@mcs.vuw.ac.nz

References

- Nelson, W B. 2003. *Recurrent Event Data Analysis for Product Repairs, Disease Recurrences, and Other Applications*. Society for Industrial and Applied Mathematics.

Evaluating the Performance of Radiotherapy Design Models

Vitesh V. Bava
Department of Engineering Science
University of Auckland
New Zealand
vbav001@ec.auckland.ac.nz

Abstract

In this paper we have evaluated the performance of different radiotherapy planning models. Several linear programming models exist for radiotherapy, each having its own set of constraints and its own objective function. Ten different models are evaluated for four different cancer cases to compare how they perform relative to one another.

A process called Data Envelopment Analysis (DEA) is used to evaluate the models. This technique uses a set of inputs and outputs for several systems called DMUs and generates a set of weights and efficiency scores which can be analysed to evaluate the performance of the DMUs. In this project, the DMUs are the linear models, the inputs are the different cancer cases and the outputs are a list of performance measures generated from each treatment plan.

Through analysing the results from DEA, it is found that different models perform well for different performance measures, and also many of the models perform equally across many performance measures for all cancer cases. These results suggest that choosing a model should be based on which performance measures are considered to be the most important for a particular cancer case.

1 Introduction

Cancer is one of the largest causes of health related death in many countries around the world, therefore it can become a major concern for people if they themselves or people close to them become affected. The site within the body where the cancerous cells are located is called the tumour, and may be surrounded by critical organs which the life of the patient may be dependent on (Wikipedia (2006)).

Radiotherapy treatment can be used to completely destroy a tumour by delivering ionizing radiation to the cancerous cells. However care must be taken to minimize the amount of radiation delivered to any critical organs which may be nearby. The treatment involves a radiation source which, in theory, can travel in a full circle around a patient's body. We can divide a cross section of the patient's body up into discrete elements called *voxels*, and we can also divide the different beams from different positions into *bixels*. Different amounts of radiation from different bixels can be delivered to different sets of voxels from different angles of the radiation source. Therefore a treatment plan needs to be generated to enable a safe and optimal radiotherapy treatment for the patient. To achieve this, a linear programming model is solved. However, several linear programming models exist. Furthermore, there also

exist many performance measures which can be used to determine how well a particular model is performing. Due to the variety of performance measures, there is no consensus on what treatment plan is considered ideal, therefore the performance of each linear model should be evaluated. This project is concerned with evaluating the performance of several linear models in radiotherapy (Wikipedia (2006), Hamacher and Küfer (2002)).

1.1 Linear Programming Models

To use linear programming to obtain an optimal treatment plan, we must first have the cancer case represented in matrix form so it can be used in a computer. A CT image of the patient's body is converted into a set of voxels and from this a prescription matrix is generated, which describes each voxel and what type of tissue that voxel represents (cancerous, critical or regular). A dose deposition matrix, A_{ij} , is generated which describes how the intensity from each bixel j contributes to the total radiation delivered to each voxel i . An x vector of decision variables corresponds to the radiation intensity for each bixel in the treatment configuration, and a right hand side (b) vector is defined which corresponds to lower or upper bounds on the dose to all the voxels in the prescription. When a linear model is formulated and solved, we obtain a treatment plan which is described by the x vector. We can then calculate Ax which describes the dose distribution delivered to all voxels within the patient (Billups & Kennedy (2003), Belli, Lane, Morrill & Rosen (1991)).

A number of linear models exist for radiotherapy treatment, each having different sets of constraints and objective functions to achieve different desired outcomes. For this project, ten different linear models will be tested against each other. When we have the necessary information represented in the computer from the CT image, these ten linear models can be implemented into MATLAB and then solved.

1.2 RAD and Performance Measures

To use optimization to come up with a treatment plan, a piece of software called Radiotherapy optimAl Design, better known as RAD, is used through the program MATLAB. This program enables a user to generate their own cancer cases and then solve that case using an optimization model (Holder, (2000)).

A model which is considered to be ideal is one which:

- Delivers a uniform dose to the tumorous region, destroying all tumorous cells completely.
- Delivers as little radiation to critical structures as possible
- Minimises the number of excessively high radiation doses to voxels which are not tumorous.

Performance measures allow us to test how well these guidelines have been followed by each linear model. When we have solved the linear model, we calculate Ax which tells us the amount of radiation received by each voxel in our prescription. We can then use this information to generate the values for our list of performance measures. The list of performance measures is as follows:

Total radiation dose received by

- Tumorous structures
- Critical structures

- Regular structures
 - All structures
- Average radiation dose received by

- Tumorous structures
- Critical structures
- Regular structures

Maximum deviation from

- Tumour lower bound (TLB)
- Tumour upper bound (TUB)
- Critical structure upper bound (CUB)
- Regular structure upper bound (RUB)

An ideal model will minimize as many of these performance measures as possible. Measures will also be included which can be used to judge the performance of an optimal treatment plan are conformity index ratios, called *Ian Paddick Conformality Index* (IPCI) ratios (Cheek, Holder, Fuss & Salter, (2004)). These ratios represent how well the radiation dose on the patient matches the shape of the tumour. The objective functions of each of the linear models are also themselves performance measures. The objective functions which will be added to the list of performance measures are as follows:

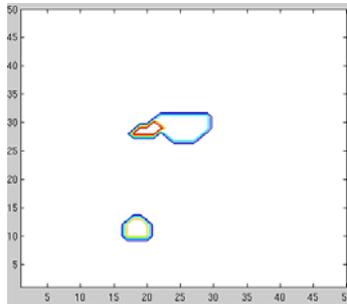
- Total amount of radiation delivered to all the voxels
- Total weighted dose to all structures
- Total weighted dose to non tumorous structures
- Total weighted dose to non tumorous structures (with a different set of weights)
- Maximum deviation from the prescribed dose to the tumour
- Difference between the radiation received by the tumour and the critical structures
- Maximum amount of radiation received by any voxel in the critical structure
- Sum of maximum amount of tumour under dosage and maximum critical/regular overdose
- Sum of average amount of tumour dose deviation, average amount of critical dosage and average amount of regular over dosage
- Sum of maximum amount of tumour under dosage and maximum critical/regular overdose

The goal of this project is to formulate a mathematical approach to evaluate how each of the linear models are performing according to the list of performance measures with respect to different cancer types.

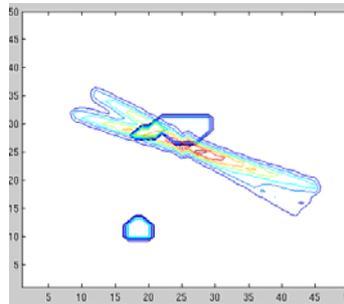
2 Methodology

Ten linear programming models exist which can be used to obtain optimal treatment plans in radiotherapy, each having their own objective functions and set of constraints. Furthermore, we have four different, real cancer cases and an artificial case which these models can be tested on. In this project we are assuming that these cancer cases are good representatives of their particular cancer types. Therefore when we see how our linear models are performing on these examples, we will assume that this is how the models will behave in any case of that particular cancer.

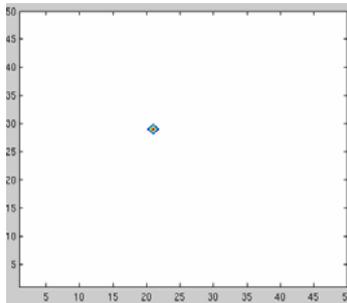
Figure 1 shows contour plots of the prescription matrices, together with an example treatment plan for each on the right hand side. The plans were generated from RAD by solving one of the linear models for each cancer case.



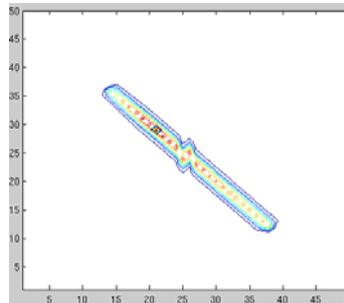
Acoustic Neuroma (Brain) Cancer



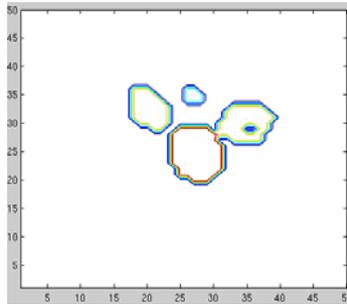
Model 1 Treatment Plan on Acoustic Cancer



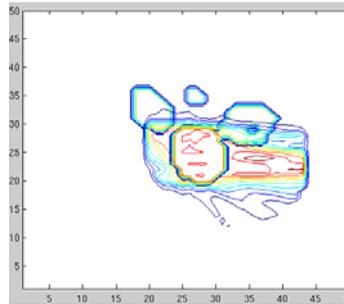
AVM (Arteriovenous Malformation) Cancer



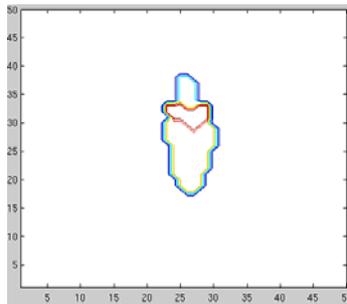
Model 1a Treatment Plan on AVM Cancer



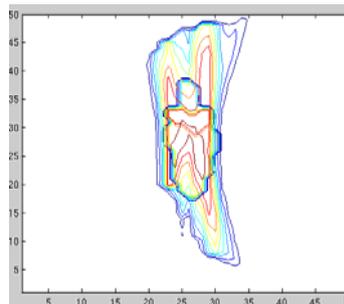
Lung Cancer



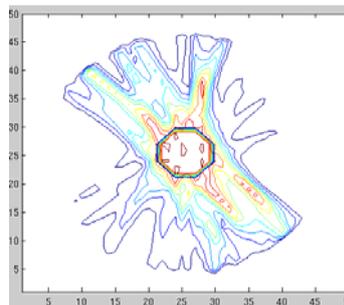
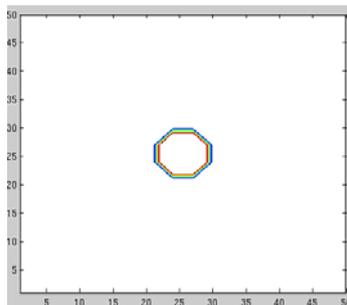
Model 1b Treatment Plan on Lung Cancer



Prostate Cancer



Absolute Model Plan on Prostate Cancer



*Circle Target Case**Average Model Plan on Circle Target Case*

Figure 1. Cancer Cases

We wish to solve every linear model on every one of these cancer cases and then evaluate their performances against one another. The circle target cancer case is an artificial example which was generated to test the angle placement optimization procedures which we will see later in section 2.4. Out of interest, we will also test our linear models on this example. The method which will be used to compare the linear models is a mathematical technique called *Data Envelopment Analysis*, or DEA. To implement DEA, RAD has been modified and new functions have been written. Many of the existing functions in RAD have also been used.

2.1 Introduction to Data Envelopment Analysis

DEA is a linear programming based procedure which can be used to evaluate the performance of a *system* with respect to a set of *inputs* and *outputs* used and generated by that system. The system is referred to as a *Decision Making Unit* or a DMU, which takes a set of inputs and then generates a set of outputs.

To evaluate the performance of these DMUs, we can solve an optimization model and obtain an efficiency score and a set of weights for each of the inputs and outputs for each of the DMUs, which will give us an idea of how that particular DMU is performing with respect to the others. This process is called DEA. If a particular DMU, with respect to the other DMUs, is producing larger output values for a given set of inputs, or is producing a given set of outputs with smaller input values, then it is considered to be efficient with respect to those other DMUs. The DEA optimization model is used to drive each DMU to efficiency with respect to all DMUs in the model, and involves allowing the DMU to choose most favourable values for the weights to determine how the inputs and outputs are contributing to this efficiency [10]. The DEA model can be written as follows.

Indices

k = output index

i = input index

j = DMU index

Parameters

Y = input matrix

X = output matrix

p = total number of outputs

m = total number of inputs

n = total number of DMUs

Decision Variables

u_k = weight for output k

v_i = weight for input i

DEA Model

1) Maximise $(\sum_{k=1}^p u_k Y_{kj}) / (\sum_{i=1}^m v_i X_{ij})$

2) $(\sum_{k=1}^p u_k Y_{kj}) / (\sum_{i=1}^m v_i X_{ij}) \leq 1$ for $j=1, \dots, n$

Explanation

- 1) Objective is to maximize the efficiency of the DMU
- 2) Maximum efficiency score possible is 1.

To enable easier computation, this model can be converted to linear form by realising that we are maximising a fraction, therefore the ratio between the numerator and denominator is of interest rather than their actual values. This means we can set the denominator equal to a constant and maximise the numerator (Emrouznejad, (2006)).

Our model now becomes:

Indices

p = total number of outputs

j = DMU index

Parameters

Y = input

X = output

Decision Variables

μ = weight for output

ν = weight for input

DEA Model

- 1) Maximise $\mu^T Y^j$
- 2) $\nu^T X^j = 1$
- 3) $\mu^T Y - \nu^T X^j \leq 0$
- 4) $\mu, \nu \geq \varepsilon 1$
- 5) $\varepsilon > 0$

Explanation

- 1) Maximize efficiency of the DMU
- 2) Maximum efficiency of a DMU is 1
- 3) Rearrangement of constraint\
- 4) Weights must not be negligible
- 5) Parameter to ensure non negligible weights

With this linear optimization model, we can take a set of DMUs with their inputs and outputs and perform the DEA process which is illustrated in figure 2:

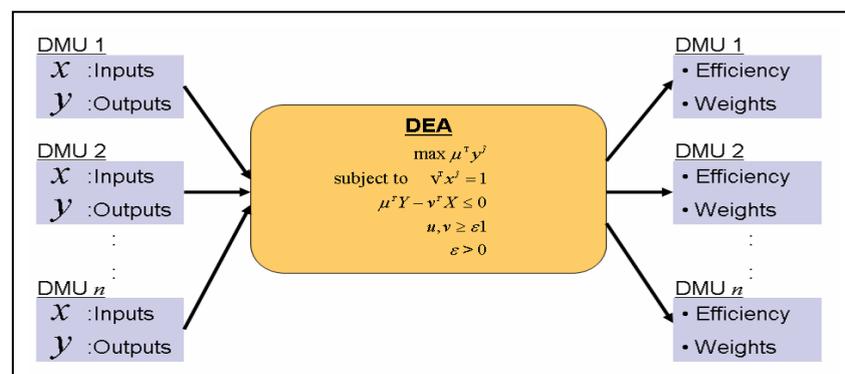


Figure 2. General DEA Process

We interpret the results by observing the efficiency scores and the weights. The DMUs which score an efficiency of 1 are considered to be *efficient*, meaning they convert their inputs to outputs in an efficient way. The weights give us an idea of which inputs and outputs lead to this efficiency, and therefore which inputs and outputs this particular DMU performs well for (Emrouznejad, (2006), Cooper, Seiford & Zhu, (2004), Cheek, Holder, Fuss & Salter, (2004)).

2.2 Application of DEA in Radiotherapy

When we apply DEA to radiotherapy, our DMUs become the different linear models we wish to evaluate the performance of. The inputs for each of the models is the data representing a particular cancer case, and is the same for each model, i.e. each linear model is applied to the same particular cancer case. This fact enables us to set $x_{ij} = 1$ for all DMUs. We can also consider the weights corresponding to the cancer case inputs to be equal for all DMUs, which enables us to set $v^T = 1$. Our DEA model becomes:

DEA Model

- 1) Maximise $\mu^T Y^j$
- 2) $\mu^T Y \leq 1$
- 3) $\mu \geq \varepsilon 1$
- 4) $\varepsilon > 0$

Explanation

- 1) Maximize efficiency of the DMU
- 2) Rearrangement of constraint
- 3) Weights must not be negligible
- 4) Parameter to ensure non negligible weights

The new DEA process is now represented in figure 3.

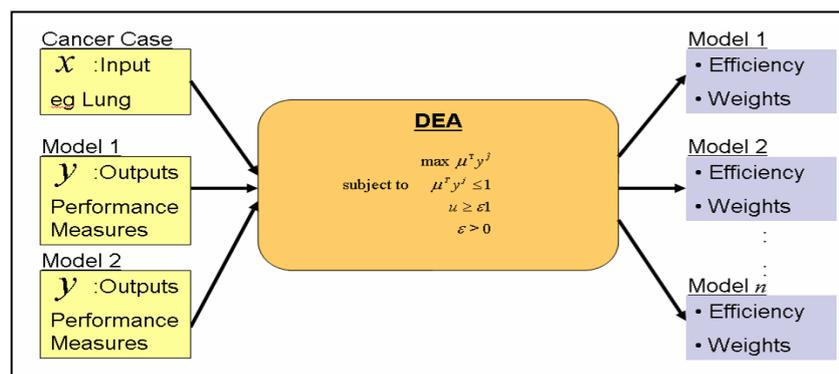


Figure 3. Radiotherapy Application of DEA

The outputs y_{ij} for each DMU become the performance measures for each of the linear models which we calculated using the optimal treatment plans. We are interested in minimizing the performance measures as much as possible, therefore before inputting them into the DEA model we must swap the signs of their values, due to the fact that the DEA model finds larger outputs more favourable.

The linear models who score an efficiency of 1 correspond to those models which are 'performing well' with respect to certain performance measures. The weights corresponding to each performance measure give us an idea of which performance

measures a particular model is performing well for. For example, if a model scores an efficiency of 1 and has large weights for the tumour lower bound and the critical structure upper bound performance measures, it suggests the model is performing well for these measures which is leading to its perfect efficiency score. Zero weights on performance measures do not necessarily mean that a particular model is performing badly there, but instead shows that those performance measures cannot be used to compare the respective models. The zero weights could also mean that all the models are performing equally compared to one another for those measures. In general, the weights from the DEA results will tell us how the models are performing relative to each other for the different performance measures. The larger the weight, the better that model is doing against the other models for that performance measure. A smaller weight indicates that the difference in performance for that measure between that model and the other models is not as significant.

We can run the DEA process each time a particular cancer case has been solved by all the linear models. This will show us which models are performing well for that cancer case.

2.3 Implementation of DEA in RAD

RAD was modified to implement the DEA algorithm to evaluate the performance of each model. Functions were written which read in data files which contain information about each of the different cancer cases, and then the appropriate matrices and vectors needed for the models were created.

The different cancer cases were then passed through each of the ten linear models to obtain ten different treatment plans. From these plans, a list of performance measures for each model were calculated, and then stored in an *outputs* matrix. This matrix became the input for DEA function. When the DEA model was performed on all the linear models for each cancer case, the results were stored in a *weights* matrix. The data from this matrix was then imported into excel for graphing and analysis.

2.4 Optimization of Beam Directions

So far 72 beam angle positions have been allowed when solving each linear model. Utilising all of these bixels and angles would enable a more accurate treatment plan to be generated, one which will very closely comply with the requirements of the prescription. However, if radiation is to be delivered from many angles, the treatment time will increase, therefore the patient will need to remain stationary for a greater period of time, which could be difficult. In general, only a small number of beam directions out of those available to us should be chosen to reduce the amount of radiation time, therefore the position of these angles around the patient also needs to be optimized to evaluate an optimal treatment plan (Ehrgott, & Johnston, (2003), Chhagan, (2004), Ehrgott, Holder & Reese, (2005)).

New functions were implemented into the RAD program which take the problem structure and then, through running beam optimization procedures, will choose an optimal set of angles out of those available. From these new angles, a new problem structure for each cancer case is created, which are then used in RAD instead of the original problem structures.

3 Results

Analysing the weights from DEA does not allow us to see this. All DEA tells us is how the models are performing relative to one another.

4 Conclusions

- In general, different models have weights attached to different performance measures, therefore a single model cannot be suggested for a particular cancer case. Our selection of a model will therefore be based on what performance measures we consider to be most important.
- The existence of zero weights show that many of the performance measures cannot be used to compare the performance of the models.
- DEA analysis alone is not a sufficient way to accurately determine how each of the models are behaving, and therefore may not allow us to see important information such as the dose distribution.

5 References

- Belli, J., Lane, R., Morrill, S. & Rosen, T., (1991) "Treatment plan optimization using linear programming," *Medical Physics*, vol.18(2), pp. 141-152.
- Billups, S, & Kennedy, J.M., (2003) "Minimum-Support Solutions for Radiotherapy Planning," *Annals of Operations Research*, vol.119, pp. 229-245.
- Cheek, D., Holder, A., Fuss, M. & Salter, B., (2004) "The Relationship Between the Number of Shots and the Quality of Gamma Knife Radiosurgeries", *Optimization and Engineering*, vol 6, pp. 449-462.
- Chhagan, S., (2004) *Optimisation of Beam Directions in Radiotherapy Planning*, Department of Engineering Science, The University of Auckland, Auckland, New Zealand.
- Cooper, W. W., Seiford, L. M. & Zhu, J., (2004) *Handbook On Data Envelopment Analysis*. Kluwer Academic Publishers. Dordrecht.
- Ehrgott, M., Holder, A., & Reese, J., (2005) *Beam Selection in Radiotherapy Design*, Technical Report 95, Department of Mathematics, Trinity University, USA.
- Ehrgott, M. & Johnston R., (2003) "Optimization of beam directions in intensity modulated radiation therapy planning," *OR Spectrum*, vol.25, pp. 251-264.
- Ali Emrouznejad's DEA Homepage*. Ali Emrouznejad's Data Envelopment Analysis. Retrieved June 18th, 2006 from <http://www.deazone.com/tutorial/Introduction.htm>.
- Hamacher, H. W. & Küfer, K. H., (2002) "Inverse radiation therapy planning – a multiple objective optimisation approach," *Discrete Applied Mathematics*, vol.118(1-2), pp. 145-161.
- Holder, A., (2003) "Designing Radiotherapy Plans with Elastic Constraints and Interior Point Methods", *Health Care Management Science*, vol. 6, pp. 5-16.
- Cancer*. Wikipedia, the free encyclopaedia. Retrieved September 5th, 2006 from <http://en.wikipedia.org/wiki/Cancer>.

Computational Models for Large Airline Network Revenue Management Problems

Michael J. Frankovich
 Department of Engineering Science
 The University of Auckland
 New Zealand
michaelfrankovich@gmail.com

Abstract

This paper seeks to improve computational models for optimizing the Airline Network Revenue Management (RM) Problem for medium to large networks. It is shown that the expected revenue obtained from a bid-pricing model using a common deterministic linear program (DLP) increases with the number of times the DLP is solved, the limiting case being to solve before every customer arrival. The application of Stochastic Dual Dynamic Programming (SDDP) is investigated for medium to large sized problems beyond the capability of conventional Stochastic Dynamic Programming (DP). SDDP produces higher expected revenues than the standard bid-pricing model for small to medium problems. In addition, better upper bounds on optimal expected revenues are obtained. Improvements to the DLP method have also been investigated. An outer approximation method which uses the DLP is proposed and shown to produce significantly better results than the standard bid-pricing model for single fare-class problems. A trigger function is then introduced, optimizing the re-solve points, and shown to further improve the performance for single fare-class problems. The improvements are less significant for multiple fare-class problems.

1 Outline of the Problem

1.1 The Network RM Problem

The methods described in this paper can be used to solve revenue maximization problems over medium to large networks. These problems are characterised by interdependence between products, which draw on common resources. This interrelatedness of the products makes these problems difficult to solve, particularly as problem size increases.

Resources are represented by edges of the network graph, while products are paths made up of a number of resources. The network is defined by the $m \times n$ matrix A , where

$$A_{ij} = \begin{cases} 1, & \text{if product } j \text{ uses resource } i \\ 0, & \text{otherwise.} \end{cases}$$

Here we deal specifically with airline networks, but this need not be the case. A resource is a seat on a flight leg and a product is a seat on an itinerary sold at a given price, using any number of flights. Figure 1.1 below is an example network taken from E.L. Williamson [8]. This network, along with its corresponding data, will form the main example used to test our methods. This problem has four *fare-classes*, during each of which a different price is

paid for the same itinerary, with prices increasing as departure draws nearer. Note that this difference in price produces *different products* for the *same itinerary*.

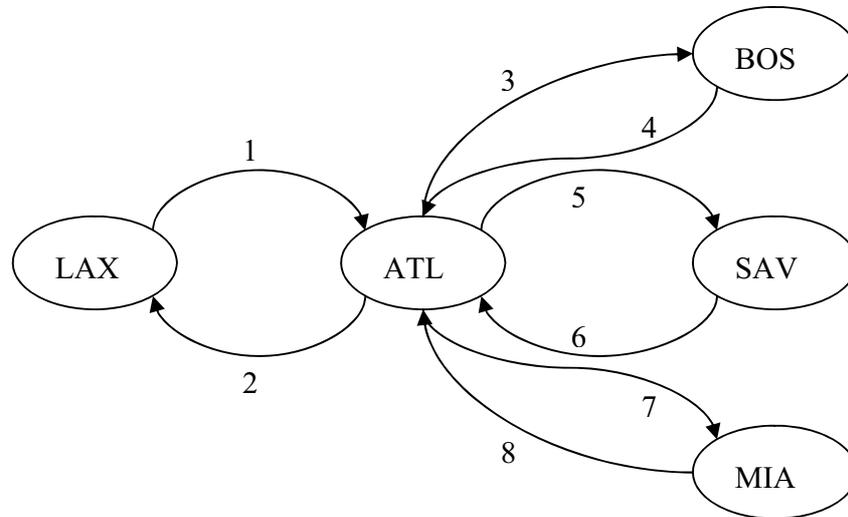


Figure 1.1: This is the American airport network from [8], which serves as the main test problem.

1.2 Modelling of the Arrival Process

We represent the arrival of customers by a Poisson process, assuming independence. In this way, we need no information about the distribution of the demand. We require solely the expected demand for each product, making models easier to implement.

The booking horizon is broken up into many small time intervals, during each of which there is some probability of each arrival. There are $n+1$ possible occurrences in each time interval: either the arrival of a single customer for product $j = 1, 2, \dots, n$, or no arrival at all ($j = 0$). Therefore given an arrival for product j , we have capacity $\mathbf{x}_t = \mathbf{x}_{t-1} - \varphi_t \mathbf{A}_j$, where $\varphi_t \in \{0, 1\}$ is our decision variable showing whether we accept the customer or not and \mathbf{A}_j is the j^{th} column of \mathbf{A} .

There is a significant increase in computational cost associated with allowing for the probability of no arrival, and so necessity demanded that we assume customers to arrive at regular intervals. We also assume them to arrive in order of increasing fare-class, $1, 2, \dots, f$.

We aim to come up with a policy which finds a balance somewhere between the following two extremes in order to maximize revenue:

- Accept every customer until resources are exhausted.
- Reject customers until expected max demand to go equals our initial resource level.

This policy will simply tell us whether to accept or decline a customer. Note that we only actually reject customers in computational simulations. In reality, airlines perform the equivalent action of making unavailable the product which the hypothetical customer was about to purchase. For example, “No seats currently available” may be displayed.

2 Dynamic Programming (DP) Approach

2.1 DP Formulation

Revenue maximization problems of the type we consider can be tackled using dynamic programming. Talluri and van Ryzin (p. 88) [7] propose a DP formulation for the airline network revenue management problem, on which ours is based:

We denote by $V_t(\mathbf{x})$ the maximum expected revenue that can be earned in periods $t, t+1, \dots, T$ when $\mathbf{x} = \mathbf{x}_t$. The decision on whether to accept a request in stage t is modelled by the control $\mathbf{u} \in U(\mathbf{x}) = \{0, 1\}^n \cap \{\mathbf{u}: \mathbf{A}\mathbf{u} \leq \mathbf{x}\}$. This gives the following recursion:

$$\text{DP: } V_t(\mathbf{x}) = E[\max_{\mathbf{u} \in U(\mathbf{x})} \{ \mathbf{P}_t^T \mathbf{u}(t, \mathbf{x}, \mathbf{p}) + V_{t+1}(\mathbf{x} - \mathbf{A}\mathbf{u}) \}],$$

with the boundary condition $V_{T+1}(\mathbf{x}) = 0$,

where \mathbf{P}_t is a vector containing the prices of products for which we have an arrival in stage t , and zeros for all other products.

In order to exactly evaluate the expected future value function for all values of \mathbf{x} and t for large state spaces, we would need an astronomical number of calculations. The main example problem used had 8 legs, with capacities of between 160 and 220 depending on the load factor, giving state spaces of between 160^8 and 220^8 , ruling out the possibility of an exact solution, with stochastic dynamic programming reaching its limit at powers of 5 or 6. Thus although stochastic dynamic programming appears a suitable approach, it is useless for all but a few problems of interest. Instead of discretely calculating the future value function, we can analytically approximate it. This can be done using an outer approximation method called Stochastic Dual Dynamic Programming.

2.2 Stochastic Dual Dynamic Programming (SDDP)

SDDP was developed by M.V.F. Pereira and L.M.V.G. Pinto [3] in 1988 as a tool to solve large stochastic scheduling problems for hydrothermal generating systems in Brazil. Instead of discretising the state-space like conventional dynamic programming, it analytically approximates the expected cost-to-go (or future value) function using convex polyhedral functions, created using Benders [1] cuts.

Consider the following two-stage maximization problem. We shall use notation taken from Philpott [5]:

$$\begin{aligned} \text{MP: } \quad & \text{Max } \mathbf{c}^T \mathbf{x} + V(\mathbf{x}) && \text{[duals]} \\ \text{s/t. } \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0 \\ \text{where } V(\mathbf{x}) = & \text{Max } \sum_{i=1}^N p_i d_i^T y_i \\ & \text{s/t. } \mathbf{W}\mathbf{y}_i = \mathbf{h}(\omega_i) - \mathbf{T}(\omega_i)\mathbf{x}, \quad \forall i = 1, 2, \dots, N \quad [\mathbf{p}_i \boldsymbol{\pi}(\omega_i)] \\ & \mathbf{y}_i \geq 0 \quad \forall i = 1, 2, \dots, N \end{aligned}$$

This problem can be represented as follows:

$$\begin{aligned} \text{MP: } \quad & \text{Max } \mathbf{c}^T \mathbf{x} + \theta \\ \text{s/t. } \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0 \\ & \theta \leq \sum_{i=1}^N p_i \pi_1(\omega_i)^T [\mathbf{h}(\omega_i) - \mathbf{T}(\omega_i)\mathbf{x}] \end{aligned}$$

$$\begin{aligned} \theta &\leq \sum_{i=1}^N p_i \pi_2(\omega_i)^T [h(\omega_i) - T(\omega_i)x] \\ &: \\ \theta &\leq \sum_{i=1}^N p_i \pi_K(\omega_i)^T [h(\omega_i) - T(\omega_i)x]. \end{aligned}$$

Each one of these K linear functions of θ in x is a “cut” representing a hyperplane in a multidimensional space. Its slope, termed the subgradient, is represented by the sum of the corresponding dual vectors of the second stage problem (these are scaled by their probabilities). If the second stage problem is solved using a first stage decision of x' , then the single cut obtained using the duals exactly represents the future value function at the point x' , and approximates it at all other points in the domain. The cut is a tangent to the true future value, as shown in the figure below:

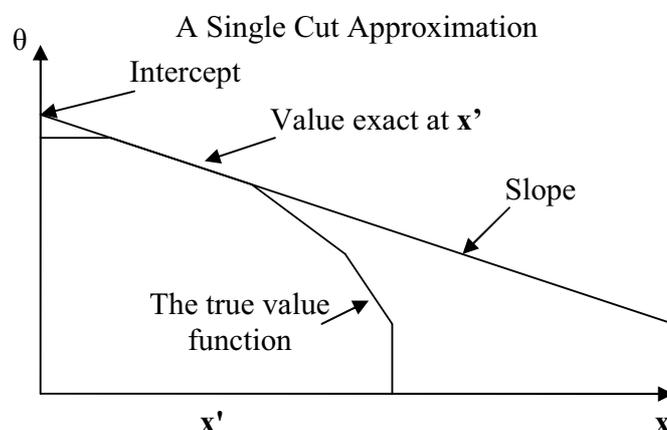


Figure 2.1: This shows, in one dimension of x , a single cut approximating the future value function.

SDDP applied to the airline network RM problem creates a series of cuts, the pointwise minimum of which represents the future value function in x , the capacity remaining. SDDP avoids the curse of dimensionality through its use of sampling. A forward simulation is performed, sampling a single scenario at each stage. This forward simulation sets values of x'_t for each time step. A backward pass is then carried out, performing $n+1$ single-variable optimizations at each stage (outlined later).

A simple algorithm for SDDP is as follows:

1. Simulate forwards, sampling a single scenario (i.e. arrival) at each stage, selling to every arriving customer if we have enough capacity remaining.
2. Roll backwards from $t = T$ to $t = 1$ one step at a time, optimizing at each stage to find a subgradient in each dimension and a future value, which are used to calculate a cut at stage $t-1$ in each dimension. Each optimization takes into account any cuts which have been made.
3. Simulate forwards as per (1), but using cuts made in the backward pass(es) to obtain a better optimized decision, ϕ , at each stage.
4. Repeat steps (2) and (3) until the upper bound has sufficiently converged or the maximum number of iterations has been reached. The upper bound is the expected stage one future value, since the cuts form an outer approximation.

5. Simulate forwards N times using the K sets of cuts produced above. The average revenue gained gives an estimate of the expected revenue from our candidate policy, and so is an estimate of a lower bound to the optimal solution.

Since our arrival process only allows for any single arrival j in a time step, we only have to decide whether to sell this seat or not, as determined by $\varphi \in \{0, 1\}$. This simplifies the analysis. Also, this can only be 1 when we have sufficient capacity remaining. In creating cuts, we shall allow φ to be non-integer (i.e. lying in $[0, 1]$).

$$\begin{aligned}
 Q(\mathbf{x}_{t-1}, j): \quad & \text{Max} \quad p_j(t)\varphi + \theta_{t+1} \\
 & \text{s/t.} \quad \varphi \leq u \\
 & \quad \mathbf{A}_j \varphi = \mathbf{x}_{t-1} - \mathbf{v} \\
 & \quad \alpha_{t+1}(k) + \boldsymbol{\beta}_{t+1}(k)^T \mathbf{v} \geq \theta_{t+1}, \quad k = 1, 2, \dots, K \\
 & \quad \varphi \geq 0,
 \end{aligned}$$

where j is the product for which a customer arrives in this period,
 \mathbf{v} is the vector of capacity at the end of period t ,
 \mathbf{x}_{t-1} is the known capacity at start of period t , henceforth referred to as \mathbf{x}'

$$\text{and } u = \min \left\{ \min \left\{ \frac{x'_i}{A_{ij}} : A_{ij} > 0 \right\}, 1 \right\}.$$

For $k = 1, 2, \dots, K$, let $\mathbf{a}(k) = \alpha_{t+1}(k) + \boldsymbol{\beta}_{t+1}(k)^T \mathbf{x}'$,
and $\mathbf{b}(k) = \boldsymbol{\beta}_{t+1}(k)^T \mathbf{A}_j$.

This problem is equivalent to

$$\begin{aligned}
 P(t, j): \quad & \text{Max} \quad p_j(t)\varphi + \theta_{t+1} && \text{[Duals]} \\
 & \text{s/t.} \quad \mathbf{b}(k)\varphi + \theta_{t+1} \leq \mathbf{a}(k), \quad k = 1, 2, \dots, K && [\mu(k)] \\
 & \quad \varphi \leq u && [\sigma] \\
 & \quad \varphi \geq 0.
 \end{aligned}$$

At each time step t in each backward pass k , we create a cut for the preceding stage. This cut has a slope vector $\boldsymbol{\beta}_{t+1}(k)$ calculated by taking the expectation of the subgradients, \mathbf{B}_j (obtained by solving the above Q/P), over all $n+1$ possible outcomes.

$$\boldsymbol{\beta}_{t+1}(k) = \sum_{j=0}^n P_{ji} B_j(\mathbf{x}'),$$

where P_{ji} is the probability of arrival j in the current fare-class, i .

In order to get the height of the cut, $H_{(t-1)j}$, we take the expectation of the optimal objective values z_j over all $n+1$ possible outcomes:

$$H_{(t-1)j} = \sum_{j=0}^n P_{ji} z_j.$$

The intercept of the cut is therefore $\alpha_{t+1}(k) = H_{(t-1)q} - \boldsymbol{\beta}_{t+1}(k)^T \mathbf{x}'$.

These slopes, $\boldsymbol{\beta}$, and intercepts, α , are stored. So with each backward pass another cut is added to the optimization problem Q , improving the representation of the expected future value function. Recalling that $V(\mathbf{x})$ is the optimal expected value to go at \mathbf{x} , this cut is represented by the inequality

$$V(x) \leq \alpha_{t+1}(k) - \boldsymbol{\beta}_{t+1}(k)^T (\mathbf{x}' - \mathbf{x}).$$

3 Deterministic Linear Programming Approach

3.1 The State of the Art

Currently, most airline companies practice network revenue management by solving a deterministic linear program (DLP) based on expected demand for each itinerary.

$$\begin{aligned}
 \text{DLP}(t): \quad \text{Max} \quad z &= \sum_{i=g}^f p_i^T y_i && [\text{duals}] \\
 \text{s/t.} \quad \sum_{i=g}^f A y_i &\leq x_t && [\boldsymbol{\pi}] \\
 y_i &\leq \frac{T-t}{T} E[D_{0i}], \quad \forall i = g, g+1, \dots, f && [\boldsymbol{\rho}_i] \\
 y_i &\geq 0, \quad \forall i = g, g+1, \dots, f
 \end{aligned}$$

where z is the revenue,

g is the fare-class we are in at time t ,

\mathbf{p}_i is the n -vector of the price of a seat on each itinerary in fare-class i ,

\mathbf{y}_i is the n -vector decision variable of how many of each itinerary sold in fare-class i ,

and $E[\mathbf{D}_{0i}]$ is the n -vector of initial expected demand for each itinerary in fare-class i .

Let \mathbf{A}_j be the j^{th} column of \mathbf{A} if $j \leq n$, or the $[(j-1) \% n + 1]^{\text{th}}$ column of \mathbf{A} if $j > n$. Now letting $\mathbf{p}^T = [p_1^T \quad p_2^T \quad \dots \quad p_f^T]$, we have the following policy:

- If $p_j < \sum_{i \in A_j} \pi_i$, then reject offer for purchase of product j .
- If $p_j \geq \sum_{i \in A_j} \pi_i$, then sell product j .

Cooper [2] has shown that as the size of this DLP problem is scaled up by multiplying $E[\mathbf{D}]$ and \mathbf{x} by some $k \rightarrow \infty$, the expected revenue converges to optimality. He also shows that the optimal solution value, z , to the DLP is an upper bound on the expected revenue of any feasible policy and is therefore an upper bound on the optimal expected revenue.

In order to improve the accuracy of this system, airlines may re-solve the above problem during the booking horizon, altering the right-hand side $E[\mathbf{D}]$ according to how far into the booking horizon they are, and the right-hand side \mathbf{x} according to how much they have sold. This yields a new $\boldsymbol{\pi}$, which is more accurate at $t+\Delta t$ than the old $\boldsymbol{\pi}$ at that instant.

3.2 DLP Cut Propagation

From $\text{DLP}(t)$, we can take the optimal expected value, z_t , and the associated dual variables, $\boldsymbol{\pi}_t$, of the capacity constraints to create a cut at $t-1$. Letting $V(\mathbf{x}_t)$ be the true value function, we have for a single fare-class model

$$\begin{aligned}
 V(\mathbf{x}_t) &\leq z_t - \boldsymbol{\pi}_t^T (\mathbf{x}'_t - \mathbf{x}_t), \\
 V(\mathbf{x}_t) &\leq \alpha_t + \boldsymbol{\beta}_t^T \mathbf{x}_t,
 \end{aligned}$$

where $\boldsymbol{\beta}_t = \boldsymbol{\pi}_t$ is the subgradient

and $\alpha_t = z_t - \boldsymbol{\beta}_t^T \mathbf{x}'_t$ is the intercept.

We can scale this cut down so that it applies throughout the entire horizon from t to T . The slope, $\boldsymbol{\beta}_t$, remains the same – all we need to do is adjust the height. It can be shown that at $t+\Delta t$, we have for single fare-class problems

$$\alpha_{t+\Delta t} = \frac{T - (t + \Delta t)}{T - t} \alpha_t.$$

From a single solution to the DLP at time t , $DLP(t)$, we have a single cut which can be propagated until the end of the booking horizon, T . So if at t we have solved K DLPs, we will have K sets of cuts $k = 1, 2, \dots, K$, each retaining valuable information placing a value on our capacity, π_{tk} . These cuts are each defined by an intercept, α_{tk} , and a slope, β_{tk} . We can then perform a simple single-variable optimization taking into account these cuts, as with SDDP, to make a decision at each stage.

For one solution at $t = 0$, this is exactly equivalent to using the single set of bid-prices from $DLP(0)$. For K solutions at t , it is equivalent to using the best of all previously calculated bid-prices depending on the current state of the system. We would thus expect better results from this policy than a bid-pricing policy.

For the multiple fare-class case, the scaling down of cuts is not as simple as for the single fare-class model. This is because the expected demand does not decrease uniformly over the whole booking horizon. $E[\mathbf{D}]$ as a function of t is piecewise linear, the bounds of each segment being the ends of each fare-class.

So for a multiple fare-class problem π_i , obtained in fare-class i , is no longer optimal along a given capacity/demand trajectory for fare-classes $i+1, i+2, \dots, f$. However, it does still represent a valid subgradient. This means that whilst π_i is optimal during fare-class i , cuts produced from it, although valid, do not necessarily support the true expected value function in subsequent fare-classes. Therefore for a single solve in a multiple fare-class model we would not expect an equivalent benefit to that gained in single fare-class model.

3.3 Triggering Re-solves of the DLP

It was of interest to develop a method for maximizing the expected revenue for any given number of re-solves, considering that airlines may be unable to re-solve after every single arrival. This meant determining when exactly in the booking horizon to perform the v re-solves. In other words, we want to get the best “value for money” from our re-solves.

The main consideration is how far our capacity sold has strayed from the “expected path”, since we know that if we are far from the expected path then our current cuts are likely to be poor and if we are on the expected path then there is clearly no need to re-solve. The approach used takes all past optimal bases $k = 1, 2, \dots, K$ from previous re-solves and tests them each with the right-hand side of $DLP(t)$ to obtain a candidate solution

$$u_t^k = B_k^{-1} \begin{bmatrix} x_t \\ E[D_t] \end{bmatrix}.$$

Note that B_k is made up of a set of columns from $\begin{bmatrix} A & I & 0 \\ I & 0 & I \end{bmatrix}$, being $m \times n \times f$ in length if we

let A be the concatenation of f network matrices A .

If u_t^k is feasible for any k then we have an optimal dual solution and do not need to re-solve. Otherwise, we can use each u_t^k to compute a difference between the current upper bound as determined by our solution obtained from cuts already made, and a lower bound calculated by rounding the infeasible parts of u_t^k up to zero.

Let $[y_k^T \ w_k^T \ z_k^T] = (u_i^k)^T$, and let $[\bar{y}_k^T \ \bar{w}_k^T \ \bar{z}_k^T]^T$ be the vector obtained by taking the positive part, $\max\{0, (u_i^k)_i\}$, of each component of u_i^k . Now let

$$L_k = \max_{k=1,2,\dots,K} \{y_k^T p + (-A\Delta y_k - \Delta w_k)^T \bar{\pi} + (-\Delta y_k - \Delta z_k)^T \bar{\rho} + (\Delta y_k)^T p\},$$

where $\Delta y_k = \bar{y}_k - y_k \geq 0$, $\Delta w_k = \bar{w}_k - w_k \geq 0$ and $\Delta z_k = \bar{z}_k - z_k \geq 0$. Now compute the optimal value using the cuts computed thus far. This is $U_k = \min_{k=1,2,\dots,K} \{a_k + b_k x_i\}$.

We then have $L_k \leq V(DLP(t)) = V(DualDLP(t)) \leq U_k$.

Philpott [4] has proved the validity of these bounds, which define our trigger function. If the value of $U_k - L_k$ is sufficiently large then we re-solve DLP(t) to obtain a new B_k and primal and dual solutions.

4 Results

Figure 4.1 below shows the significant effect of DLP cut propagation and the trigger function for single fare-class problems. For multiple fare-classes, this effect is reduced.

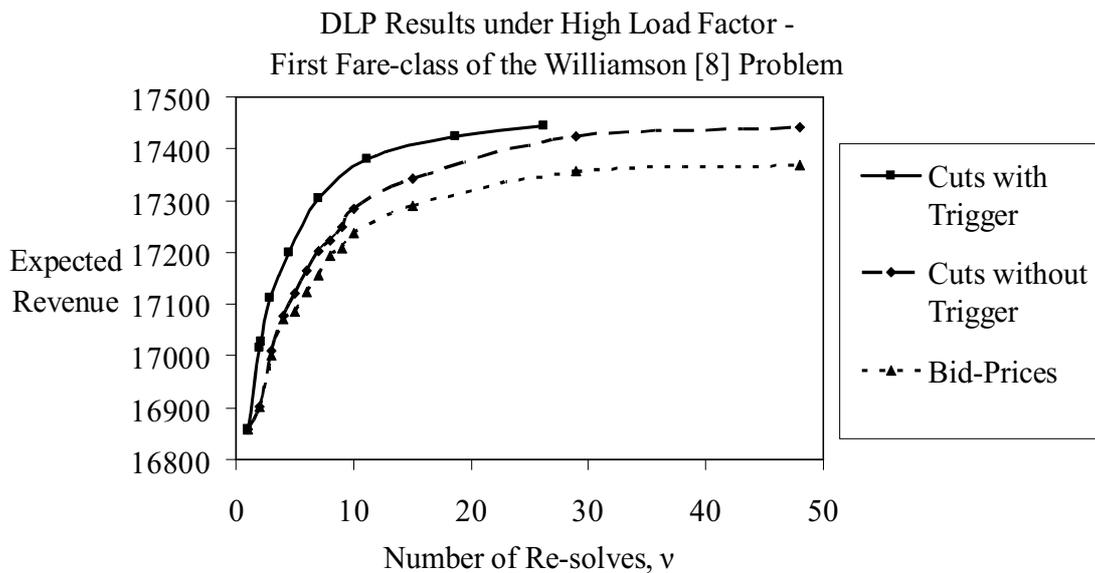


Figure 4.1: This shows the effect of cut propagation and the trigger for single fare-class problems.

Now we assumed airlines to be able to re-solve the DLP after every arrival. Expected revenues for the problem from [8] under a high load (obtained by setting each leg capacity to 160) are summarised in the following table:

	DLP	DLP with Cuts	SDDP
Cuts	-	3500	3500
E[R]	162113.5	162285.2	162183.5
95% C.I.	(162051.6, 162175.4)	(162223.3, 162347.2)	(162119.5, 162247.6)
Upper Bound	169136.6	169136.6	168830

Table 4.1: This shows 95% confidence intervals for the expected revenues under high load.

Expected revenues under a low load factor (setting each leg capacity to 200) are as follows:

	DLP	DLP with Cuts	SDDP
Cuts	-	3500	3500
E[R]	178337.6	178342.0	178506.7
95% C.I.	(178269.4, 178405.7)	(178274.9, 178409.1)	(178436.6, 178576.7)
Upper Bound	184617.3	184617.3	183710

Table 4.2: This shows 95% confidence intervals for the expected revenues under low load.

Therefore under high load the DLP method with cut propagation gave the highest estimate of expected revenue, then SDDP, followed by the standard DLP method. Under low load, SDDP performed best, then the DLP method with cut propagation and then the standard DLP method. Of additional note, SDDP produced upper bounds on the optimal expected revenues which were better than those obtained from the DLP.

Paired data t-tests were performed using the statistical package R [6] to find evidence of any differences between the expected revenues from each policy. Confidence intervals for the differences under high load are outlined in the following table:

	DLP	DLP with Cuts	SDDP
DLP	-	(84.21, 259.32)***	(-19.00, 159.12)
DLP with Cuts	-	-	(-190.80, -12.60)*
SDDP	-	-	-

Table 4.3: This shows 95% confidence intervals for the difference between expected revenues under high load. Differences correspond to the column heading minus the row heading. (***) = very strong evidence, (*) = evidence and () = no evidence of a difference.

For low load, we had the following confidence intervals:

	DLP	DLP with Cuts	SDDP
DLP	-	(-91.17, 100.07)	(71.38, 266.80)***
DLP with Cuts	-	-	(67.62, 261.66)***
SDDP	-	-	-

Table 4.4: This shows 95% confidence intervals for the difference between expected revenues under low load. Differences correspond to the column heading minus the row heading. (***) = very strong evidence and () = no evidence of a difference.

5 Conclusions

It was learned that given the complex nature of the airline network revenue management problem, SDDP was unsuitable for large problems despite performing well on small to medium ones, particularly under low load. This is because, in the small space of time available to airlines before the start of the booking horizon, the method is unable to build up an accurate representation of the value function as problem size increases and this function becomes more complex.

Methods which use the DLP, although sub-optimal, are efficient and easy to implement. Their widespread use in industry is a testament to this. The DLP approach is very versatile; we have learned as expected that the more frequently it is solved over the booking horizon, the higher expected revenues become. For single fare-class problems, the use of the DLP for creation and propagation of cuts has been shown to yield substantially greater expected

revenues than the standard bid-pricing policy. When a trigger function is added to this approach, optimizing the re-solve points, we obtain even better results. The results are less significant, however, when these two methods are applied to multiple fare-class problems. Nevertheless, there is still a significant improvement over the standard bid-price method.

In terms of implementation, airlines could set up a program to automatically compute bid-prices after every single customer arrival using all of the methods described here, thus utilising each method to its fullest potential. If such a system were not available and the airline could realistically only re-solve on several occasions, then the trigger would be of use to determine when best to perform any possible re-solves.

With the DLP cut propagation method, we may also compute many sets of cuts before the start of the booking horizon (as with SDDP), meaning that it would not be necessary to compute more cuts during the booking horizon, making implementation easier.

Whatever the implementation scheme an airline uses, the propagation of DLP cuts has been shown to generate the greatest expected revenues for medium to large problems.

It is also notable that SDDP and DLP cut propagation may be combined. This is an area of interest for future work.

6 Acknowledgments

I am indebted to Professor Andy Philpott, my supervisor, for his invaluable input to this project. Also much appreciated was the input of Professor Garrett van Ryzin of Columbia University. Finally, I wish to express my gratitude to the entire staff of the Department of Engineering Science for sharing their experience and energy.

7 References

- [1] J.F. Benders. Partitioning procedures for solving mixed variable programming problems. *Numerische Mathematik*, 4:238-252, 1962.
- [2] W.L. Cooper. Asymptotic behaviour of an allocation policy for revenue management. *Operations Research*, 50:720-727, 2002.
- [3] M.V.F. Pereira and L.M.V.G. Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52:359-375, 1991.
- [4] A.B. Philpott. [Personal communication]. 2006.
- [5] A.B. Philpott. Stochastic Programming, *ENGSCI 763 – Simulation and Stochastic Modelling*. The University of Auckland. September 2006.
- [6] R Development Core Team (2004). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.
- [7] K.T. Talluri and G.J. van Ryzin. *The Theory and Practice of Revenue Management*. Kluwer, New York, NY, 2004.
- [8] E.L. Williamson. *Airline Network Seat Inventory Control: Methodologies and Revenue Impacts*. PhD thesis, Flight Transportation Laboratory, MIT, Cambridge, MA, 1992.

Analysis of network strategic decisions for airlines

Amir Joshan
Department of Engineering Science
University of Auckland
New Zealand
a.joshan@auckland.ac.nz

Abstract

In this article we present a multi-period game where two airlines compete over a time horizon and try to optimize their revenue by setting their prices and releasing an appropriate quantity of seats in the market. We assume airlines have differentiated products and as a result each observes a unique demand function. We assume such demand functions can be affected by other competitor's prices for a similar product. We will discuss equilibrium issues for such a model and conditions under which it is strategically beneficial for an airline to expand its operating network.

Key words: Competitive pricing, Revenue management, Game theory, Perishable asset, Airline network structure.

1 Introduction

1.1 Motivation

Game theory is commonly used to model firms' competition in efficient markets. These kinds of models are particularly useful to derive pricing policies that is the most critical lever in firms' profitability and in particular revenue optimization competitions in airline markets. In this paper we use such a technique to show how strategic network decisions for airlines can improve the pricing policies and consecutively increase revenue.

In our model different airlines are operating over different networks however they may offer the same itineraries on their networks. We assume that each airline for each itinerary class has a different demand function. The idea that airlines have differentiated products is consistent with real market operations as airlines usually run frequent-flyer programs and try to make customer loyalty among passengers. However the airlines demand function is affected by other airlines and this prevents them from making their own monopoly market and enjoying a monopolist profit. This kind of competition is known as an oligopolistic market in economics literature. Since all the airlines simultaneously maximize their own revenue and they don't

cooperate with each other we consider a Nash equilibrium point (i.e. a set of strategies that no airline can get better off by changing just her own strategy).

We show that under certain conditions on demand functions, an airline can decide about the expansion of its current network, and enter a new market for an itinerary based on her current market information.

1.2 Related literature and contribution

Compared to the revenue management age as a research and practice field there's a huge literature on different aspects of this problem. Among all we refer to the most recent surveys. McGill and van Ryzin [6] present a comprehensive review of different issues in revenue management such as seat inventory control, cancellation and overbooking, and bid pricing models. Bitran and Caldentey [2] survey models of monopoly and pricing methods in the revenue management problem when products are perishable and resources are non-renewable and firms observe a stochastic price-sensitive demand process over a finite period of time. They categorized the results where demand is deterministic or non-deterministic; products are single or diverse, and whether models use a static or dynamic pricing method. Elmaghraby and Keskinocak [5] review the literature and current practices in dynamic pricing in industries where capacity or inventory is fixed in the short run and perishable. They classify monopolistic models on the basis of whether inventory can be replenished or not, whether demand is dependent over time or not, and whether customers are myopic or strategic optimizers.

On the competitive prospect, Vives [9] discusses oligopoly pricing models and its development. Cachon and Netessine [3] also survey the problem of competition and discuss both non-cooperative and cooperative games in static and dynamic settings.

Perakis and Sood[7] first developed a model characterized for oligopolistic competition on a single perishable product where all competitors have a fixed level of inventory. They have driven the conditions under which a unique equilibrium point could be reached in a market when demand for different firms has a degree of uncertainty using a robust optimization technique.

In this paper we are focusing on the same kind of model. However there exist a few differences that is as followed. First we characterized it for airline markets and extended it for the case of having multiple products as airlines offer several itineraries on their operating network. Our own contribution is connecting strategic decisions that usually are made prior to entering competition to the strategies which are followed during the ensuing pricing game.

1.3 Paper structure

The rest of this paper is structured as follows. In section 2, we describe the model and conditions in detail. In section 3 we describe market equilibrium and properties, and in section 5 we provide the connection between the market equilibrium point and network strategic decisions. In the last section we will review the paper contribution and draw conclusions.

2 Model formulation and conditions

In this section we describe the best response problem of an airline in a duopoly and our conditions. We assume that given the other competitor's policies each airline tries her best to optimize her revenue. We call this solving the best response problem. As a result, the market equilibrium is a set of policies that solve both of the best response problems simultaneously and as a result no one can get better off by changing its own strategy. This situation is known as a Nash Equilibrium in the game. We will return to conditions and proof of existence of such a set in section 3.

We assume that airline itineraries are differentiated products. This means that each airline has its particular demand for each itinerary that depends on prices offered for that itinerary by all the existing competitors on it. For the purpose of analysis we assume that demand functions are continuous. In our model all airlines are competing over a time horizon that contains T periods. Demand functions are assumed to be independent in each time period and could be different. Furthermore we are assuming that demand functions have the properties listed below:

1) Prices in any period vary between a minimum and maximum allowable level. We assume $p_{i \min}^{jt}$ to be strictly positive and $p_{i \max}^{jt}$ to be a level at which demand for airline i disappears irrespective of competitor prices in that period. We also assume that the allowable price interval for each itinerary is large enough that contains the unit elasticity point on the demand price function. We explain the definition of such a point in section 5. This condition makes sure that the space of feasible price strategies is bounded.

2) The amount of sale made by each airline in any period should be strictly positive. ie. $d_i^t > 0, \forall i, t$. This prevents airlines influencing the market by offering a price without actually selling any tickets for that itinerary.

3) The demand function $h_i^t(p_i^t, p_{-i}^t)$ is a concave and separable function of (p_i^t, p_{-i}^t) over the set of feasible prices, for all airlines and over all time periods. This property is to ensure that the feasible set of best response problems is convex. It implies that demand decreases more rapidly as price increases.

4) For any period t , for any fixed p_{-i}^t , the function $h_i^t(p_i^t, p_{-i}^t)$ is decreasing with respect to p_i^t and increasing with respect to p_{-i}^t over the set of feasible prices. This condition shows that different airlines are offering substitutable products and customers show a reaction to the prices that they observe.

2.1 Best response problem:

The best response policy for airline i is a policy that maximizes airline i 's revenue knowing the other airline's prices on the itineraries that she is acting over for the whole time horizon.

Airline multi-period pricing model: Consider that I , is set of airlines i (i.e. $I = \{1, 2\}$). Each airline has a network and a set of existing itineraries that may differ from the other's. Suppose J_i and L_i represent the set of itineraries and legs for airline i respectively. We will denote the number of existing legs and the number of existing itineraries for airline i by $|L_i|$ and $|J_i|$ respectively. Suppose c_i^l represents

the capacity of leg $l \in L_i$. Therefore each network is comprised of combinations of $|L_i|$ legs with capacities $C_i := (c_i^1, c_i^2, \dots, c_i^{|L_i|})$, and could be represented by an $|L_i| \times |J_i|$ matrix $A_i \equiv (n_i^{l,j})$. The entry $(n_i^{l,j}) \in \{0, 1\}$ indicates whether itinerary j uses leg l . The strategy of each seller is to optimally set her prices for different itineraries in each period of time. The demand observed by each seller in each period of time is equal to the number of buyers at that price level given by a unique demand price function $h_i^{jt}(p_i^{jt}, p_{-i}^{jt})$. It is obvious that this function also depends on the other airline price in that period p_{-i}^{jt} . Seller i will realize that demand only if she has enough seats available otherwise demand vanishes.

The best response problem of airline i given the other competitors' pricing policies over the whole time horizon is the solution of the following problem:

Indices:

i =Competitors, I is set of all the airlines.

j =Itineraries.

t =Time periods, T is set of all time periods on the time horizon.

l =Legs, L_i is set of all network legs of airline i .

Parameters:

c_i^l = capacity of leg l of airline i . In vector notation: $\mathbf{C}_i = (c_i^1, \dots, c_i^{|L_i|})^T$.

$d_i^{jt \min}$ = The least possible quantity that seller i can release in market for itinerary j in period t . In vector notation: $\mathbf{d}_i \min = (d_i^{11 \min}, \dots, d_i^{|J_i|1 \min}, \dots, d_i^{1|T| \min}, \dots, d_i^{|J_i||T| \min})^T$.

$p_i^{jt \min}$ = The least possible price that seller i can offer for itinerary j in period t . In vector notation: $\mathbf{p}_i \min = (p_i^{11 \min}, \dots, p_i^{|J_i|1 \min}, \dots, p_i^{1|T| \min}, \dots, p_i^{|J_i||T| \min})^T$.

$p_i^{jt \max}$ = The maximum allowable price that seller i can offer for itinerary j in period t . In vector notation: $\mathbf{p}_i \max = (p_i^{11 \max}, \dots, p_i^{|J_i|1 \max}, \dots, p_i^{1|T| \max}, \dots, p_i^{|J_i||T| \max})^T$.

$h_i^{jt}(p_i^{jt}, p_{-i}^{jt})$ =Demand price function for itinerary j in time period t for airline i .

It is strictly decreasing with respect to airline own price p_i^{jt} and strictly increasing with respect to other airlines' prices for that itinerary p_{-i}^{jt} . In vector notation:

$\mathbf{h}_i(\mathbf{p}_i, \mathbf{p}_{-i}) = (h_i^{11}(p_i^{11}, p_{-i}^{11}), \dots, h_i^{|J_i|1}(p_i^{|J_i|1}, p_{-i}^{|J_i|1}), \dots, h_i^{1|T|}(p_i^{1|T|}, p_{-i}^{1|T|}), \dots, h_i^{|J_i||T|}(p_i^{|J_i||T|}, p_{-i}^{|J_i||T|}))^T$.

Decision variables:

p_i^{jt} =Price of itinerary j at time period t for airline i . In vector notation:

$$\mathbf{p}_i^t = (p_i^{1t}, \dots, p_i^{|J_i|t})^T \text{ and } \mathbf{p}_i = (\mathbf{p}_i^1, \dots, \mathbf{p}_i^{|T|})^T.$$

d_i^{jt} =Number of seats to be sold for itinerary j at time period t for airline i . In vector notation: $\mathbf{d}_i^t = (d_i^{1t}, \dots, d_i^{|J_i|t})^T$ and $\mathbf{d}_i = (\mathbf{d}_i^1, \dots, \mathbf{d}_i^{|T|})^T$.

$$\begin{aligned} \max_{d_i^{jt}, p_i^{jt}} \quad & \sum_T \sum_J d_i^{jt} p_i^{jt} \\ \text{s.t.} \quad & d_i^{jt} \leq h_i^{jt}(p_i^{jt}, p_i^{-jt}) \quad \forall t \in T, \forall j \in J_i \\ & \sum_T A_i \mathbf{d}_i^t \leq \mathbf{C}_i \\ & p_i^{jt \min} \leq p_i^{jt} \leq p_i^{jt \max} \quad \forall t \in T, \forall j \in J_i \\ & d_i^{jt} \geq d_i^{jt \min} \quad \forall t \in T, \forall j \in J_i \end{aligned} \quad (1)$$

Explanation:

1. The objective is to maximize total revenue gained out of all itineraries over the entire time horizon.

2. At each price level for each itinerary in each time period we can not sell more than the demand that is observed at that price level.
 3. Quantity sold using a leg over the whole time horizon can not exceed the leg capacity.
 4. Price can vary between a minimum and a maximum allowable level for each itinerary in each time period.
 5. Quantity sold for each itinerary in each time period must be strictly positive.
- Let's assume that $\mathbf{z}_i = (\mathbf{d}_i, \mathbf{p}_i)$, then the above can be written in vector notation as:

$$\begin{aligned}
 \max_{\mathbf{z}_i=(\mathbf{d}_i, \mathbf{p}_i)} \quad & J_i(\mathbf{z}_i) = \frac{1}{2} \mathbf{z}_i^T \mathbf{Q}_i \mathbf{z}_i \\
 \text{s.t.} \quad & \\
 & \mathbf{d}_i \leq \mathbf{h}_i(\mathbf{p}_i, \mathbf{p}_{-i}) \\
 & \sum_T A_i \mathbf{d}_i^t \leq \mathbf{C}_i \\
 & \mathbf{p}_i \min \leq \mathbf{p}_i \leq \mathbf{p}_i \max \\
 & \mathbf{d}_i \geq \mathbf{d}_i \min
 \end{aligned} \tag{2}$$

Where $\mathbf{Q}_i = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{pmatrix}$ and \mathbf{I} denotes a suitable size square identity matrix for airline i .

The feasible set of the problem above is convex, and the best response policy of each airline only depends on other competitor's pricing policies rather than information on their capacity or network. In practice different airlines prices are almost always accessible via their website. However it is impossible to find out about their capacity and running network.

The problem above can be reformulated as a variational inequality problem as below. In the next section we will show how this variational inequality problem can be used to show the existence of the market equilibrium. Let us call the the feasible region of the problem above D_i and assume that $J_i = \mathbf{d}_i^T \mathbf{p}_i$. Then the corresponding variational inequality is searching for a pair of $(\mathbf{d}_i^*, \mathbf{p}_i^*)$ such that:

$$-\nabla J_i(\mathbf{d}_i^*, \mathbf{p}_i^*) \begin{pmatrix} \mathbf{d}_i - \mathbf{d}_i^* \\ \mathbf{p}_i - \mathbf{p}_i^* \end{pmatrix} \geq \mathbf{0} \quad \forall (\mathbf{d}_i, \mathbf{p}_i) \in D_i \tag{3}$$

It can be shown that a solution to the best response optimization problem is a solution to the variational inequality 3. It has been proved that the inverse is true for the case when there is only one itinerary (i.e. $|J_i = 1|$)[8]. However it can be shown that the conditions still hold when there is more than one itinerary. In the next section we will see how this is connected to the existence of market equilibrium.

3 Market Equilibrium

In this section we list the important lemmas from [8] that are used to show that a market equilibrium exists and it is unique. The market equilibrium problem can be defined as a joint quasi-variational inequality that is a combination of variational inequalities of each seller. Feasible space of the joint variational inequality is:

$$D = z = (z_1, z_2, \dots, z_{|I|}) | z_i \in D_i(z_{-i}), \forall i \in I$$

Then the joint quasi-variational inequality is searching for a pair of $(\mathbf{d}^*, \mathbf{p}^*)$ such that:

$$-\nabla F(\mathbf{d}^*, \mathbf{p}^*) \begin{pmatrix} \mathbf{d} - \mathbf{d}^* \\ \mathbf{p} - \mathbf{p}^* \end{pmatrix} \geq \mathbf{0} \quad \forall (\mathbf{d}, \mathbf{p}) \in \mathbf{D} \tag{4}$$

Where $F_i(z^*) = -\nabla J_i(z^*), \forall i \in I$.

Proposition3.1 : Solution from solving the variational inequality 3 and the best response problem 1 are the same.

Proof: Since the conditions that are cited in [8] hold, the solution to the best response problem 1 is a solution to the variational inequality 3 and vice versa.

Proposition3.2 : Solution gained from the joint quasi variational inequality 4 versus solution gained from solving variational inequality problems for each seller all simultaneously, will give the same answers.

Proof: Refer to [8].

Proposition3.3 : The joint quasi variational inequality has a unique solution.

Proof: Since the feasible set is nonempty and compact, and the objective function is continuous refer to [1] there exist a solution to this variational inequality.

Proposition3.4 : Solution gained from joint quasi variational inequality is also a Nash Equilibrium.

Proof: Refer to [8].

Hence there exists a unique Nash equilibrium for the market model.

4 Analysis of Network Strategic Decisions

In the previous sections we have seen under certain conditions for price demand functions, that equilibrium prices exist. However there exist some decisions that are usually made before airlines face competitive markets. How to decide on a new itinerary or a route is often a problem that airlines face. In this section we will connect this problem to the competitive market equilibrium point. In particular we look at conditions under which competitors can make sure that by offering a new product (i.e. a new itinerary) they dont loose any revenue that they can obtain in the market. To do so we use the demand elasticity concept.

Definition 4.1-Price elasticity of demand: Price elasticity of demand is a dimensionless measure of the responsiveness of the quantity demanded of a good, to a change in its price, when all other influences on buyers' plan remain the same. We can calculate it by using the formula:

$$\text{Price elasticity of demand (at a particular price)} = \frac{\text{Percentage Change in Quantity Demanded}}{\text{Percentage Change in Price}}$$

In this paper by 'price elasticity' we refer to 'price elasticity of demand'.

Definition 4.2-Definition of elastic demand: If percentage change in quantity demanded exceeds percentage change in price, demand is elastic and price elasticity is greater than 1.

Definition 4.3-Definition of inelastic demand: If percentage change in quantity demanded is less than percentage change in price demand is inelastic and price elasticity is less than 1.

Definition 4.4-Definition of unit elastic demand: If the percentage change in the quantity demanded equals the percentage change in price then price elasticity equals 1 and it is called unit elastic.

Lemma 4.1-Total Revenue Test: We express the revenue test for two states of demand price function as below:

- a) When demand is elastic a decrease in price will result in more total revenue.
- b) When demand is inelastic a decrease in price will result in less total revenue.

Proof: First we consider latter case (b). The argument for part (a) is similar. Total revenue is equal to price multiplied by the quantity sold. Demand is inelastic so we have:

$$\frac{\Delta d/d_{ave}}{\Delta p/p_{ave}} \leq 1 \Rightarrow d_{ave} \cdot \Delta p \geq p_{ave} \cdot \Delta d$$

Total revenue before price cutting is $p_0 \cdot d_0 = (p_{ave} + \Delta p/2)(d_{ave} + \Delta d/2)$ which is greater than revenue after price cutting that is $(p_0 - \Delta p)(d_0 + \Delta d) = (p_{ave} - \Delta p/2)(d_{ave} + \Delta d/2)$ due to the inequality above. Therefore a price cut for inelastic demand worsens the total revenue.

Lemma 4.2-Elasticity of demand in equilibrium: At equilibrium, demand constraints for each airline hold as equality and demand elasticity is equal to or greater than 1 for all itineraries in each time period.

Proof: Consider the first argument. Demand constraints can not hold unequally, since if quantity sold is greater than demand then demand constraint is violated and if it is less than demand, then price can be increased to the point that equation holds tight. In that case airline can get better off which is in contradiction with the equilibrium definition. Second argument is hold because of the total revenue test lemma. Suppose demand is inelastic (demand elasticity is less than 1) for an itinerary in equilibrium. Then according to the total revenue test lemma, player can decrease the quantity sold and increase revenue by increasing the price for that itinerary. Hence again the equilibrium definition is violated.

Proposition 4.1-Positive slopes for best response price curves: At the equilibrium point, the best response curves of each airline in each period, have a

positive slope for each itinerary.

Proof: Suppose that competitors are playing equilibrium pricing policies. We want to show that:

$$\frac{dp_j^{it}}{dp_j^{-it}} \geq 0$$

for this purpose we will use the KKT condition of the best response problem of airline i and we will show that the expression above holds at equilibrium. We have seen in lemma 4.2 demand constraints always hold at equality so we can rewrite the the best response problem (1) as:

$$\begin{aligned} \max \quad & Z(P_i) = \sum_T \sum_J p_i^{jt} h_i^{jt}(p_i^{jt}, p_{-i}^{jt}) \\ \text{s.t.} \quad & G_i^L(P_i) = \sum_T A_i \mathbf{d}_i^t \leq \mathbf{C}_i \\ & p_i^{jt \min} \leq p_i^{jt} \leq p_i^{jt \max} \quad \forall t \in T, \forall j \in J^i \\ & G_i^J(P_i) = h_i^{jt}(p_i^{jt}, p_{-i}^{jt}) \geq d_i^{jt \min} \quad \forall t \in T, \forall j \in J_i \end{aligned} \quad (5)$$

Note that the second constraint never binds because of Lemma 4.2 and the first condition on $p_i^{jt \min}$. Suppose that J_i^L and J_i^J are Jacobians of the functions $G_i^L : \mathfrak{R}^{J^i|T|} \rightarrow \mathfrak{R}^1$ and $G_i^J : \mathfrak{R}^{|T|} \rightarrow \mathfrak{R}^1$ respectively then we have:

$$\begin{aligned} J_i^L(P_i) &= \left[\frac{\partial G_i^L}{\partial p_i^{jt}} \right]_{(t=1, \dots, |T|, j=1, \dots, |J^i|, t=1, \dots, |T|)} \\ J_i^J(P_i) &= \left[\frac{\partial G_i^J}{\partial p_i^{jt}} \right]_{((j=1, \dots, |J^i|, t=1, \dots, |T|), 1)} \end{aligned}$$

Therefore the KKT conditions for the above problem are:

$$\begin{aligned} -\nabla Z(P_i) + J_i^L(P_i)^T \lambda^L + J_i^J(P_i)^T \lambda^J &= 0 \\ G_i^L(P_i) + y^L &= C_i \\ -G_i^J(P_i) + y^J &= -d_i^{J \min} \\ \lambda^L y^L &= 0 \\ \lambda^J y^J &= 0 \\ (\lambda^L, \lambda^J, y^L, y^J) &\geq 0 \end{aligned} \quad (6)$$

Let us consider the price of itinerary j in time period t at equilibrium. Considering the first equation in 6 we have:

$$\begin{aligned} -(h(p_i^{*jt}, p_{-i}^{*jt}) + p_i^{*jt} \frac{\partial h(p_i^{*jt}, p_{-i}^{*jt})}{\partial p_i^{*jt}}) + (\sum_{(l|n_i^{l,j} \neq 0)} \lambda^{*l} - \lambda^{*jt}) \frac{\partial h(p_i^{*jt}, p_{-i}^{*jt})}{\partial p_i^{*jt}} &= 0 \\ \Rightarrow (p_i^{*jt} - \sum_{(l|n_i^{l,j} \neq 0)} \lambda^{*l} + \lambda^{*jt}) \frac{\partial h(p_i^{*jt}, p_{-i}^{*jt})}{\partial p_i^{*jt}} + h(p_i^{*jt}, p_{-i}^{*jt}) &= 0 \end{aligned}$$

Note that the latter also shows that at equilibrium price is elastic as we have:

$$p_i^{*jt} \frac{\partial h(p_i^{*jt}, p_{-i}^{*jt})}{\partial p_i^{*jt}} - h(p_i^{*jt}, p_{-i}^{*jt}) \leq 0 \text{ since } \sum_{(l|n_i^{l,j} \neq 0)} \lambda_i^* \geq 0$$

We can calculate $\frac{dp_i^{*jt}}{dp_{-i}^{*jt}}$ by implicitly differentiating the equation above:

$$\frac{dp_i^{*jt}}{dp_{-i}^{*jt}} = - \frac{\frac{\partial h(p_i^{*jt}, p_{-i}^{*jt})}{\partial p_{-i}^{*jt}} + (p_i^{*jt} - \sum_{(l|n_i^{l,j} \neq 0)} \lambda_l^* + \lambda_j^*) \frac{\partial^2 h(p_i^{*jt}, p_{-i}^{*jt})}{\partial p_{-i}^{*jt} \partial p_i^{*jt}} - \sum_{(l|n_i^{l,j} \neq 0)} \frac{\partial \lambda_l^*}{\partial p_{-i}^{*jt}} \frac{\partial h(p_i^{*jt}, p_{-i}^{*jt})}{\partial p_i^{*jt}}}{2 \frac{\partial h(p_i^{*jt}, p_{-i}^{*jt})}{\partial p_i^{*jt}} + (p_i^{*jt} - \sum_{(l|n_i^{l,j} \neq 0)} \lambda_l^* + \lambda_j^*) \frac{\partial^2 h(p_i^{*jt}, p_{-i}^{*jt})}{\partial p_i^{2*jt}} - \sum_{(l|n_i^{l,j} \neq 0)} \frac{\partial \lambda_l^*}{\partial p_i^{*jt}} \frac{\partial h(p_i^{*jt}, p_{-i}^{*jt})}{\partial p_i^{*jt}}}$$

This is always positive since $\frac{\partial^2 h(p_i^{*jt}, p_{-i}^{*jt})}{\partial p_i^{2*jt}} = 0$.

Proposition 4.2-Existance of a congested leg at equilibrium: There exist a congested leg at equilibrium for all airlines that are offering at least one elastic itinerary in one period of time horizon.

Proof: Consider the best response problem of airline i and suppose $L_i^{congested}$ be the set of congested legs for that airline. we want to show that:

$$if \ p_i^{jt} \frac{\partial h(p_i^{*jt}, p_{-i}^{*jt})}{\partial p_i^{*jt}} \leq h(p_i^{*jt}, p_{-i}^{*jt}) \text{ then } L_i^{congested} \neq \phi$$

It's easy to show that $L_i^{jt} = \{l|n_i^{l,j} \neq 0\}$ is a nonempty set (otherwise airline i can get better off by offering more of itinerary j at time period t which is inconsistent with definition of Nash Equilibrium). So we have: $L_i^{jt} \subseteq L_i \Rightarrow L_i^{congested} \neq \phi$.

Proposition 4.3-: Let's L_i be the set of legs that airline i is operating over. Adding a new itinerary that uses $L_i^{new}|L_i^{congested} \subset L_i^{new}$ doesn't decrease best response prices of airline i for previous itineraries.

Proof: By adding a new itinerary on the current network that uses all the congested legs, lagrange multipliers of those constraints increase [1]. Since $\sum_{(l|n_i^{l,j} \neq 0)} \lambda_l^l = p_i^{*jt} + \frac{h(p_i^{*jt}, p_{-i}^{*jt})}{\partial h(p_i^{*jt}, p_{-i}^{*jt})/\partial p_i^{*jt}}$ must hold at optimal point, p_i^{*jt} must increase for all j and t because $p_i^{*jt} + \frac{h(p_i^{*jt}, p_{-i}^{*jt})}{\partial h(p_i^{*jt}, p_{-i}^{*jt})/\partial p_i^{*jt}}$ is increasing with respect to p_i^{*jt} on the elastic part of demand function $h(p_i^{*jt}, p_{-i}^{*jt})$.

Proposition 4.4: If adding a new itinerary that uses set of legs $L_i^{new}|L_i^{congested} \subset L_i^{new}$ is beneficiary with respect to the current competitors' strategies, adding that itinerary will be beneficiary at new equilibrium point.

Proof: According to **Proposition 4.3** adding such an itinerary doesn't decrease other itineraries best response strategies. In fact it increases price of those itineraries that are using congested legs since \mathbf{d}_i^{new} is strictly positive (constraint 4 in best response formulation). as we have shown in **Proposition 4.1** that best response curves have positive slopes, new equilibrium set of prices are higher than the previous set of prices.

5 Conclusion

In this paper we have reviewed a duopoly in which two airlines are competing on several itineraries over a time horizon. Their set of itineraries are not necessarily the same. We have shown that for such a game there exist a unique nash Equilibrium

set of strategies. We have also shown the conditions that ensure an airline will be better off after adding a new itinerary to her operating network.

References

- [1] M. Bazaraa, H. Sherali, C. Shetty *Nonlinear Programming, 2 edn.* New York: John Wiley Sons, 1993
- [2] G. Bitran, R. Caldentey, *An overview of pricing models for revenue management*, Manufacturing and Service Operations Management, 2002
- [3] G. Cachon, S. Netessine, *Game theory in supply chain analysis*, Supply Chain Analysis in E-business era, Kluwer Academic Publishers, 2004
- [4] L. Chen and T. Homem-de-Mello, *Re-Solving Stochastic Programming Models for Airline Revenue Management*, Annals of Operations Research, To appear.
- [5] W. Elmaghraby, P. Keskinocak *Dynamic pricing in the presence of inventory considerations: Research overview, current practices and future directions*, Management Science 49, 2003
- [6] J.I. McGill, G.J van Ryzin, *Revenue management: Research overview and prospects*, Transportation Science, 1999
- [7] G. Perakis, A. Sood, *Competitive Multi-period Pricing for Perishable Products: A Robust Optimization Approach*, Math Programming, January 6, 2006
- [8] G. Perakis, E. Adida, *Dynamic Pricing and Inventory Control: Uncertainty and Competition Part A: Existence of a Nash Equilibrium*, Submitted to Math Programming, January 2006
- [9] X. Vives, *Oligopoly Pricing - Old Ideas and New Tools*, MIT Press, 1999

A Comparison of Solution Strategies for Biobjective Shortest Path Problems

Andrea Raith
Department of Engineering Science
University of Auckland
New Zealand
a.raith@auckland.ac.nz

Abstract

Biobjective shortest path (BSP) problems arise in various applications. Since obtaining the set of efficient solutions to a BSP problem is more difficult (i.e. \mathcal{NP} -hard and intractable) than solving the corresponding single objective problem there is a need for fast solution techniques. Our aim is to compare different strategies for solving the BSP problem. We consider a standard label correcting method, a purely enumerative near shortest path approach, and the two phase method, investigating different approaches to solving problems arising in phase 1 and phase 2. In particular, we propose to combine the two phase method with ranking in phase 2. In order to compare the different approaches, we investigate their performance on three different types of networks. We employ grid networks and random networks, as is generally done in the literature. Furthermore, road networks are utilised to compare performance on networks with a structure that is more likely to actually arise in applications.

1 Introduction

When studying shortest path problems, it is often not sufficient to restrict oneself to one objective. Applications often indicate the necessity of taking two or more objectives into account, resulting in biobjective or multiple objective shortest path problems. Examples include transportation problems (Pallottino and Scutellà 1998), routing in railway networks (Müller-Hannemann and Weihe 2006), and problems in satellite scheduling (Gabrel and Vanderpooten 2002).

BSP belongs to the class of multiple objective combinatorial optimisation (MOCO) problems. Our objective is to find efficient solutions of a BSP problem. BSP is an \mathcal{NP} -hard problem and it also is intractable, i.e. the number of efficient solutions may be exponential in the number of nodes. Supported efficient solutions are efficient solutions whose image is situated on the boundary of the convex hull of Z , the image of the feasible set. All other efficient solutions are called non-supported, they are situated in the interior of $\text{conv}(Z)$. Non-supported solutions can only be obtained by enumeration.

In the literature the main approaches to solving BSP problems are enumerative approaches: Ranking (Clímaco and Martins 1982) and label correcting (Skriver and Andersen 2000; Brumbaugh-Smith and Shier 1989) or label setting (Martins 1984; Tung and Chew 1988). Clímaco and Martins (1982) employ a k -shortest path method for ranking. We investigate ranking with a near shortest path method by Carlyle and Wood (2005).

The two phase method is an approach taking advantage of the problem structure (Mote, Murthy, and Olson 1991; Ulungu and Teghem 1995). In phase 1, extreme supported solutions are computed. In phase 2 the remaining efficient solutions are computed using an enumerative approach, which can be employed efficiently as the search area can be restricted by results of phase 1.

2 Biobjective Shortest Path Problems

Let $G = (N, A)$ be a *directed network* with a set of nodes $N = \{1, \dots, n\}$ and a set of arcs $A = \{(i_1, j_1), \dots, (i_m, j_m)\} \subseteq N \times N$. Positive costs $c_{ij} = (c_{ij}^1, c_{ij}^2) \in \mathbb{N} \times \mathbb{N}$ are associated with each arc $(i, j) \in A$. A *path* in G from node $i_0 \in N$ to node $i_l \in N$ is a sequence $\{(i_0, i_1), (i_1, i_2), \dots, (i_{l-1}, i_l)\}$ of arcs in A . We study the *BSP problem*, seeking shortest paths from a *source* $s \in N$ to a *target* $t \in N$. We furthermore define the *biobjective shortest path tree (BSPT) problem*, a formulation better suited when using a network simplex algorithm to solve single criterion shortest path problems:

$$\min \quad z(x) = \begin{cases} z_1(x) = \sum_{(i,j) \in A} c_{ij}^1 x_{ij} \\ z_2(x) = \sum_{(i,j) \in A} c_{ij}^2 x_{ij} \end{cases} \quad (1)$$

$$\text{s.t.} \quad \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} n-1 & \text{if } i = s \\ -1 & \text{if } i \neq s \end{cases} \quad (2)$$

$$x_{ij} \geq 0 \text{ and integer, } \forall (i, j) \in A. \quad (3)$$

The model ensures that one unit of flow is transported through the network from s to each other node i , corresponding to paths from source node s to all other nodes. The *feasible set* X is described by constraints (2) and (3) and its image is $Z := z(X)$.

We are seeking those paths, that do not allow the improvement of one component of the objective vector $z(x)$ without deteriorating the other one.

Definition 1 A feasible solution $\hat{x} \in X$ is called *efficient* if there exists no $x' \in X$ with $(z_1(x'), z_2(x')) \leq (z_1(\hat{x}), z_2(\hat{x}))$ and $(z_1(x'), z_2(x')) \neq (z_1(\hat{x}), z_2(\hat{x}))$, $z(\hat{x}) = (z_1(\hat{x}), z_2(\hat{x}))$ is called *non-dominated*. Let X_E denote the set of efficient solutions and $Z_N := z(X_E)$. We distinguish two different types of efficient solutions.

Efficient solutions which are optimal solutions to a weighted sum problem

$$\min_{x \in X} \lambda^1 z_1(x) + \lambda^2 z_2(x) \text{ for some } \lambda^1 > 0, \lambda^2 > 0 \quad (4)$$

are called *supported*. The set of all supported efficient solutions is denoted by X_{SE} , and $Z_{SN} := z(X_{SE})$. The set Z_{SN} lies on the boundary of $\text{conv}(Z)$. The remaining efficient solutions in $X_{NE} := X_E \setminus X_{SE}$ are called *non-supported efficient solutions*, and $Z_{NN} := z(X_{NE})$. They lie in the interior of $\text{conv}(Z)$.

Definition 2 Two feasible solutions x and x' are called *equivalent* if $z(x) = z(x')$. A complete set X_E is a set of efficient solutions such that all $x \in X \setminus X_E$ are either dominated or equivalent to at least one $x \in X_E$.

We will only consider solution approaches that compute a complete set X_E . It remains to define *lexicographic minimisation*.

Definition 3 Let $k \in \{1, 2\}$ and $l \in \{1, 2\} \setminus \{k\}$. Then $z(\hat{x}) <_{lex(k,l)} z(x')$ if either $z_k(\hat{x}) < z_k(x')$ or both $z_k(\hat{x}) = z_k(x')$ and $z_l(\hat{x}) < z_l(x')$. We call \hat{x} a *lex(k, l)-best solution*. Let $x_{lex(k,l)}$ denote a *lex(k, l)-best solution*.

3 Literature

The most recent survey on BSP problems is by Skriver (2000). Both surveys on MOCO problems by Ehrgott and Gandibleux (2000) and (2002) contain shortest path problems. We mention recent literature on exact methods here.

Martins and Dos Santos (2000) prove boundedness and finiteness results for the multi-objective shortest path problem with loops and discuss complexity analysis. They present generic labelling algorithms, and an algorithm for acyclic networks.

Guerrero and Musmanno (2001) investigate different strategies of label correcting and label setting methods for the multi-objective shortest path tree problem. Computational results are presented for two different classes of test problems.

Gabrel and Vanderpooten (2002) apply a multiple objective shortest path label setting procedure to the daily scheduling of earth observing satellites, which they formulate as a shortest path problem with three objectives on acyclic networks. They present an interactive procedure to select one of the enumerated paths.

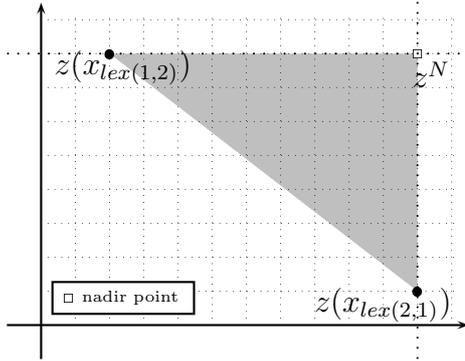
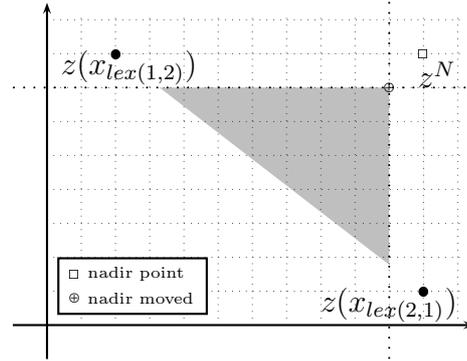
Sastry, Janakiraman, and Mohideen (2003) propose several algorithms for multi-objective shortest path problems with positive and negative arc costs. They describe how to detect negative circuits. Furthermore they present a label correcting algorithm for multi-objective shortest path problems similar to the one by Brumbaugh-Smith and Shier (1989) and two other ones similar to Corley and Moon (1985).

Müller-Hannemann and Weihe (2006) investigate the cardinality of the set of efficient solutions that arises in practical applications. They examine the characteristics of shortest path problems in train networks with two and three objectives. They relate network and problem characteristics to the actual number of efficient solutions. They find that this number is very low despite the fact that shortest path problems are intractable in general.

4 Label Correcting

A biobjective label correcting method is a straight forward extension of the single objective version. For two or more objectives there may be several labels at a node, each corresponding to one path. The labels at one node are all non-dominated among each other. We follow the approach identified as most successful approach to solving BSP problems by Skriver and Andersen (2000).

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) . 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, this operation is called *merging*. After removing dominated labels at 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,

Figure 1: Bounds on z_1 and z_2 .Figure 2: Improved bounds on z_1 and z_2 .

the mark of the node is deleted. When no nodes are marked for reconsideration any more, the algorithm terminates. Merging is the most expensive component of a biobjective label correcting algorithm.

Once the label correcting algorithm terminates, the set $Labels(t)$ contains all non-dominated path costs at the target node t . The corresponding efficient solutions (the paths) can be obtained by backtracking the appropriate labels.

5 Ranking – Near Shortest Path (NSP)

Solving the BSP problem with a k -shortest path method, means repeatedly computing paths with increasing length. Starting with the optimal value for one objective, the second best solution, then the third best one etc. is obtained until it is guaranteed that all non-dominated solutions have been found. As solving the BSP problem with k -shortest path methods is not competitive with label correcting methods (Huang, Pulat, and Shih 1996; Skriver 2000), we use the NSP method by Carlyle and Wood (2005), which aims at finding all paths whose length is within a certain deviation ϵ from the optimal path length ω , thus having a maximal path length of $\delta = \omega + \epsilon$.

In order to use the NSP procedure, a weighted sum problem (4) corresponding to BSP is considered. Weighting factors $\lambda^1 > 0$ and $\lambda^2 > 0$ depending on the $lex(1, 2)$ - and $lex(2, 1)$ -best solutions obtained in an initialisation phase are defined:

$$\lambda^1 = z_2(x_{lex(1,2)}) - z_2(x_{lex(2,1)}) \text{ and } \lambda^2 = z_1(x_{lex(2,1)}) - z_1(x_{lex(1,2)}) \quad (5)$$

Upper bounds originating from the two lexicographically best solutions $x_{lex(1,2)}$ and $x_{lex(2,1)}$ can be used to restrict enumeration (Figure 1), they can be further improved by the fact that we are dealing with integer problems as indicated in Figure 2. For every candidate solution \hat{x} with $z(\hat{x}) = (z_1(\hat{x}), z_2(\hat{x}))$ we get the bounds:

$$z_1(\hat{x}) \leq z_2(x_{lex(2,1)}) - 1 \text{ and } z_2(\hat{x}) \leq z_1(x_{lex(1,2)}) - 1. \quad (6)$$

We call $z^N = (z_1(x_{lex(2,1)}), z_2(x_{lex(1,2)}))$ the *nadir point* of the BSP problem.

A cost $c_{ij}^\lambda > 0$ is associated with each arc (i, j) , where $c_{ij}^\lambda = \lambda^1 c_{ij}^1 + \lambda^2 c_{ij}^2$. The maximum path length is $\delta = \lambda^1(z_1(x_{lex(2,1)}) - 1) + \lambda^2(z_2(x_{lex(1,2)}) - 1)$, the weighted sum value of the nadir point shifted one unit down and to the left.

The NSP algorithm repeatedly computes candidate solutions within the bounds (6) and with length $< \delta$. Only after the algorithm terminates, we know that the

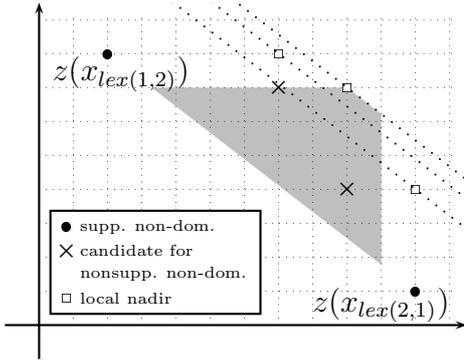


Figure 3: Weighted sum bounds.

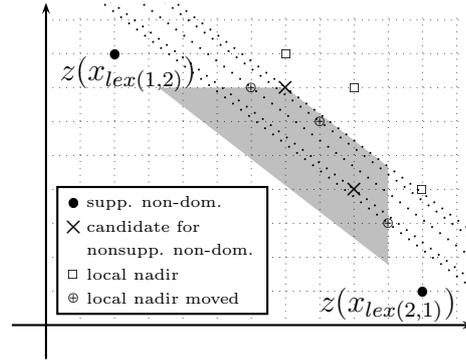


Figure 4: Improved sum bounds.

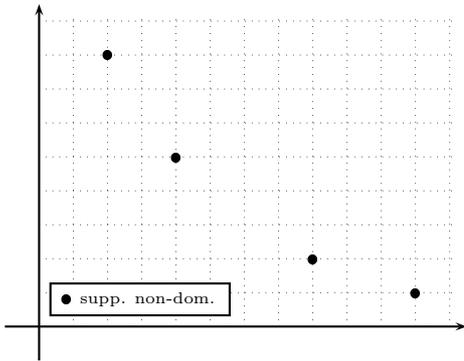


Figure 5: Supported non-dominated.

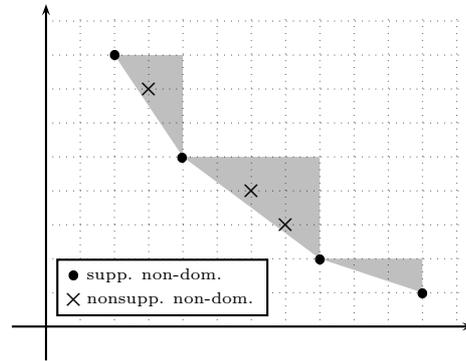


Figure 6: All non-dominated points.

remaining candidate solutions are indeed non-dominated. It is however possible to exploit candidate solutions to improve the upper bound δ by taking advantage of the fact that every computed candidate excludes a certain area by domination.

The *local nadir point* of two points $z^k = (z_1^k, z_2^k)$ and $z^l = (z_1^l, z_2^l)$ with $z_1^k < z_1^l$ and $z_2^k > z_2^l$ is $z^{LN} = (z_1^l, z_2^k)$. We consider straight lines parallel to the one connecting $z(x_{lex(1,2)})$ and $z(x_{lex(2,1)})$ through the local nadir point of any two neighbouring candidate solutions, $z(x_{lex(1,2)})$, and $z(x_{lex(2,1)})$ as indicated in Figure 3. The upper bound corresponds to the line with maximal distance from the line connecting $z(x_{lex(1,2)})$ and $z(x_{lex(2,1)})$. As we are dealing with integer problems, this upper bound can be improved like bound (6) was, see Figure 4. The value of δ is updated whenever a new candidate solution is computed.

6 Two Phase Method

In phase 1 of the two phase method (Mote, Murthy, and Olson 1991; Ulungu and Teghem 1995) extreme supported efficient solutions are computed, see Figure 5. In phase 2 the remaining efficient solutions are computed using an enumerative approach. The search space in phase 2 can be restricted to triangles given by two neighbouring supported non-dominated solutions as indicated in Figure 6. An initialisation phase is needed to compute one or two initial solutions.

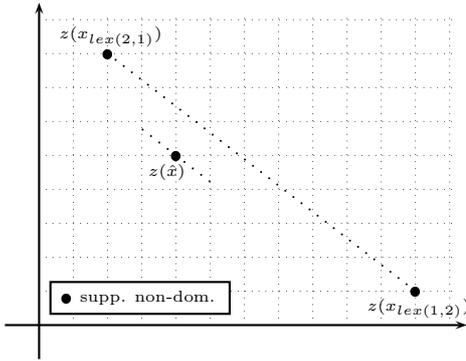


Figure 7: Dichotomic, first iteration.

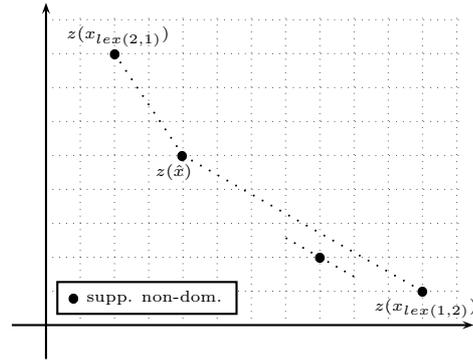


Figure 8: Dichotomic, second iteration.

6.1 Initialisation

We need to compute a $lex(1,2)$ -best or $lex(2,1)$ -best solution or both depending on the approach chosen in phase 1. Here, single objective shortest path problems are solved with appropriate objective functions. We investigate the following options:

- (LC) Single objective label correcting algorithm.
- (D) Single objective label setting algorithm: Dijkstra's algorithm.
- (S) Network simplex. An initial artificial solution for (BSPT) is constructed. Multiple partial pricing is used to speed up the selection of basic entering arcs.

6.2 Phase 1

We investigate two different approaches in phase 1. On the one hand we use a dichotomic approach. Weights are chosen to obtain a supported efficient solution that has the maximal distance to the straight line connecting the two initial solutions $z(x_{lex(1,2)})$ and $z(x_{lex(2,1)})$ as illustrated in Figure 7. The obtained efficient solution \hat{x} leads to two new weighted sum problems: one between $z(x_{lex(1,2)})$ and $z(\hat{x})$, and one between $z(\hat{x})$ and $z(x_{lex(2,1)})$, see Figure 8. Whenever a problem is solved at most two new subproblems are derived from it. The dichotomic method iterates until all weighted sum problems and arising subproblems are solved, a complete set of the extreme supported solutions is obtained.

On the other hand we use a parametric network simplex approach. Starting off with a $lex(1,2)$ -best solution, the supported efficient set X_{SE} is explored from the upper left to the lower right. The goal is to reach a $lex(2,1)$ -best solution of the problem, and basic entering arcs are chosen so that every supported efficient solution is computed on the way to the $lex(2,1)$ -best solution. All non basic arcs have to be considered when choosing a basic entering arc, as an arc with minimal ratio between improvement of z_2 and the deterioration of z_1 has to be chosen. Thus partial pricing can not be used to speed up the simplex algorithm. We employ the following solution strategies to solve the single objective problems arising here:

- (SDIC) Network simplex dichotomic: Use network simplex algorithm.
- (LDIC) Label correcting dichotomic: Use label correcting.
- (SPAR) Network simplex parametric: Use network simplex algorithm.

Table 1: Grid network test problems.

	w×h	$ Z_N $		w×h	$ Z_N $		w×h	$ Z_N $		w×h	$ Z_N $
G1	20 × 80	80	G9	200 × 100	132	G17	288 × 17	15	G25	53 × 92	93
G2	50 × 90	124	G10	200 × 150	204	G18	196 × 25	18	G26	44 × 111	137
G3	90 × 50	46	G11	50 × 50	52	G19	140 × 35	32	G27	35 × 140	209
G4	50 × 200	290	G12	100 × 100	113	G20	111 × 44	54	G28	25 × 196	244
G5	200 × 50	12	G13	200 × 200	309	G21	92 × 53	53	G29	17 × 288	371
G6	100 × 150	149	G14	2450 × 2	6	G22	79 × 62	77	G30	8 × 612	819
G7	150 × 100	122	G15	1225 × 4	6	G23	70 × 70	93	G31	4 × 1225	1383
G8	100 × 200	247	G16	612 × 8	10	G24	62 × 79	95	G32	2 × 2450	1594

6.3 Phase 2

In phase 2 it is possible to benefit from results from phase 1 to significantly reduce runtime of the enumerative methods used. Let z^1, \dots, z^k , where $z^i = (z_1(x^i), z_2(x^i))$ and z^i are sorted by increasing z_1 , be (at least) the extreme points of a complete supported efficient set obtained in phase 1. Non-supported non-dominated solutions can only be situated in the triangle defined by two neighbouring supported non-dominated solutions, see Figure 6. For every pair of neighbouring solutions x^i and x^{i+1} with $i \in \{1, \dots, k-1\}$ an enumerative approach is used to compute non-supported solutions.

- (NSP) Near Shortest Path for every pair of neighbouring solutions with bounds as in Section 5, paths can often be discarded early during computation.
- (LCOR) Biobjective Label Correcting from Section 4 for all considered triangles at the same time, and discard labels once they are not in any of the triangles.

7 Test Networks

In order to investigate the performance of the different solution methods, we introduce three types of networks and then present computational results.

Problem instances of *grid networks* are listed in Table 1. Nodes are arranged in a rectangular grid with given height and width. Every node has at most four outgoing arcs (up, down, left and right), to its immediate neighbours. Source node s and target node t are connected to nodes on the left and right margin of the grid, respectively. Arc costs are chosen randomly with $c_{ij}^k \in \{1, 2, \dots, 10\}$, $k = 1, 2$.

Skriver and Andersen (2000) propose *NetMaker networks*, instances are given in Table 7. A random Hamiltonian cycle ensures connectedness of the network. A random number of arcs out of every node is generated, in between a minimum and maximum number of outgoing arcs. The arcs can only reach a certain number of nodes forward and backward, specified by the node interval. Arc costs are determined randomly either $c_{ij}^1 \in \{1, 2, \dots, 33\}$ and $c_{ij}^2 \in \{67, 68, \dots, 100\}$ or the other way round. We investigate three modifications to the structure of *NetMaker*:

- for all arcs in the cycle, set c_{ij}^k , $k = 1, 2$ randomly in $\{1, 2, \dots, 10000\}$.
- half of the arcs out of a node go to nodes with higher node numbers. Arc weights like in a) for arcs in the cycle, all others: $c_{ij}^k \in \{1, 2, \dots, 10\}$, $k = 1, 2$.
- everything as in b), for all arcs in the cycle set $c_{ij}^k = 10000$, $k = 1, 2$.

We use *road networks* (Schultes 2005) to test our methods on real world data, instances are listed in Table 3. We duplicate arcs to obtain a directed network and

Table 2: NetMaker network test problems.

	nodes	node interval	outgoing arcs: min-max	Var a) $ Z_N $	Var b) $ Z_N $	Var c) $ Z_N $
NM1-2	3000	20	5-15 / 1-20	6 / 8	1 / 1	3 / 4
NM3-5	3000	50	5-15 / 1-20 / 10-40	9 / 15 / 6	2 / 3 / 3	2 / 4 / 3
NM6-7	7000	20	5-15 / 1-20	6 / 5	1 / 3	2 / 3
NM8-10	7000	50	5-15 / 1-20 / 10-40	3 / 7 / 6	2 / 1 / 6	3 / 3 / 4
NM11-12	14000	20	5-15 / 1-20	15 / 20	2 / 4	2 / 1
NM13-15	14000	50	5-15 / 1-20 / 10-40	2 / 17 / 7	2 / 2 / 3	1 / 4 / 3
NM16-17	21000	20	5-15 / 1-20	5 / 4	3 / 1	3 / 1
NM18-20	21000	50	5-15 / 1-20 / 10-40	7 / 4 / 5	3 / 2 / 3	1 / 1 / 3

Table 3: Road network test problems.

	state	nodes	arcs	$ Z_N $: average	min	max
DC1-DC9	Washington DC	9559	39377	3.33	1	7
RI1-RI9	Rhode Island	53658	192084	9.44	2	22
NJ1-NJ9	New Jersey	330386	1202458	10.44	2	21

add a cycle with high arc costs to ensure connectedness of the network. Arc costs are the time needed to travel the arc and the travel distance in meters. For each road network we test nine instances with randomly chosen source and target nodes.

8 Results

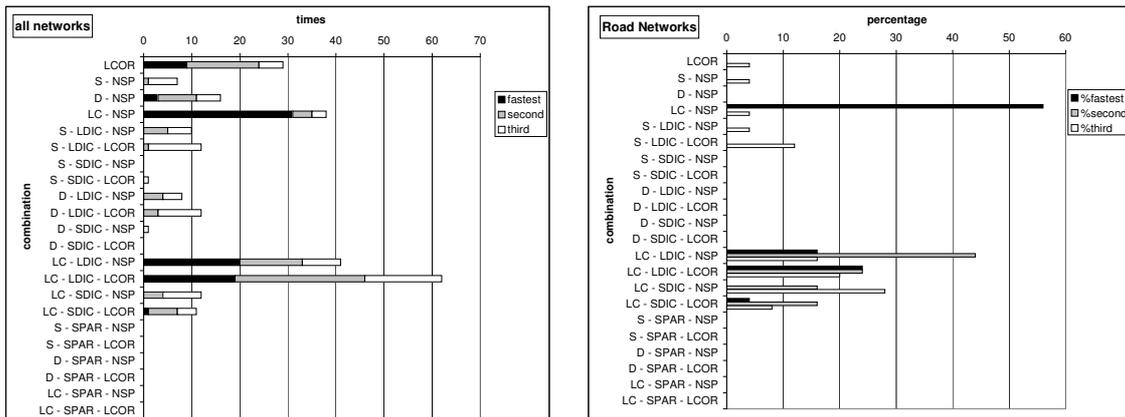


Figure 9: All - fastest three algorithms. Figure 10: Road Networks - fastest three.

The solution methods presented in Sections 4 - 6 lead to a total of 22 different approaches. We test LCOR, NSP with initialisation L/D/S and the two phase method with initialisation L/D/S, phase 1 SDIC/LDIC/SPAR, and phase 2 LCOR/NSP.

Numerical test are performed on a Linux (Fedora Core 4) desktop computer with 3GHz Pentium 4 processor and 1GB RAM. We use the gcc compiler (version 4.0.2) with compile option -O3. The methods are implemented in C, we adapt program code from Carlyle and Wood (2005) for NSP and LC. The network simplex is a modification of MCF (Löbel 2003). Runtime is measured disregarding the time it takes to read instances from file. All efficient paths and their labels are generated. Computations are aborted when their runtime exceeds 30 minutes. If an instance has only one efficient solution, this is detected in initialisation by the two phase method with dichotomic phase 1 or the NSP method, we exclude these instances from further considerations.

The fastest approaches for *all* test problem are displayed in Figure 9. The most successful combinations are LC - LDIC - NSP/LCOR, followed by LC - NSP and

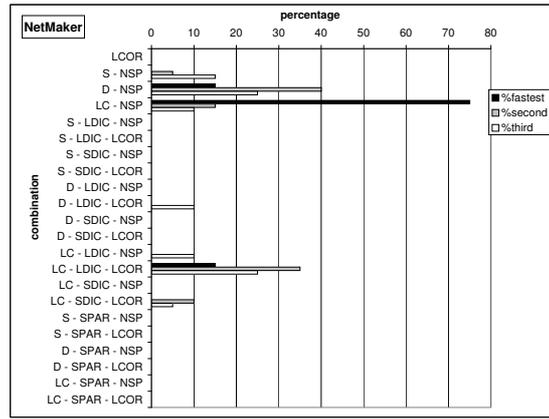
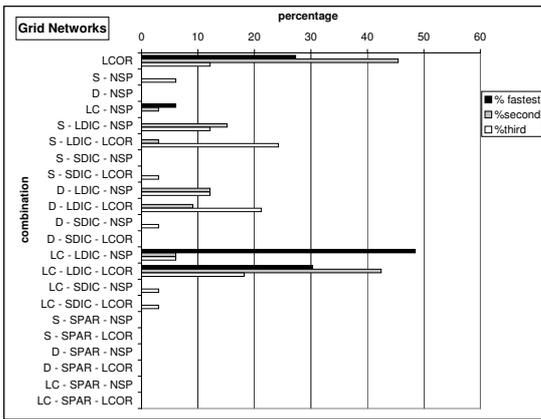


Figure 11: Grid Networks - fastest three. Figure 12: NetMaker a) - fastest three.

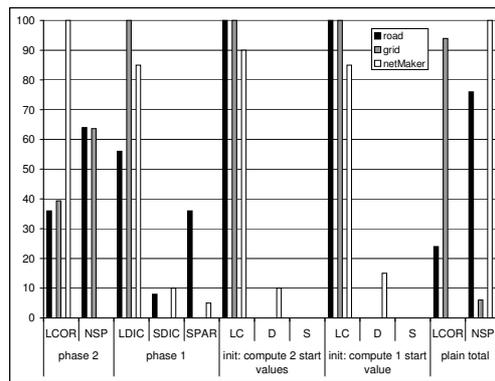


Figure 13: Fastest approaches in different phases.

LCOR. We can discard some approaches: Initialisation with S or D is not very successful and SPAR in phase 1 of the two phase method does not perform well.

It is, however, worthwhile considering each network type individually. In Figures 10 - 12 we display in how many percent of all computational tests of a network type, a combination is fastest (or second/third fastest). For road networks we identify the NSP algorithm with LC initialisation as quickest approach. Also, the two phase method with LC - LDIC - NSP/LCOR is quite successful, employing SDIC in phase 1 is only slightly worse. Performance on grid networks is very interesting. The NSP algorithm performs very badly here, it is aborted 28 times out of 32. LCOR performs quite well on grid networks. However, considering the two phase method LC - LDIC - NSP is the best approach. This demonstrates well the benefit of the two phase method: NSP performs a lot better when used in phase 2 than it does by itself. We can observe the opposite behaviour for NetMaker networks. The combination LC - NSP is very successful, whereas LCOR does not appear among the fastest three at all. But the fastest two phase approach is the combination LC - LDIC - LCOR.

We identify the best approaches to be chosen for each phase, again distinguishing the different network types, see Figure 13. Whatever the network type, LC performs best in initialisation. The results for phase 1 indicate that LDIC is the best approach to employ. The preferred approach in phase 2 depends on the network type, LCOR is better suited for grid networks, NSP for road and NetMaker networks. For the two purely enumerative approaches behaviour is just the opposite.

Variations in arc costs may lead to drastically different results. The investigated

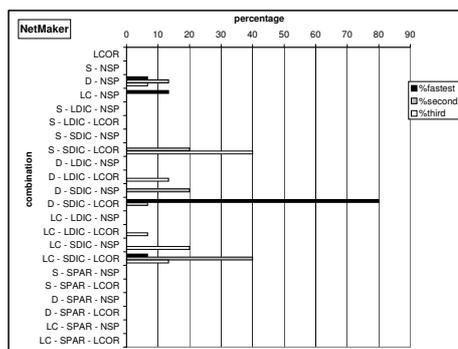


Figure 14: NetMaker Var b)

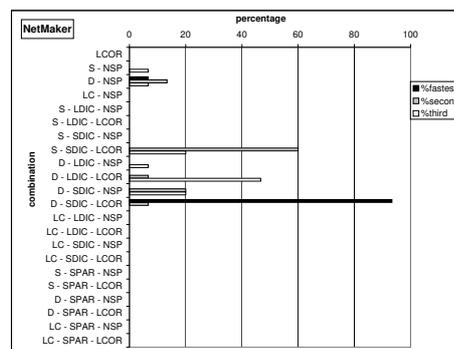


Figure 15: NetMaker Var c)

variations of *NetMaker* have the same parameters, yet computational results are very different, see Figures 12, 14, and 15. Variations b) and c) are the only examples where single objective label *setting* (D) was successful. In variations b) and c) the two phase method performs a lot better than in variation a).

9 Conclusion

The two phase method is competitive with other commonly applied approaches to solve BSP problems. The two phase method works well with both ranking and label correcting in phase 2. The purely enumerative NSP approach was aborted on several instances, but performed well on others. It becomes clear that it depends a lot on the network type which approach performs best, even small variations on the network may have a high impact on performance.

References

- Brumbaugh-Smith, J., and D. Shier. 1989. "An empirical investigation of some shortest path algorithms." *European Journal of Operational Research* 43 (2): 216–224.
- Carlyle, W.M., and R.K. Wood. 2005. "Near-Shortest and K-Shortest Paths." *Networks* 46 (2): 98–109.
- Clímaco, J.C.N., and E.Q.V. Martins. 1982. "A bicriterion shortest path problem." *European Journal of Operational Research* 11:399–404.
- 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.
- Ehrgott, M., and X. Gandibleux. 2000. "A survey and annotated bibliography of multiobjective combinatorial optimization." *OR Spektrum* 22:425–460.
- . 2002. "Multiobjective Combinatorial Optimization – Theory, Methodology, and Applications." In *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*, edited by M. Ehrgott and X. Gandibleux, International Series in Operations Research & Management Science, 369–444. Kluwer.
- Gabrel, V., and D. Vanderpooten. 2002. "Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an earth observing satellite." *European Journal of Operational Research* 139 (2): 533–542.

- 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.
- Huang, F., S. Pulat, and L.-H. Shih. 1996. "A Computational Comparison of some Bicriterion Shortest Path Algorithms." *The Chinese Institute of Industrial Engineers. Journal* 13 (2): 121–125.
- Löbel, A. 2003. MCF, Version 1.3. <http://www.zib.de/Optimization/Software/Mcf/>.
- 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.E. Dos Santos. 2000. "The labelling algorithm for the multiobjective shortest path problem." Technical Report, Universidade de Coimbra, Portugal, Departamento de Matemática.
- Mote, J., I. Murthy, and D. L. Olson. 1991. "A parametric approach to solving bicriterion shortest path problems." *European Journal of Operational Research* 53:81–92.
- Müller-Hannemann, M., and K. Weihe. 2006. "On the Cardinality of the Pareto Set in Bicriteria Shortest Path Problems." *Annals of Operations Research* 147:269–286.
- Pallottino, S., and M.G. Scutellà. 1998. "Shortest path algorithms in transportation models: classical and innovative aspects." In *Equilibrium and Advanced Transportation Modelling*, edited by P. Marcotte and S. Nguyen, 245–281. Kluwer Academic Publishers.
- Sastry, V.N., T.N. Janakiraman, and S.I. Mohideen. 2003. "New Algorithms For Multi Objective Shortest Path Problem." *Opsearch* 40 (4): 278–298.
- Schultes, assembled by D. 2005. Tiger Road Networks for 9th DIMACS Implementation Challenge – Shortest Path. online. From <http://www.dis.uniroma1.it/~challenge9/data/tiger/>.
- Skriver, A.J.V. 2000. "A Classification of Bicriterion Shortest Path (BSP) Algorithms." *Asia-Pacific Journal of Operational Research* 17:199–212.
- 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.
- Tung, C.T., and K.L. Chew. 1988. "A Bicriterion Pareto-optimal path algorithm." *Asia-Pacific Journal of Operations Research* 5:166–172.
- Ulungu, E. L., and J. Teghem. 1995. "The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems." *Foundations of Computing and Decision Sciences* 20 (2): 149–165.

Faster Shortest Path algorithms for Siren

Peter Ebden
Supervised by Dr Andrew Mason

Abstract

A major drawback to standard implementations of Dijkstra's algorithm is that shortest paths are calculated in all directions from the start node of the search. An approach that has been developed recently is to preprocess the network by associating a geometric container with each arc. These containers encompass all of the destinations that have a shortest path beginning with that arc. This paper will focus on the implementation of these containers in the software package Siren, which was originally created in the University of Auckland and is now being developed by The Optima Corporation. This implementation reduces shortest path computation times to around 30%. A new approach to constructing the containers is discussed, which increases preprocessing speed by approximately 3x. Finally, approaches to complications in Siren such as multiple representations of arc length are compared.

1 Introduction

Siren Predict is a software package created in the University of Auckland and now being developed by The Optima Corporation to simulate emergency vehicle deployments to aid optimisation of base locations. In order to run these simulations, it must calculate many shortest paths, an operation which can be computationally expensive.

One of the major drawbacks to Dijkstra's algorithm, which Siren is based on, is that it searches in all directions from its start node. In order to assure optimality of its solution it scans all nodes in order of increasing distance. Hence all nodes that are closer to the source than the destination is must be examined, even if they lie in a direction that is not useful to the search. It would be desirable to be able to direct the algorithm so it searches only in directions that are likely to be useful.

2 Geometric Containers

There have been several papers explaining the use of geometric containers to speed up shortest path searches. Wagner & Willhalm (2003) covers this concept as well as testing to show the advantages of this approach. It is noted that many repeated shortest path queries are made on an unchanged network, which justifies expensive preprocessing calculations to accelerate later searches. The networks considered are typically large, so approaches with quadratic space requirements such as precalculating all of the shortest paths in the network are not possible.

Geometric containers are proposed as a means of decreasing the search times of Dijkstra's algorithm. For each arc in the network, a geometric container can be constructed. A container is a geometric figure (typically a rectangle) that encloses, and hence defines, a subset of nodes. Our containers are the smallest rectangles encompassing all of the destinations that have a shortest path beginning with the arc they are associated with. A container is defined to encompass a node (or any point) if that point lies within or on the boundaries of the area of the container.

When running Dijkstra's algorithm, any arc whose container does not intersect the

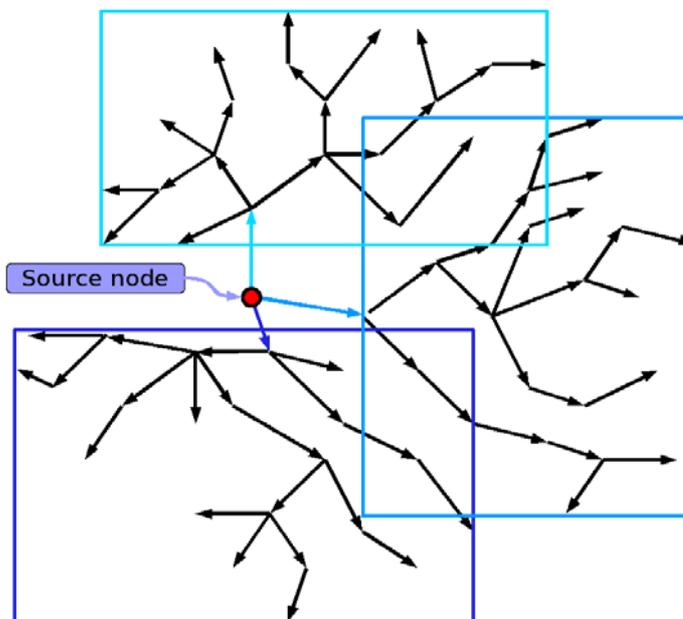


Figure 1: Consistent geometric containers for one node

destination of the search can be discarded rather than being added to the list of nodes to scan. These containers are referred to as being "consistent" since they do not exclude any nodes from the Dijkstra search that are on the shortest path from source to destination.

Figure 1 illustrates a set of consistent geometric containers for one node.

3 Creating Containers

The algorithm to generate consistent containers is very similar to Dijkstra's algorithm and is given in Wagner et al (2003). It is reproduced here as Algorithm 1.

This algorithm runs Dijkstra's algorithm from all the nodes in the network in succession. Since no end node is given, Dijkstra's algorithm terminates once all

nodes have been scanned. For the purposes of building the containers, it is necessary to know the source arc of each node. This is the first arc on the shortest path to that node from the root node. For example, in Figure 2, the green arc is the source arc for all of the arcs marked in red.

Once a complete shortest path tree has been built, the algorithm then walks through the list of nodes. For each node, it expands the container associated with the node's source arc to cover the node. When this has been completed, the current root node is referred to as being processed, and the algorithm moves to the next root node.

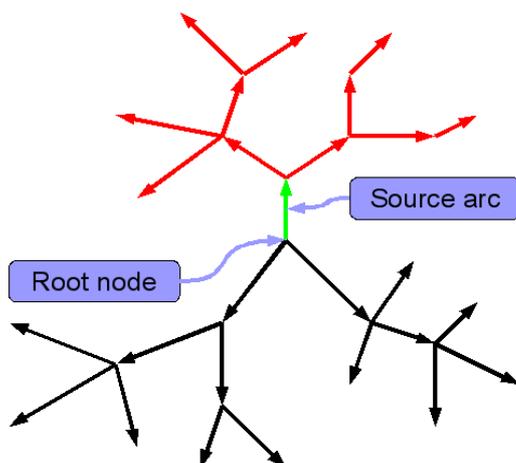


Figure 2: Source Arcs

Algorithm 1: Create-Containers

```

// initialise a complete set of containers
for all a in E do
  create new container C(a) to cover H(a)
for all s in V do
  // run Dijkstra's algorithm from node s
  set d(u) = ∞ for all u in V
  create empty priority queue Q
  insert s in Q and set d(s) = 0
  while Q is not empty
    get node u with smallest d(u) in Q
    for all arcs a in DecisionForwardStar(u) do
      set v = H(a)
      set new-dist = d(u) + l(u,v)
      if new-dist < d(v)
        if d(v) = ∞
          insert v in Q with priority new-dist
        else
          set priority of v in Q to new-dist
          set d(v) = new-dist
      // set the source arc of this node
      if u = s
        set A(v) = (s, v)
      else
        set A(v) = A(u)
  // enlarge the containers associated
  // with the arcs leaving s
  for all nodes y in V \ {s}
    enlarge C(A(y)) to contain y
  
```

Note: Additions to Dijkstra's algorithm are shown in bold print. This algorithm runs in $O(n^2 \log n)$ operations, since Dijkstra's algorithm runs in $O(n \log n)$ operations for sparse graphs, where n is the number of nodes in the network.

4 Reusing the Shortest Path Tree during construction

Previous literature has described the preprocessing algorithm as expensive, but they do not elaborate on typical computation times. Our calculation in Siren proved to be considerably slower than was desirable, and this would be amplified on larger networks in the future. Since the size and time required for these networks was not known, it was necessary to improve the running time of the algorithm as much as possible on existing networks.

4.1 Method

The various Create-Container algorithms all create a large number of shortest path trees. However, on each repeated call to the algorithm they discard all previous information and re-solve for a complete tree at a new start node (ie. new root node for the tree). This takes a considerable length of time and does not re-use any of the information gained in previous iterations.

However, the structure of the tree generally remains mostly unchanged between adjacent root nodes. Hence there are considerable savings to be gained if this is exploited by moving the root of the tree to a neighbour of its current root. This is done by decreasing the distance label on the new root node by the distance from the new root to the old one. Dijkstra's algorithm is then run from the new source starting with distance labels set by the last iteration (except for the decremented new root node distance). This is shown in Figure 3.

In general when moving to a new root in this way, a large portion of the tree will remain unchanged as shorter paths to it are not found. This is the green area in Figure 3. The part of the tree which the root moves towards will have to be entirely rebuilt, even though its structure may not change – this is the red area in the diagram. It is necessary to rebuild this part of the tree in order to update the distance labels on those nodes. Also, a shorter path may be found to some nodes on the edge of the green area. These nodes will move to a different part of the tree as illustrated.

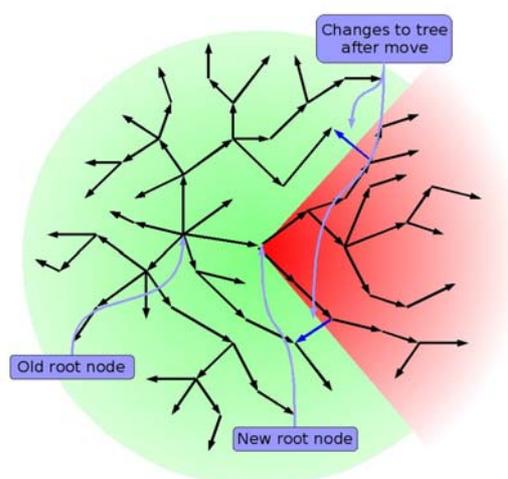


Figure 3: Moving the shortest path tree to a new root

The exact amount of the tree which is required to be rebuilt depends on the structure of the network on question, and in particular the average degree (the number of arcs intersecting at a node) of the network.

4.2 Choosing moves efficiently

In order to gain the maximum benefit of reusing the constructed shortest path tree, it would be helpful to select the best neighbouring node to move to rather than picking one randomly. In this case the best node would be the one that leads to the least change of the shortest path tree.

As well as length, arcs in Siren have a speed associated with them. This represents the average speed vehicles can travel along that arc. Our conjecture was that moving along arcs with faster speeds would lead to the least reconstruction of the tree. In general, most shortest paths starting from a node on a motorway or other fast road would tend to move along that road for some distance before leaving it, hence moving a few nodes up or down

that road will have little impact on the shape of the shortest path tree. This effect would be particularly pronounced for motorways as they only have offramps periodically, so for large numbers of decision nodes there would be effectively no change in the structure of the tree. Therefore we choose to construct successive trees by moving the root node along the fastest arcs available.

Since we are attempting to accelerate the container construction, we do not wish to spend long searching for the best node to use as our new root. This choice is not required to be optimal, so we use a simple heuristic to quickly search a few nodes near our current root.

We inspect all the arcs leaving the current root, and select the fastest arc amongst those leading to a node that has not yet been processed. If all those nodes have been processed, then we move along the fastest arc and search again from that node. This process is repeated until a valid new root is found, or until the number of steps taken reaches some maximum amount. It is necessary to limit the number of steps in this way as the process of moving the tree becomes less efficient as more steps are taken, and once the majority of nodes have been processed it becomes difficult to find an unprocessed node in this way.

4.3 Algorithms

The preprocessing algorithm was altered to add the ability to reuse the tree. Because a large part of the tree is not altered, many of the arcs will be labelled with source arcs coming from the old root node. Their source arc is hence updated to be the source arc of the old root node, which is set during the **ChooseNewRootNode** algorithm.

If a new source node to move the tree to cannot be found (because all the nearby nodes to the old source have already been processed), then the

Algorithm 2: PreprocessReusingTree

```
// initialisation
CalculateReverseDecisionArcs()
for all  $a$  in  $E$  do
  create new container  $C(a)$  to cover  $H(a)$ 
 $s = 0$ 
// begin by building a complete tree
BuildShortestPathTree( $s$ )
while ( $s \neq -1$ )
  for all  $v$  in  $V \setminus \{s\}$ 
    enlarge  $C(A_s(v))$  to contain  $v$ 
  BuiltContainers( $s$ ) = true
  ChooseNewRootNode( $s$ )
  if  $s \neq -1$ 
    // move the tree to the new root
    BuildShortestPathTree( $s$ )
  for all  $v$  in  $V \setminus \{s\}$ 
    if  $T(A_s(v)) \neq s$ 
       $A_s(v) = A_s(T(A_s(v)))$ 
    else
      //search for an unprocessed node to
      //build a clean tree
      for all  $s$  in  $V$ 
        if !BuiltContainers( $s$ )
          BuildShortestPathTree( $s$ )
          break
```

Algorithm 3: ChooseNewRootNode

```
set  $CurrentNode = s$ 
set  $StartDistance = 0$ 
set  $old\_s = s$ 
set  $s = -1$ 
for  $i = 0$  to  $MaxNoSteps$ 
  set  $IncumbentNode = CurrentNode$ 
  // look for a node to move to
  for  $x$  in DecisionForwardStar( $CurrentNode$ )
    set  $NewNode = H(x)$ 
    if  $P_s(NewNode) = x$  and  $x \neq OldArc$  and
       $x.ReverseArc \neq -1$ 
      // compare node to our incumbent
      if BuiltContainers( $NewNode$ )
        if BuiltContainers( $IncumbentNode$ )
          set  $IncumbentNode = NewNode$ 
          set  $IncumbentSpeed = x.Speed$ 
          set  $IncumbentArc = x$ 
        else if  $x.Speed > IncumbentSpeed$ 
          set  $IncumbentNode = NewNode$ 
          set  $IncumbentSpeed = x.Speed$ 
          set  $IncumbentArc = x$ 
      else
        if BuiltContainers( $IncumbentNode$ ) and
           $x.Speed > IncumbentSpeed$ 
          set  $IncumbentNode = NewNode$ 
          set  $IncumbentSpeed = x.Speed$ 
          set  $IncumbentArc = x$ 
  if not BuiltContainers( $IncumbentNode$ )
    // the new node hasn't been processed
    set  $StartDistance += l(IncumbentArc.ReverseArc)$ 
    set  $s = IncumbentNode$ 
    set  $d(s) = d(old\_s) - StartDistance$ 
    set  $A_s(old\_s) = P_s(IncumbentNode).ReverseArc$ 
    break
  else
    if  $IncumbentNode = CurrentNode$ 
      // no new node, terminate
      break
    else
      // the new node has been processed
      // move to it and search again
      set  $OldArc = IncumbentArc.ReverseArc$ 
      set  $StartDistance += l(OldArc)$ 
      set  $CurrentNode = IncumbentNode$ 
```

current tree is abandoned and the list of nodes is scanned to find one that has not yet been processed. The preprocessing algorithm with these alterations is shown here as Algorithms 2 and 3.

A limitation of this algorithm is that we can only move the root node along arcs that are present in the shortest path tree, but since we only store the parent arc of the nodes in the tree, we must simply inspect all nodes and discard them if their parent does not come from the node we are scanning from. This is somewhat inefficient, but not significant to the running time of the algorithm since comparatively few steps are taken in this search before either a new source is found or the existing tree is abandoned.

It was also necessary to modify **BuildShortestPathTree** as Siren uses unsigned integers for distance labels on nodes. However, once the tree is moved, the distance label on the new root must be negative since it is less than that on the original root. It was decided that the best solution would be to mark the root node with an arbitrarily large distance (2^{30}) when rebuilding the entire tree. Converting to signed integers would require a significant amount of recoding and would slow the initial operation to set all distance labels to infinity, as the built-in memset function cannot be used to set signed integers to their maximum value.

4.5 Tests and benchmarking

The PreprocessReusingTree algorithm was tested by building sets of containers for the Melbourne, Tiny Perth and Perth networks. Results are shown in Table 1.

Table 1: Container construction times

Network	Melbourne (2000 nodes)	Perth CBD (3,814 nodes)	Perth (50,000 nodes)
Construction time without reusing tree	3:38	7:25	9:47:25
Construction time reusing tree	1:23	2:34	3:21:02
Speedup	2.62x	2.9x	2.9x

The acceleration of this algorithm by reusing the shortest path tree appears to remain fairly constant with network size. This was expected as the number of operations remains approximately proportional to the size of the network, since that proportion of the network that the root moves closer to must be entirely rescanned.

5 Dijkstra's algorithm with Pruning

Once a set of containers had been constructed, the modifications to Dijkstra's algorithm were then carried out to use the containers to accelerate the search.

In order to test and compare the shortest path code, many random pairs of nodes were generated and shortest paths between them were calculated. This allowed calculation of

Algorithm 4: Dijkstra's Algorithm with Pruning

```

set  $d(u) = \infty$  for all  $u$  in  $V$ 
create empty priority queue  $Q$ 
insert  $s$  in  $Q$  and set  $d(s) = 0$ 
while target  $t$  is not marked finished and
   $Q$  is not empty
  get node  $u$  with smallest  $d(u)$  in  $Q$  and
    mark it finished
  for all nodes  $v$  in DecisionForwardStar( $u$ ) do
    if  $C(u,v)$  contains  $t$ 
      set new-dist =  $d(u) + l(u,v)$ 
      if new-dist <  $d(v)$ 
        if  $d(v) = \infty$ 
          insert  $v$  in  $Q$  with priority new-dist
        else
          set priority of  $v$  in  $Q$  to new-dist
          set  $d(v) = \text{new-dist}$ 

```

Note: Addition to Dijkstra's algorithm shown in bold print. Algorithm taken from Wagner and Wilhelm (2003).

the average time required to compute a shortest path with and without containers.

The code was compared on the Melbourne, Tiny Perth and Perth networks to see the difference between network sizes. 20,000 random pairs of nodes were determined for each and the results timed and averaged.

Table 2: Comparison of average shortest path calculation times

Network	Melbourne	Tiny Perth	Perth
Average time without containers (ms)	2.22	3.22	10.86
Average time with containers (ms)	0.78	0.98	2.42
Percentage of time required with containers	35.1%	30.4%	22.3%

The results show a significant speedup in search time when the containers are being used. This acceleration ranges from just under 3x for a small network like Melbourne to nearly 5x for the large Perth network. This is as we expected, as the effect of containers is more pronounced over longer paths, and the average path length in the full Perth network is considerably longer than in the others.

6 Multiple Container Sets

6.1 Travel Priorities

Siren uses multiple different travel representations of “distance” in its simulation. Typically there are three stored for most networks:

- Distance
- Travel time
 - Normal travel speed
 - Lights and sirens speed

These different travel priorities have completely independent arc distances. Because they do not share any data we must construct and store a separate set of containers for each priority. This increases the calculation time and storage requirements by a factor of 3.

6.2 Travel Times

Siren scales arc lengths according to the time of day and day of the week. Arc lengths are calculated as weighted averages of three underlying distances:

- Morning peak
- Afternoon peak
- Off peak

Each day is split into 48 half-hour time slots, so there are 336 of these in a week, each with its own weights and hence its own set of arc “distances”. Clearly the storage requirements for storing separate containers for each of these times rapidly become intractable for a network of any significant size, so other approaches must be investigated.

6.3 Approaches to multiple travel times

Two approaches to this problem were considered. The first was to store sets of containers corresponding to the key travel times. We considered that there would be four of these sets required to accurately and conveniently match the arc scalings. These would be morning peak, midday off peak, afternoon peak and off peak. These container sets are easily created by running the algorithm to create containers for each travel time, and storing different sets of containers for each time. The times themselves must also be stored in order to determine which sets of containers are appropriate when the containers are being used.

It seemed reasonable to assume that for an arc to be useful in a shortest path, the destination node will lie inside either the container associated with that arc for the previous time, or the container for the next time period, or both. When running Dijkstra's algorithm, both containers would have to be considered, and the node included in the search if it fell into either of them.

An alternative approach is to maintain only one set of containers, but expand them to cover the shortest paths for all time periods. This is done simply by running the algorithm to create containers for multiple times and priorities. This has the advantage of smaller storage requirements by a factor of at least 4 from the multiple sets approach. Also those storage requirements remain constant regardless of how many time periods are required. The code required was also slightly simpler, as it was not necessary to store and load the times that the containers were generated for.

Both of these approaches were implemented, and a comparison was made, based on solving 20,000 random shortest paths on the Tiny Perth network:

Table 3: Comparison of shortest path calculation times

	Multiple container sets	Single container set
Average time to solve a random shortest path (ms)	1.08	1.05

While the single set approach was slightly faster, the difference in solution time between the two approaches was almost negligible. While the multiple set approach is more efficient in the sense that it excludes more nodes from the Dijkstra search than the single set approach, it is slightly slower computationally due to the need to do two container comparisons for each node considered.

Since the multiple set approach proved to have no advantage in practice, it was decided to store only the single set of containers.

6.4 Maintaining solution optimality

Some consideration was given to the question of whether either approach to the travel times would retain optimality in our solutions. The arc lengths are a linear combination of the key arc distances, so it might appear that the optimal solution between two time points must be the optimal solution at one of those time points. However this is not the case, as we now show.

Consider a problem where three different paths are found between a pair of nodes. The first path is the shortest when arc lengths are computed for one time, but the longest for a second time. The second path is the longest for the first time, but the shortest for the second, and the third path falls in between at both times. If we interpolate linearly to

determine path lengths for times between these two, there may be times when the third path is the shortest. This is illustrated by Figure 4.

It can be seen that while the time taken on the optimal path between two nodes may vary linearly, the shortest path at all times in between Time 1 and Time 2 is not necessarily the shortest path at either of those times.

While this is a contrived example, we were able to observe it in Siren by selecting a series of random pairs of nodes to calculate shortest paths for, and comparing the results both with and without containers. The effect could however be mitigated partially by expanding the containers by considering more travel times, or completely by considering all travel times. A number of experiments were conducted in which 5,000 random pairs of nodes were chosen. Shortest paths were found between them using containers and compared to the shortest paths found without containers to see how many produced different results. This procedure was repeated for several different sets of containers calculated for different numbers of travel times.

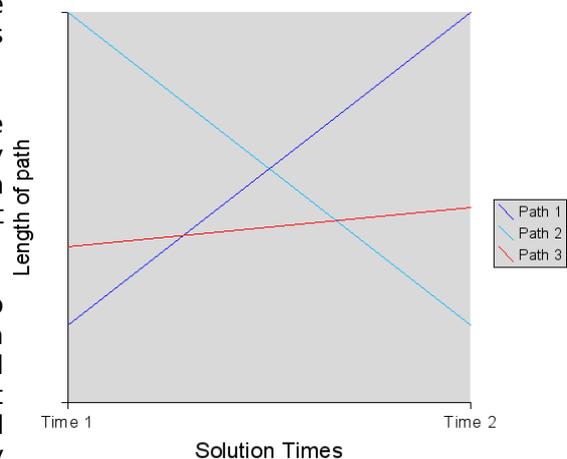


Figure 4: Heuristic solution approach

Table 4: Effect of number of travel times considered

No. of times	3	21	336
Incorrect solutions found	148 (2.9%)	20 (0.4%)	0
Average error	17 (1.1%)	1 (0.06%)	0

As expected, the addition of extra travel times improved the heuristic error. This improvement came at the expense of a significant increase in preprocessing time, by a factor of 7. However, this increase in the number of times considered did not entirely eliminate error. The consideration of all 336 time indices served as a control since it eliminated the heuristic error entirely, but these containers took a significant length of time to calculate. The time taken is proportional to the number of times calculated for, so while this took some hours for the Melbourne network, for the Perth network it could be expected to take approximately 140 days.

However the figure of 336 could be reduced considerably since it is only necessary to consider time indices with unique weightings. The weightings remain constant overnight, and in between the morning and afternoon peaks, as well as frequently from one day to the next. In these cases each unique weighting need only be calculated once, so the figure of 336 could be reduced significantly if the times used are chosen carefully. Around 30 times would be sufficient to generate a consistent set of containers for the Melbourne network.

Even this figure is too high for a large network such as Perth. Calculation of containers for 30 times could still take as long as 12 days. However, due to the nature of the algorithm, it would be straightforward to split the calculation across multiple computers since each time is calculated separately. If a network of 20 computers were available, it would reduce computation time to approximately 16 hours. Currently the travel times to be calculated for are hard-coded into Siren, but the code could be modified so these are selected at run-time.

It would also be necessary to combine all containers once they are calculated. This is a simple addition, as the containers are stored as a pair of locations defining opposite corners

of the rectangle. Loading a set of containers for a different time would consist of reading the locations representing the new set of containers, and expanding the existing set to encompass these where required, in the same way that nodes are added when running the algorithm to compute a set of containers.

7 Simulation and Results

Initially the Melbourne network was the only network that simulation data was available for. The simulation covered one month with calls based on historical data. Table 5 shows the shortest path computation time required to complete simulation runs on this network both with and without containers.

Table 5: Time spent on shortest path calculations for Melbourne

Without containers	With containers
25:52	10:15

This showed that the shortest path computation time within the simulation run had been cut to 39% of what it was without containers.

Once the Melbourne simulation had been carried out, it was of interest to see what impact the containers would have on computation time of a larger network. At this point Optima kindly made available a full Melbourne network that was the largest we had access to. Table 6 shows the time required to complete a simulation on this network with and without containers.

Table 6: Time spent on shortest path calculations for Full Melbourne

Without containers	With containers
23:48:42	7:47:02

The time required to complete a simulation on the Full Melbourne network was reduced to slightly under 1/3 once containers were used to accelerate the calculations.

8 Conclusions

The aim of this project was to accelerate Siren's existing shortest path algorithms by spending time in a preprocessing phase prior to running shortest path searches. This has been accomplished by building an algorithm to generate geometric container data in Siren. The containers reduce the average shortest path search time to between 30 and 40 percent, depending on network size. This effect is more pronounced on more complex networks. This is desirable since these networks take longer to solve shortest paths on; therefore their searches require more acceleration.

The method of reusing the shortest path tree during container construction is not known to have been used previously. It results in a considerable reduction of the computation time required for container construction to approximately 1/3 of that required when rebuilding the entire tree at each iteration.

Similarly, there was no prior literature describing the process of reusing the shortest path tree between Dijkstra searches. Since reusing the tree between searches was an existing

optimisation in Siren, it was necessary for it to be implemented with containers to see if it would remain useful. It was shown to cause a performance degradation with containers, and hence will not be implemented in Siren. However, it is useful to know that we are not abandoning a potential optimisation in this case.

References

1. Cherkassky, B., Goldberg, A. and Radzik, T., Shortest Path Algorithms – Theory and Experimental Evaluation, Stanford University/King's College, 1993
2. Wagner, D. and Wilhalm, T., Geometric Speed-Up Techniques for Finding Shortest Paths in Large Sparse Graphs, Universität Konstanz, Germany, 2003
3. Wagner, D., Wilhalm, T. and Zaroliagas, C., Dynamic Shortest Paths Containers, Universität Karlsruhe, Germany/University of Patras, Greece, 2003
4. Wagner, D., Wilhalm, T. and Zaroliagas, C., Geometric Containers for Efficient Shortest Path Computation, Universität Karlsruhe, Germany/University of Patras, Greece, 2004

A Study of Optimised Ambulance Redeployment Strategies

David P. Richards
Department of Engineering Science
University of Auckland
New Zealand
dric038@ec.auckland.ac.nz

Abstract

The Optima Corporation is currently developing an ambulance real-time management system called Siren Live for the Toronto Ambulance Service. Siren Live will be used by the Toronto Ambulance Service to relocate ambulances in real time in order to provide the best ambulance response times possible. A new redeployment model has been developed for use in Siren Live. This model re-distributes the currently available ambulances across alternative base locations in order to try to improve the response coverage. An older redeployment model was implemented within Optima's simulation product called Siren Predict. Our aim is to thoroughly test the prototype redeployment model and to improve the model using Optima's experience and academic literature.

Evaluating the performance of a redeployment model is not possible in the real world without putting people's lives in danger. Therefore we used the simulation tool in Siren Predict to evaluate the performance of the redeployment model.

We have made a number of modifications to the prototype redeployment model. Using the current redeployment model we can increase the percentage of high priority calls responded to within the target time of 8 minutes by up to 20%. However, this is a best case scenario and different results may be produced with a more realistic limited number of redeployments.

1 Introduction

Ambulance organisations are always under pressure to respond faster to emergency calls. To try to improve ambulance response times, the locations of ambulance bases are optimised to provide the best coverage of calls. These coverage plans can start to perform poorly when ambulances become busy. When ambulances are dispatched to calls, we often find that holes in ambulance coverage occur, where the ambulances that had been planned to serve calls in an area are all busy serving other calls. This means that if a call occurs in this area, the first choice ambulance would not be able to respond to the call and therefore, the call may not be responded within the required response time. To remedy this, a redeployment operation is needed to be performed in which the available ambulances are redistributed to cover the hole.

The Optima Corporation is currently developing an ambulance real-time management system called Siren Live. A new redeployment model has been developed for use in Siren Live. This model re-distributes the currently available ambulances across all bases in order to try to cover any holes in the ambulance coverage. Our aim is

to thoroughly test the prototype redeployment model and to improve the model using Optima's experience and academic literature.

The main performance measure of an ambulance system is the percentage of calls that are responded to within a certain target time which depends on the severity of the call. We measure the response time for a call as the elapsed time from when the call is received until the time an ambulance first arrives at the scene.

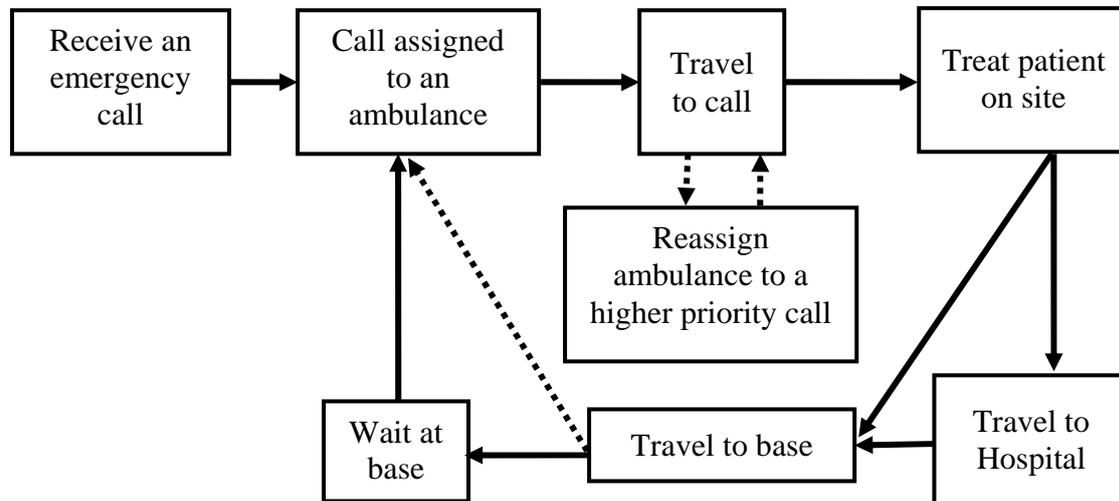


Figure 1. Flow chart of the operational process for ambulances

2 The redeployment model

A redeployment model is used to re-locate some of the currently idle ambulances to new base locations to provide better response coverage. Siren's prototype redeployment model works by:

- 1) Evaluating all the possible locations each ambulance can reach within a drive time limit. The time for an ambulance to travel to a base is the time to travel the shortest path through the road network when travelling at a non-urgent pace which means without 'lights and sirens' and at normal maximum road speeds. This time is dependent on the time of day taking into account the varying road congestion.
- 2) Calculating the ambulance coverage at each location. A zone is considered covered by an ambulance if the ambulance can reach the zone within the coverage time limit. Unlike the time to reach a new base location the time to cover a zone is calculated when an ambulance is travelling with 'lights and sirens'. The coverage of an ambulance is calculated at the look ahead time which is the time at which we optimise the locations of the ambulances. The drive time limit which restricts the distance we can redeploy an ambulance can be larger than the look ahead time. If this is the case then the ambulance will be on the road when we calculate its coverage rather than at its new base location.
- 3) Choosing one location for each ambulance so that each zone is covered by some target number of ambulances.
- 4) Penalising zones that are covered less than the target coverage. Penalties are determined by the zone's expected call arrival rate. Where a zone's call arrival rate is the number of calls that is expected to occur in the zone in a day.

The formulation of the model is shown below:

Indices

- i = idle ambulance 1, 2, ..., M
 j = base 1, 2, ..., N
 k = zone 1, 2, ..., L

Parameters

- Q_i = the current base index of ambulance i .
 C_{ij} = cost of moving ambulance i to base j . There is no cost associated with returning to the current base so $C_{i(Q_i)}=0$.
 N = number of base locations to station an ambulance.
 M = number of idle ambulances.
 L = number of zones.
 P = the maximum allowed number of ambulances relocated in a redeployment operation.
 D_k = the desired number of ambulances covering zone k (target coverage).
 Z_k = the cost of under covering zone k .

$$Y_{jk} = \begin{cases} 1 & \text{if an ambulance at base } j \text{ covers zone } k. \\ 0 & \text{otherwise} \end{cases}$$

$$W_{ij} = \begin{cases} 1 & \text{if ambulance } i \text{ can reach base } j \text{ within the time limit or base } j \text{ is} \\ & \text{ambulance } i \text{'s st andby base.} \\ 0 & \text{otherwise} \end{cases}$$

Decision variables

$$x_{ij} = \begin{cases} 1 & \text{if the ambulance } i \text{ is moved to base } j \\ 0 & \text{otherwise} \end{cases}$$

s_k = the number of ambulances zone k is under covered by.

Redeployment Model

$$1) \text{ Minimise } \sum_{i=1}^M \sum_{j=1}^N x_{ij} C_{ij} + \sum_{k=1}^L s_k Z_k$$

$$2) \sum_{j=1}^N x_{ij} = 1, \text{ for } i = 1, 2, \dots, M$$

$$3) \sum_{j=1}^N (\sum_{i=1}^M x_{ij}) Y_{jk} + s_k \geq D_k, \text{ for } k = 1, 2, \dots, L$$

$$4) \sum_{i=1}^M \sum_{j=1, j \neq Q_i}^N x_{ij} \leq P$$

$$5) x_{ij} \leq W_{ij}, \text{ for } i=1, 2, \dots, M, j=1, 2, \dots, N$$

$$6) s_k \geq 0, \text{ integer, for } k = 1, 2, \dots, L \quad (\text{Logical constraints})$$

$$7) 0 \leq x_{ij} \leq 1, \text{ integer, for } i=1, 2, \dots, M, j=1, 2, \dots, N \quad (\text{Logical constraints})$$

Explanation

- 1) Objective is to maximise coverage while minimising transportation costs.
- 2) Each ambulance must have a location.
- 3) Penalises the solution if the target coverage is not met.
- 4) There cannot be more than P ambulance redeployments. $x_{i1} = 1$ if ambulance i was not moved.
- 5) Can only relocate ambulances to bases that the ambulance can reach within the drive time limit.
- 6) There is no benefit of supplying more coverage than the target.

The value of the penalty of under covering a zone is calculated using the expected call arrival rate of the zone. This call arrival rate is determined by taking a large set of calls and determining how many of these calls occur in each zone. The call arrival rate is then the number of the calls that occurred in a zone divided by the time length of the call set. The call arrival rates are standardised to be a value between 0 and 50. This is done by dividing the call arrival rates by the maximum call arrival rate of all zones and multiplying by 50. The value 50 can change depending on the balance we would like between the two criteria of the objective function, maximising response coverage while minimising ambulance transportation costs.

The target coverage is the desired number of ambulances covering each zone. The target coverage is currently calculated using the number of ambulances that cover each zone given that all ambulances are available. We plan to change how we calculate the target coverage and incorporate queuing theory to determine how many ambulances should be covering each zone given the zone's call arrival rate.

3 Improving the model

3.1 The slack costs (penalties) on the target coverage

When the redeployment model is called in Siren it originally used all the calls that had occurred in the simulation prior to this point to penalize solutions when zones were under covered. When the simulation first starts out there are not many calls and therefore this penalty was not an accurate representation of where calls are likely to occur. We modified Siren so that we could read in a year's worth of calls and extract the number of calls in each zone given a user specified grid. We then modified the redeployment model to use the extracted call locations instead of calls that had occurred in the simulation. This was shown in simulations to produce better suggested relocations for ambulances. We analysed typical emergency call data to identify if we should also make the call penalties time based.

The call analysis showed no real seasonal trends and the daily analysis shown in Figure 3.1 showed no real difference between the days of the week except that Friday and Saturday nights/Saturday and Sunday mornings received more emergency calls than the other nights/mornings. However, what we really wanted to know was the differences between the calls behaviour at different zone locations.

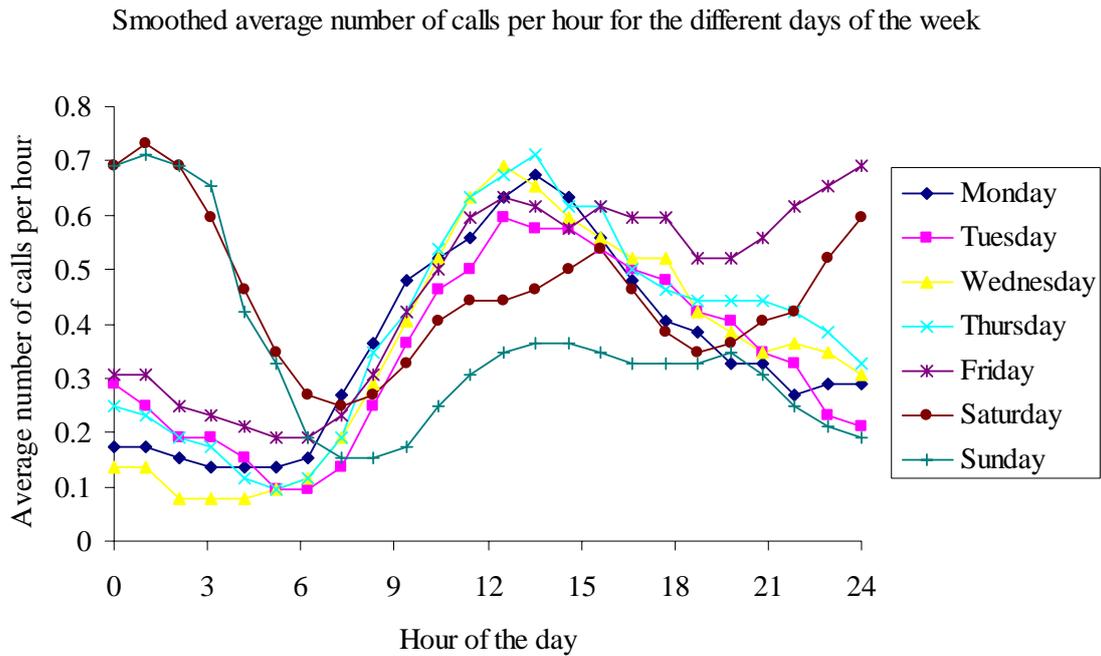


Figure 3.1. Average number of emergency calls per hour of a particular zone for a year's worth of typical emergency call data.

If we compare the calls in Figure 3.1 to that of a different zone in Figure 3.2 then we observe a quite significant difference in the daily trends. Most noticeable is the difference in the number of calls on Friday/Saturday nights and Saturday/Sunday mornings. Since the zones have different call trends we made the call arrival rates time dependent by the hour of the day and the day of the week. Having call arrival rates time dependent also improved the response times of ambulances in the simulations as the redeployments were more specific to the time of day.

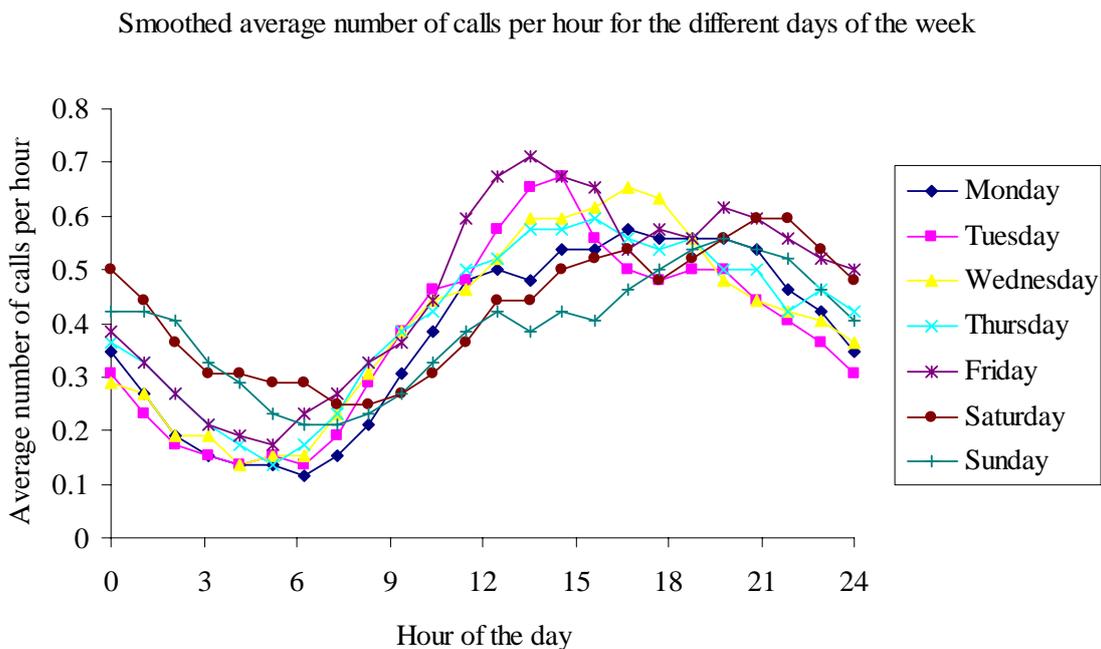


Figure 3.2. Average number of emergency calls per hour of a different zone for a year's worth of typical emergency call data.

3.2 Reducing the time taken to construct the redeployment model

The redeployment model does many shortest path (modified Dijkstra's) calculations when determining the bases that each ambulance can be relocated to within the drive time limit. This can be very computationally expensive. We made the model more than twice as fast by first calculating the straight line distance to each base and comparing the time to travel this distance at 120km/h to the drive time limit. The shortest path calculation is now only done if the base is within the straight line time estimate as it is the minimum time in which the ambulance could travel to the base.

Further analysis of the time to run the redeployment model showed the bottleneck of the redeployment model was the time taken to determine the zones an ambulance covers when located at each zone. The zone coverage constructed was cached so that successive redeployment model runs used the zone coverage already calculated for ambulances that had not moved. However, the zone coverage was refreshed every 30 minutes to take into account the changing road travel times/traffic congestion and this was taking a long time to calculate. The reason for the zone coverage taking so long to calculate was again due to the shortest path calculations. For example if we had 80 idle ambulances and 10,000 zones then we would calculate the shortest path from each ambulance location to each of the 10,000 zones to identify whether the ambulance could reach the zone within the coverage time limit. We improved the construction time dramatically by using the time to travel a straight line distance to each zone at 120km/h and comparing this time to the coverage time limit before calculating the shortest path.

3.3 A new method of solving the redeployment model

The current model formulation is now fairly quick to solve. However, when we are running simulations we may be solving the model every time the number of available ambulances changes. This means we call the redeployment model each time we send an ambulance out to a new emergency call and each time an ambulance is cleared or finishes responding to an emergency call. This can mean that if we want to simulate a months worth of emergency calls, which can be about 17,500 calls, we would call the redeployment model 35,000 times in the simulation. Obviously any reduction in the solve time of the model is therefore beneficial. We developed a new formulation of the model in order to reduce the solve time. The new model formulation is a relaxation of the original model and takes into account that each ambulance has the same coverage if they are at the same location. To understand the new formulation, consider a simple example where there are 3 idle ambulances. The model formulation for this example is shown below.

$$\begin{aligned}
 &\text{Minimise } \sum_{j=1}^N x_{1j} C_{1j} + \sum_{j=1}^N x_{2j} C_{2j} + \sum_{j=1}^N x_{3j} C_{3j} + \sum_{k=1}^N s_k Z_k \\
 &\text{Subject to: } \sum_{j=1}^N x_{1j} = 1, \sum_{j=1}^N x_{2j} = 1, \sum_{j=1}^N x_{3j} = 1 \\
 &\quad \sum_{j=1}^N (x_{1j} + x_{2j} + x_{3j}) Y_{jk} + s_k \geq D_k, \text{ for } k = 1, 2, \dots, L \\
 &\quad \sum_{j=1, j \neq Q_i}^N (x_{1j} + x_{2j} + x_{3j}) \leq P \\
 &\quad x_{1j} \leq W_{1j}, x_{2j} \leq W_{2j}, x_{3j} \leq W_{3j}
 \end{aligned}$$

Each ambulance has the same coverage if they are at the same location so the coverage columns are the same for each ambulance at each base. We can therefore relax the model above by only selecting one possible ambulance to be stationed at each base. The ambulance chosen for a base is the ambulance with the lowest cost associated with travelling to the base. Instead of having the parameter C_{ij} we now have C_{\min_j} which is the minimum cost of locating an ambulance at base j . Also, now there is no constraint on the maximum distance an ambulance can be relocated nor the maximum number of redeployments (constraints 4 and 5 of the original model). However, since we are choosing the lowest cost or travelling to each base we would expect that this travel distance is less than the maximum allowed. The relaxation formulation is shown below; we keep track of each ambulance vehicle by recording the ambulance that belongs to the minimum cost of travelling to base j .

$$\begin{aligned} \text{Minimise } & \sum_{j=1}^N x_j C_{\min_j} + \sum_{k=1}^N s_k Z_k \\ \text{Subject to: } & \sum_{j=1}^N x_j = 3 \\ & \sum_{j=1}^N x_j Y_{jk} + s_k \geq D_k, \text{ for } k = 1, 2, \dots, L \\ & x_j \geq 0, j=1, 2, \dots, N \end{aligned}$$

In the relaxation we can get a solution which has multiple locations for the same ambulance. If this occurs we add in a constraint so that only one location can be chosen out of the locations in the solution with the same ambulance. We also then add to the model the 2nd best option in terms of least cost, for each of the locations in the solution with the same ambulance. The model is solved with the 2nd best options and the process is repeated until a solution is produced that has only one location for each ambulance.

This new formulation should dramatically reduce the solve time as if the model has 80 available ambulances and 90 bases then without the relaxation we have 7,290 decision variables (x_{ij} and s_k) while with the relaxation we start with only 180. However, we are yet to test the impact of using the relaxation model.

3.4 Changing how the response coverage is calculated

The ambulance response coverage can be quite complex to calculate. When ambulances are returning to base they can be assigned to an emergency call when a new calls arrival or to cover another ambulance which is being reassigned to a higher priority emergency. Therefore ambulances returning to their base contribute to the coverage but are moving so it is difficult to evaluate what contribution they make.

Four methods of dealing with ambulances not at a base when calculating coverage are to either:

- 1) Calculate the ambulances coverage at its current location.
- 2) Ignore the ambulance and assume it provides no coverage.
- 3) Assume the ambulance will reach its base soon and calculate the coverage at its base.
- 4) Smear the coverage the ambulance contributes by the probability of a call arriving in its coverage as it travels to its base.

The first method is what the redeployment model is currently doing but it not very realistic as we only look at the coverage of a moving ambulance at the look ahead time.

This may mean we redeploy a vehicle to a base when another ambulance may also be arriving shortly afterwards. If we ignore ambulances not at a base when calculating repose coverage then we incur the same problems as 1) and we do not count for the fact the ambulance can respond to calls while returning home. Assuming ambulances will reach their bases soon and calculating the coverage at their base will mean that we will not double up on ambulances at the base but we may be assuming we have coverage at the look ahead time that we are not likely to get for a long time in the future if the ambulance has a long time to travel. The best option in terms of the most accurate calculation of the coverage is to smear the coverage the ambulance contributes while it is moving. We have not yet implemented this approach but it is expected that doing so will increase the time to solve the model quite significantly.

4 Results

Evaluating the performance of a redeployment model is not possible in the real world without putting people's lives in danger. We used a simulation tool in Siren Predict to evaluate the performance of the prototype redeployment model. These simulations with redeployment can take a very long time to run; as a result we created a program called Siren Manager, which allows us to split up large simulations and run them in parallel on multiple computers.

The results generated in this paper have been from a typical emergency call set and the response performances reported in the results are likely to be similar to what would be seen using real call sets. All the results in this section were produced using the simulation tool inside Siren Predict. The simulations consisted of 1 month's worth of emergency calls which is approximately 17,500 emergency calls.

We ran a number of simulations varying the look ahead time and drive time limit to determine the optimal values for these parameters. The simulations showed that better results were produced when the two parameters had the same value and the best results were produced when this value was about 10 minutes. If the drive time limit is larger than the look ahead time then the redeployment model can locate an ambulance further away then the look ahead time resulting in the ambulance being route to the new location at the look ahead time. As a result the coverage calculated for the ambulance is not very realistic of the actual coverage the ambulance provides in real time. For this reason the look ahead time and drive time limit should be the same value.

The redeployment parameter values for all simulation results in this section are:

- Coverage time limit: 8 minutes
- Max number of redeployments per optimisation: 50 (No limit)
- Look ahead time: 10 minutes
- Drive time limit: 10 minutes
- Fixed cost of a redeployment: 0
- Cost per km travelled in redeployment: 1
- Number of zones: 10,000

4.1 Evaluating the effect of having redeployment

Simulations using redeployment produced faster response times to emergency calls than the simulations without redeployment. The simulations showed:

- Better mean and median response times for the simulations with redeployment. Mean and median response times of all calls decreased from 12.49 and 10.00 minutes to 11.12 and 8.76 minutes respectively.

- The percentage of high priority code 1 calls that were responded to within the target time of 8 minutes increased from 43.8% to 61.7% when using redeployment.

4.2 Limiting the distance of an ambulance from home

Analysing the results from the redeployment simulation we observed that although we were producing large improvements in average response times there were significant increases in the response times of rural areas. What was occurring was the redeployment model was relocating rural ambulances to the city where there are a lot more emergency calls. This was improving the average response times but it is not desirable. To remedy this we put a maximum distance restriction on how far an ambulance can be relocated to from its home base. We made the maximum distance an ambulances redeployment location can be from its home base equal to 120km/hr multiplied by the coverage time limit. This added constraint is also realistic as at the end of a shift the ambulance officers return to their home base and so it is therefore beneficial that the officers do not have to drive far to get back to their home base before their shift ends.

Simulations of the model with restrictions on the maximum distance an ambulance can move from its home base produced faster response times to emergency calls than the simulations without the distance restriction.

- Mean and median response times of all calls decreased a further 0.2 and 0.14 minutes to 10.92 and 8.60 minutes respectively.
- The percentage of high priority code 1 calls that were responded to within the target time of 8 minutes increased by 2.2% to 63.9% with the restriction on the maximum distance an ambulance can move from its home base.

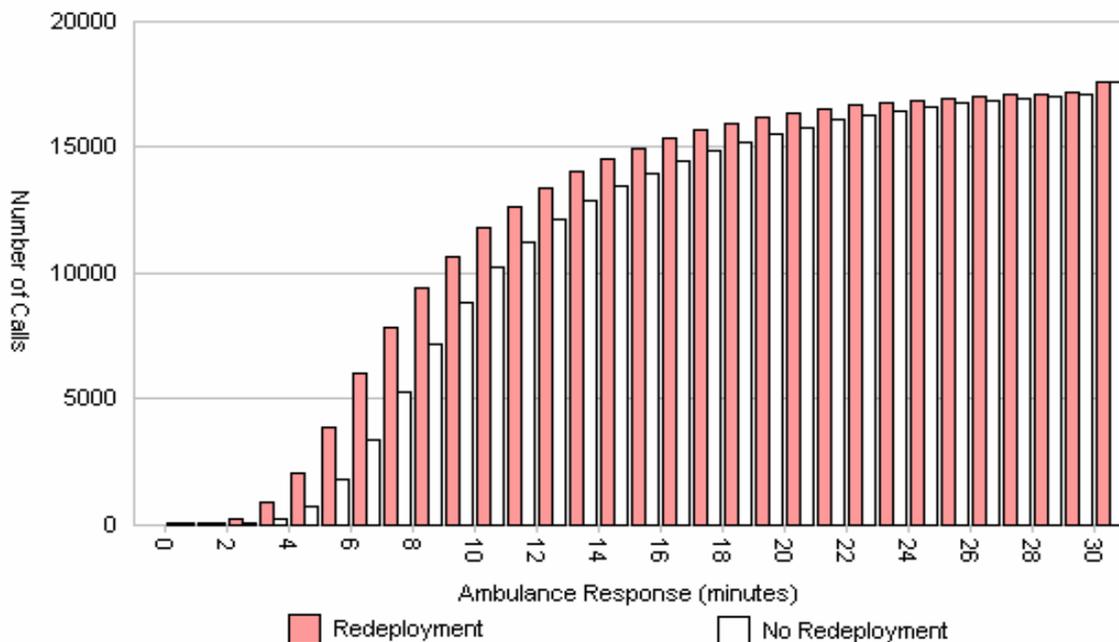


Figure 4. Cumulative number of calls under the response time for simulations with and without redeployment.

4.5 Analysis of ambulance states

In the Siren Predict simulations an ambulance has ten different awake states that it can be in. We modified siren to output the old and new states of an ambulance each time its state changes. These results from a 1 day simulation with redeployment are shown in the Table 4.

Total Count	New State								
Old State	Idle	Going to a location	Responding to a call	Going to hospital	At the call scene	At a hospital	Following another ambulance to hospital	Re-deployment	Grand Total
Asleep	135								135
Idle	4211	87	797		37			433	5565
Going to a location	4	8	42					39	93
Responding to a call	39		1		855				895
Going to hospital						461			461
At the call scene	404		4	463			7		878
At a hospital	782								464
Following another ambulance to hospital						9			7
Re-Deployment	406		59					316	781
Grand Total	5662	95	904	463	892	468	7	788	9279

Table 4. Number of different times an ambulance was in each state and the previous state of the ambulance.

Table 4 shows the majority of ambulances redeployed were originally idle or being redeployed. A lot of ambulances (316) were redeployed to a different base while already travelling to a redeployment location. This is not desired as ambulance officers do not want to be continually redeployed without reaching a destination. However, if we add in a fixed cost of 10 for each ambulance redeployed, than the number of ambulances redeployed to a different base while already travelling to a redeployment location decreases to 5. In total 788 ambulances were redeployed in the 1 day of simulation. This is approximately one redeployment every 1.83 minutes but it is more likely that multiple ambulances are redeployed at the same time so the gap between sets of redeployments would be expected to be larger.

5 Conclusions

We have made a number of modifications to the redeployment model including improving the expected call arrival rates used by producing them from larger call sets and making them time dependent. The model has been changed to build and solve about 60 times faster which was very important in making our analysis possible and we also changed how we calculate the ambulance response coverage and reduced the look ahead time and drive time limit parameters from 30 and 40 minutes respectively to 10 minutes.

Preliminary results from the simulations show that using redeployment can increase the percentage of high priority calls responded to within the target time of 8 minutes by up to 20%. However, this is a best case scenario and different results may be produced with a more realistic limited number of redeployments. We also need to do more analysis on the redeployments and simulations to determine that the benefits we receive are from having the ambulances at better base locations.

Competition Benefits of Line Capacity Expansions in Electricity Markets

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

Abstract

As it is desirable for an electricity market to operate as efficiently as possible, it is important to consider how competition is affected, when upgrading transmission networks. Line capacities have been shown to affect the existence of a Nash-Cournot equilibrium (Borenstein, Bushnell, and Stoft 2000), a state where generators are most competitive. The Grid Investment Test (Electricity-Commission 2006), states that when proposing line expansions, competition benefits should be considered, i.e. competition benefits of expansions need be taken into account if they can be quantified.

In this paper, we extend an existing, two node model to a tree network and derive a set of conditions on the line capacities, which guarantees the existence of a Nash-Cournot equilibrium. These conditions form a set of linear constraints and hence give a convex region.

A capacity expansion model is created, which determines the optimal investment in an electricity network. This model is applied to a simple three-node example.

Work is currently underway to investigate the competition benefits of expanding lines in a simplified model of the New Zealand electricity market with a tree network representing the electricity grid.

1 Introduction

When considering increasing the capacities of lines in the New Zealand electricity grid, any proposal must first pass the grid investment test. The *grid investment test* is a set of criteria that measure the benefits of a grid investment proposal. In this paper, we focus on one of these criteria, namely, the competition benefits associated with a proposed expansion. The benefit from competition is usually lower prices due to the generators competing more aggressively. In New Zealand, as we have an electricity market, generators compete – each trying to maximize their own profits. Because the electricity is distributed via a network, the level of competition can be affected by the constraints imposed by the network.

In order to be able to investigate competition benefits, an analytically tractable model of the electricity market is required. A number of simplifications need to be

made to achieve this; the grid should be simplified and assumptions must be made about the rationality of the generators.

We present a network Cournot model to allow us to investigate the competition effects of changing the capacities of lines within an electricity grid. This model differs from a regular Cournot model, in that the electricity network is represented; this is critical when examining the competition benefits from line expansions. Finding equilibria in this type of model has been shown to be very difficult due to the impact of the capacity constraints on the lines (Borenstein, Bushnell, and Stoft 2000). The network Cournot model has been solved with varying success using an EPEC (equilibrium problem with equilibrium constraints) formulation (Hu and Ralph 2006), however, the EPEC model created is non-convex, which makes finding a Cournot equilibrium difficult. In this paper, we avoid many of these problems by considering only one type of equilibrium, the most competitive, the uncongested Cournot equilibrium – in this equilibrium no line in the network is congested. We develop a set of conditions which allow us to easily determine whether the uncongested equilibrium exists – these conditions form a convex region.

With this generalized set of conditions, we can formulate a multi-stage line capacity expansion model, which ensures optimal competition at every stage; we give an example of this model over a three-node network, which demonstrates what the optimal grid investment decisions might be under different scenarios.

Transpower are interested in this work for a submission they are making to the Electricity Commission. For this, the New Zealand grid is simplified to a tree network, and the model is currently in the process of being set up with appropriate data.

2 Electricity Market Model

2.1 Modeling Assumptions

In order to be able to analyze the strategic behaviour of generators, we must make a number of simplifying assumptions about the electricity market.

2.1.1 Network

An electricity grid can be represented using a network consisting of nodes and directed lines. The *nodes* are the locations where either generators inject electricity onto the grid or consumers take electricity off. The *lines* connect the nodes to each other allowing electricity to be sent around the network. These lines are directed, however, negative flows are possible. Each line has a *capacity*, this is the maximum amount of electricity which can be sent along the line in either direction. In this model, it is assumed that no electricity is lost as it is sent between nodes. In electricity networks, if the lines of the network form a loop (a cycle) there is a constraint on the flow around that loop, caused by Kirchoff's law (Cardell, Hitt, and Hogan 1997). Since we are only dealing with tree networks, loops do not occur.

2.1.2 Generators

In this model, there are a set of generators competing against each other, the strategic generators, and a set of fixed competitive fringes, setting the nodal prices. The *strategic* generators inject quantities of electricity into the grid at zero price, as shown in figure (1) i. (this shows a step-function with price going from zero to

infinity at quantity q_i). The strategic generators are fully rational, profit maximizers, with perfect information i.e. they are Cournot players (Cournot 1838). The assumption of full rationality means that the generators understand how their decisions will affect the prices. For the purposes of this paper, it is assumed that there is one strategic generator per node and they each have no cost of generation. There is a *competitive fringe* at every node. This can either be interpreted as a tactical generator submitting a fixed linear offer curve, as shown in figure (1) ii. or as linear demand elasticity. The competitive fringe sets the nodal prices over the network (Neuhoff et al. 2005).

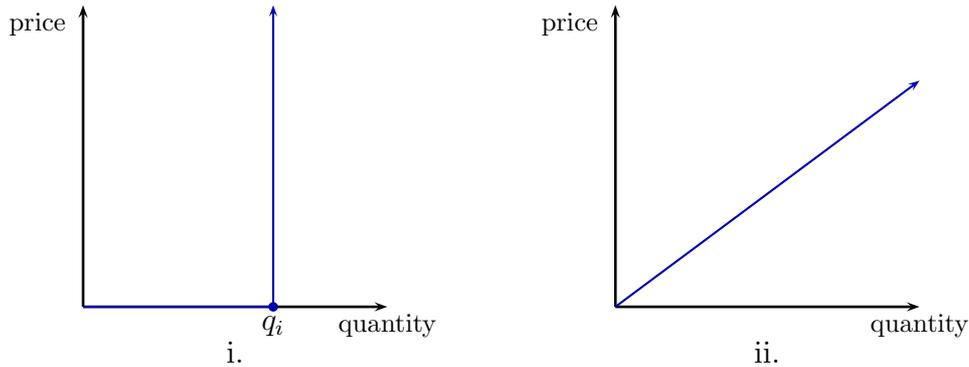


Figure 1: i. Strategic generator's injection; ii. Fringe offer curve

2.1.3 Scheduling, Pricing and Dispatch (SPD)

In the New Zealand electricity market, each day is divided into 48 half-hour periods. In each of these periods, prices of electricity at each node are calculated from the dispatch model, which takes the demand levels and the generators' offer curves as parameters. When considering capacity expansion, the peak periods (periods with highest demand) are most important, as these provide the maximum strain on the system. The demand levels for these periods at each node is assumed to be known by all generators.

2.2 Dispatch Model

The dispatch model is designed to minimize the cost of generation (the total area under the offer curves) and to calculate the nodal prices over the network. Below is a formulation of a simplified dispatch model.

Sets	\mathcal{N} is the set of nodes in the network. \mathcal{L} is the set of lines in the network.
Parameters	a_i is the inverse of the slope of node i 's competitive fringe. q_i is the injection of the strategic generator at node i . d_i is the demand at node i . K_j is the capacity of line j . A_{ij} is 1 if line j enters node i , is -1 if it leaves node i and is 0 otherwise.
Variables	Q_i is the quantity dispatched from the competitive fringe at node i . f_j is the amount of electricity sent along line j .

2.2.1 Single Node

If we first consider all generators and demand to be situated at one node, the formulation is given by the following:

Formulation

1. Minimize $\sum_{i \in \mathcal{N}} \frac{1}{2a_i} Q_i^2$
2. $\sum_{i \in \mathcal{N}} Q_i = \sum_{i \in \mathcal{N}} d_i - \sum_{i \in \mathcal{N}} q_i$

Explanation

1. Minimize cost of the dispatch.
2. The total generation equals the total demand. The nodal price is the dual variable of this constraint, shown in equation (1).

$$\pi = \frac{\sum_{i \in \mathcal{N}} d_i - \sum_{k \in \mathcal{N}} q_k}{\sum_{i \in \mathcal{N}} a_i} \quad (1)$$

2.2.2 Network

We now formulate the problem taking into account network capacities.

Formulation

1. Minimize $\sum_{i \in \mathcal{N}} \frac{1}{2a_i} Q_i^2$
2. $Q_i + \sum_{j \in \mathcal{L}} A_{ij} f_j = d_i - q_i \quad \forall i \in \mathcal{N}$
3. $|f_j| \leq K_j \quad \forall j \in \mathcal{L}$

Explanation

1. Minimize cost of dispatch.
2. Node-balance equation; demand is satisfied at every node. The dual variables of these constraints give the nodal prices, π_i .
3. Maximum flow constraints; the amount of electricity sent along lines cannot exceed the capacity of the line.

2.3 Setting up a competitive game

In an electricity market, it is desired that there is competition between generators, because when generators are more competitive prices are lower, and the market is more efficient.

2.3.1 Elements of a Game

The electricity market is treated as a competitive one-shot game to investigate competition issues. The *players* in the game are the strategic generators; each player has a decision and a payoff. Each player's *decision* is to choose the quantity of electricity to inject into the network at its node. The *payoff* for each player is its profit; i.e. their quantity of electricity times the price at their node. This is a function of the injections of all strategic generators.

3 Analysis of Equilibria

A Nash equilibrium is a competitive state in which no player can unilaterally improve its payoff. In this case, it is where all strategic generators are simultaneously maximizing their profit with respect to their injection quantity.

3.1 Impact of Capacity Constraints.

If there were no capacity limits on the lines, the game would be equivalent to all players being located at the same node. If the competitive fringe is now thought of as demand elasticity, the game becomes a standard one-shot Cournot game and under our assumptions a unique pure-strategy equilibrium will always exist (Cournot 1838).

3.2 Players' Revenue Maximization

In order to find the uncongested equilibrium, it is assumed that no line in the network is at capacity; profit, p_g , of strategic generator, g is then maximized, with respect to its injection, q_g for all strategic generators g :

$$\begin{aligned} \text{Maximize } p_g &= q_g \times \pi \\ p_g &= q_g \times \frac{\sum_{i \in \mathcal{N}} d_i - \sum_{i \in \mathcal{N}} q_i}{\sum_{i \in \mathcal{N}} a_i} \end{aligned} \quad (2)$$

As equation (2) is concave w.r.t. q_g , the optimum quantity, q_g^* , is given by:

$$\begin{aligned} \frac{\partial p_g}{\partial q_g} &= \frac{\sum_{i \in \mathcal{N}} d_i - \sum_{i \in \mathcal{N}, i \neq g} q_i - 2q_g^*}{\sum_{i \in \mathcal{N}} a_i} = 0 \\ \Rightarrow q_g^* &= \frac{1}{2} \left(\sum_{i \in \mathcal{N}} d_i - \sum_{i \in \mathcal{G}, i \neq g} q_i \right) \end{aligned} \quad (3)$$

Solving equation (3) for all generators simultaneously yields a unique, symmetric Cournot equilibrium, with injection quantity, q^C for all strategic generators and nodal price, π^C at all nodes:

$$q^C = \frac{1}{|\mathcal{N}| + 1} \sum_{i \in \mathcal{N}} d_i \quad (4)$$

$$\pi^C = \frac{1}{|\mathcal{N}| + 1} \frac{\sum_{i \in \mathcal{N}} d_i}{\sum_{i \in \mathcal{N}} a_i} \quad (5)$$

Where $|\mathcal{N}|$ is the number of nodes.

3.3 Two Node Case

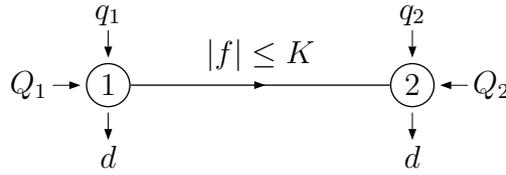


Figure 2: Two Node Symmetric Network

Borenstein, Bushnell and Stoft give an example of a two node network as shown in figure (2), in which the uncongested equilibrium is not a valid equilibrium, if the capacity of the line is small (Borenstein, Bushnell, and Stoft 2000). This is because the line connecting the two nodes gives the players an opportunity to withhold electricity, thus increasing their price. If the line were to congest from node 2 to node 1, the two nodes become separated (in terms of prices) because the strategic generator at node 2 (g_2) cannot send any more electricity to node 1. This potentially allows the strategic generator at node 1 (g_1) to abuse its position.

3.3.1 Uncongested Candidate Equilibrium

The uncongested equilibrium can be calculated for the situation when there line joining the two nodes has unlimited capacity. This, however, is not guaranteed to be an equilibrium when the line has a capacity, hence we will refer to it as a candidate equilibrium – a candidate equilibrium must be later checked as to whether or not it is an actual equilibrium. The candidate equilibrium for this case is given by:

$$q_1 = q_2 = q^C = \frac{2d}{3} \quad \pi_1 = \pi_2 = \pi^C = \frac{d}{3a} \quad (6)$$

This leads to profits for both players of $p^C = \frac{2d^2}{9a}$.

3.3.2 Deviation

As this example is symmetric, only g_1 will be considered. In order for the candidate equilibrium, given in (6), to not be a true equilibrium, there must be another injection q_1^D which leads to a higher profit for g_1 (assuming g_2 injects q^C). The only way to achieve this is by withholding electricity and causing the price to increase. If g_1 acts as a monopolist by withholding electricity and congesting the line toward its own node, we get the following:

$$q_1^D = \frac{d - K}{2} \quad (7)$$

$$p_1^D = \frac{(d - K)^2}{4a} \quad (8)$$

The quantity shown in equation (7) is the optimal monopolistic quantity taking into account how much demand is satisfied by the flow on the line. If the profit from deviating from the uncongested candidate equilibrium, shown in equation (8), is

more than not deviating, the candidate equilibrium is not a true equilibrium. This is equivalent to:

$$\begin{aligned} p_1^D &> p_1^C \\ \frac{(d-K)^2}{4a} &> \frac{2d^2}{9a} \\ K &< d \left(1 - \frac{2\sqrt{2}}{3} \right) \end{aligned} \quad (9)$$

Inequality (9) gives the condition for g_1 to have incentive to withhold electricity (as the network is symmetric this condition is the same for g_2). Thus a condition can be found, on the capacity of the line, which will guarantee the existence of the unconstrained equilibrium. This ensures that no strategic generator finds it profitable to withhold. This condition is shown in inequality (10).

$$K \geq d \left(1 - \frac{2\sqrt{2}}{3} \right) \quad (10)$$

3.4 Tree Network

3.4.1 Conditions for Existence of Cournot Equilibria

The two-node example can be extended for a tree network; this is detailed in an upcoming paper (Downward, Philpott, and Zakeri 2006). Applying the same principles as were applied in the two node case to a tree network, a convex set in the domain of line capacities can be found. If the capacities of the lines lie within this set, shown in inequality (11), the uncongested equilibrium exists.

Sets \mathcal{S} is the set of all connected subtrees in the network.
 \mathcal{N}_s is the set of all nodes in subtree s .
 \mathcal{L}_s is the set of all lines connecting a node within subtree s to a node outside.

Parameters a_i is the inverse slope of the competitive fringe at node i .
 d_i is the demand at node i .
 K_j is the capacity of line j .
 q_k^C is the single-node Cournot offer of generator k .
 π^C is the Cournot price.

$$\sum_{j \in \mathcal{L}_s} K_j \geq \sum_{i \in \mathcal{N}_s} d_i - \sum_{k \in \mathcal{G}_s, k \neq g} q_k^C - 2 \sqrt{q_g^C \pi^C \sum_{i \in \mathcal{N}_s} a_i} \quad \forall g \in \mathcal{N}_s \quad \forall s \in \mathcal{S} \quad (11)$$

4 Line Capacity Expansion Model

In order to create a grid investment model, some assumptions about the costs of upgrading transmission lines need to be made. The total cost is divided into a fixed cost, which is incurred if a line is upgraded, and a constant marginal cost, which is incurred based on how much extra capacity is added to the line. To achieve this cost structure, the model is formulated as an integer programme.

4.1 Detailed Formulation

Sets	\mathcal{N} is the set the nodes in the network. \mathcal{L} is the set of lines in the network.
Parameters	α_j is the fixed cost of choosing to expand the capacity of line j . β_j is the marginal cost of expanding the capacity of line j . L_j lower-bound for expanding line j . U_j upper-bound for expanding line j . K_j current capacities of line j . δ is the discount factor. ρ is the demand growth factor.
Variables	x_j is the amount in MW by which we choose to expand line j . y_j is a boolean variable equal to 1 if line j is upgraded and 0 otherwise.

Formulation

1. $V(\underline{K}, \underline{d}) := \text{Minimize } \sum_{j \in \mathcal{L}} (\alpha_j y_j + \beta_j x_j) + \delta V(\underline{K} + \underline{x}, \rho \underline{d})$
2. $x_j - L_j y_j \geq 0 \quad \forall j \in \mathcal{L}$
3. $U_j y_j - x_j \geq 0 \quad \forall j \in \mathcal{L}$
4. $y_j \in \{0, 1\} \quad \forall j \in \mathcal{L}$
5. $\sum_{j \in \mathcal{L}_s} x_j \geq \sum_{i \in \mathcal{N}_s} d_i - \sum_{k \in \mathcal{G}_s, k \neq g} q_k^C - 2 \sqrt{q_g^C \pi^C \sum_{i \in \mathcal{N}_s} a_i} - \sum_{j \in \mathcal{L}_s} K_j \quad \forall g \in \mathcal{N}_s \quad \forall s \in \mathcal{S}$

Explanation

1. Minimize the cost of expansion for this period plus discounted future costs.
2. A line being expanded must be expanded by, at least, the lower-bound.
3. A line being expanded can be expanded by, at most, the upper-bound.
4. The decision to expand a line is boolean.
5. The new capacities must support the uncongested Cournot equilibrium.

4.2 Three Node Example

Initial Parameters The capacity expansion model will now be applied to a three node network with two lines. The initial demand levels and capacity of the lines are shown in figure 3.

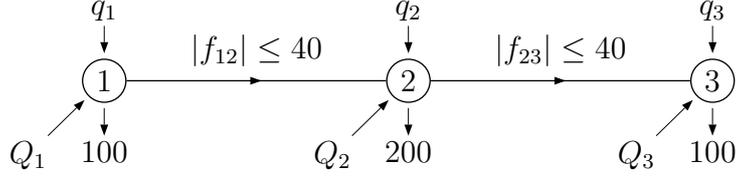


Figure 3: Three Node Network

Conditions for Uncongested Equilibrium For this network structure, the set of conditions (11), to ensure the uncongested equilibrium exists, becomes:

$$\begin{aligned}
 K_{12} &\geq d_1 - 2\sqrt{q^C \pi^C a_1} \\
 K_{12} + K_{23} &\geq d_2 - 2\sqrt{q^C \pi^C a_2} \\
 K_{23} &\geq d_3 - 2\sqrt{q^C \pi^C a_3} \\
 K_{23} &\geq d_1 + d_2 - q^C - 2\sqrt{q^C \pi^C (a_1 + a_2)} \\
 K_{12} &\geq d_2 + d_3 - q^C - 2\sqrt{q^C \pi^C (a_1 + a_2)}
 \end{aligned}$$

Costs For this example it is assumed that there is a cost of $\alpha = 2$ units of choosing to upgrade a line and a cost of $\beta = 1$ unit/MW marginal cost of expanding the line. The fringe slopes are the same at each node.

Results The model was solved using a demand growth rate of 10% per period over 7 periods, the investment plans for three scenarios using different discount factors are shown in table 1, all capacities are in MW.

Period	Minimum Requirement			$\delta = 1.0$		$\delta = 0.9$		$\delta = 0.8$	
	K_{12}	K_{23}	$K_{12} + K_{23}$	K_{12}	K_{23}	K_{12}	K_{23}	K_{12}	K_{23}
1	36.7	36.7	84.5	65.0	84.7	44.5	40	44.5	40
2	40.4	40.4	93.0	65.0	84.7	44.5	57.8	44.5	48.5
3	44.4	44.4	102.3	65.0	84.7	44.5	57.8	53.8	48.5
4	48.8	48.8	112.5	65.0	84.7	66.0	57.8	53.8	58.7
5	53.7	53.7	123.8	65.0	84.7	66.0	57.8	65.1	58.7
6	59.1	59.1	136.1	65.0	84.7	66.0	83.7	65.1	71.1
7	65.0	65.0	149.8	65.0	84.7	66.0	83.7	78.7	71.1

Table 1: Capacity expansion policies with varying discount factors

Table 1 shows the minimum line capacities supporting the unconstrained equilibrium, and also the size of in lines for each period. The plans show that, if the discount factor is near 1, it is better to invest in a few, large expansion projects, however, if the discount factor is smaller, it may be wise to invest in more, smaller line upgrades.

5 Simplifying the New Zealand Electricity Grid

The New Zealand electricity network has over 244 nodes and over 400 lines. Due to the fact that the network is so large, it needs to be reduced down to a tree network

containing only its most important nodes and lines. Given that New Zealand is a long and thin country, representing the network in this way is not unreasonable. Data for this model is currently being collected.

6 Conclusions

- Equilibria do not always exist over networks with capacity constraints.
- A set of conditions for the line capacities, which ensure the existence of an uncapacitated equilibrium, has been developed – these conditions form a convex set.
- A simple capacity expansion model has been created, and implemented on a three-node network. Also a model of New Zealand is underway.

7 Future Work

- Quantify the competition benefits.
- Enable the model to handle loops and losses in the networks.

Acknowledgments

I would like to acknowledge Prof. Andy Philpott and Dr. Golbon Zakeri for their direction and guidance throughout this work. I would also like to thank Dr. Geoff Pritchard and Mr. Bart van Campen for their help analyzing the New Zealand market. Lastly I would like to acknowledge Transpower for giving me the opportunity to apply my research to a real situation.

References

- Borenstein, S., J. Bushnell, and S. Stoft. 2000. “The competitive effects of transmission capacity in a deregulated electricity industry.” *RAND Journal of Economics* 31, No. 2:294–325.
- Cardell, J.B., C.C. Hitt, and W.W. Hogan. 1997. “Market power and strategic interaction in electricity networks.” *Resource and Energy Economics* 19:109–137.
- Cournot, A. 1838. “Recherchés sur les principes mathematiques de la theorie des richesses.”
- Downward, A., A. Philpott, and G. Zakeri. 2006. “On Cournot equilibria in electricity networks.”
- Electricity-Commission. 2006. Grid investment test. World Wide Web, <http://www.electricitycommission.govt.nz/opdev/transmis/git/>.
- Hu, X., and D. Ralph. 2006. “Using EPECs to Model Bilevel Games in Restructured Electricity Markets with Locational Prices.”
- Neuhoff, K., J. Barquin, M.G. Boots, A. Ehrenmann, B.F. Hobbs, F.A.M. Rijkers, and M. Vázquez. 2005. “Network-constrained Cournot models of liberalized electricity markets: the devil is in the details.” *Energy Economics* 27:495–525.

An improved mixed integer programming model for wind farm layout optimisation

Stuart Donovan
Department of Engineering Science
University of Auckland
New Zealand
sdon032@ec.auckland.ac.nz

Abstract

Mixed integer programming (MIP) formulations have the potential to provide a powerful new tool when applied to the field of wind farm layout optimisation. These models seek to determine the optimal positions of wind turbines within a new wind farm, subject to constraints on turbine proximity and turbine interference. This paper outlines several methods that improves the efficiency with which these MIP models can be solved, which make use of criteria designed to reduce the size of the problem, a more effective branching strategy, a stronger model formulation, and dynamic constraint generation.

The size of the problem is reduced by relating the capital cost of the turbine to a minimum payback period. This approach ensures that the model only includes locations in which the value of the electricity generated by the turbine would exceed its capital cost within the minimum required payback period. A branching strategy is implemented that prioritizes branching on decision variables corresponding to turbine positions that generate relatively high power when compared to other positions within the wind farm. The MIP formulation is strengthened by the addition of a simple family of mixed integer rounding inequalities associated with turbine interference. These improvements led to moderate reductions in branch and bound gap and solution time.

1. Introduction

Wind farms are an increasingly attractive means of expanding electricity generating capacity. The trend towards larger farms and more powerful wind turbines means that efficient turbine placement is an increasingly important aspect of wind farm design.

There is significant commercial value in determining improved wind farm layouts. For example, consider a typical New Zealand wind farm involving thirty 1.5MW turbines operating at a capacity factor of 40%, or 600kw per turbine. In this situation, a mere one percent improvement in electricity output would, at current electricity prices, equate to approximately \$17 million in additional revenue over the 20 year lifetime of the turbines.

Current approaches to wind farm layout optimisation employ heuristic algorithms to position turbines within the farm. By contrast, MIP formulations provide a guarantee on the quality of the solution.

Earlier work [1, 2] by the author has established the viability of applying exact methods, such as solving MIP formulations using LP-based branch-and-bound algorithms, to the problem of wind farm layout optimisation. As a continuation of this work, this paper improves the efficiency with which these MIP formulations are solved.

2. Formulation of the Mixed Integer Programming Model

The mixed integer programming (MIP) model is developed, firstly, by superimposing a regular spaced lattice representing possible turbine locations onto the wind farm topography. Positioning a turbine at a particular location creates zones around the turbine related to the size of the turbine and the interference it causes.

In this context, turbine proximity describes an exclusive zone around each potential turbine location that would be required if a turbine was positioned at that location. This leads to relationships between potential locations that are located within each others proximity area, which ensure that if a turbine is positioned at one location, then another turbine cannot be positioned at the other location.

Turbines located outside each others proximity zone, but relatively close to each other, can cause interference that has an adverse impact on the power generating capacity of the turbines. Turbine interference therefore describes a relationship that exists if turbines are positioned at two locations sufficiently close for interference to exist. The magnitude of the interference must therefore be accounted for when considering the power generating potential of turbines positioned at those locations.

The *Wind Farm Layout Optimisation (WFLO)* problem selects the set of lattice points corresponding to turbine positions which, subject to constraints on turbine proximity and turbine interference, generate the most power for a given upper bound on the number of turbines. Therefore, the constraints on the *WFLO* model partition the area surrounding each vertex into three distinct regions. For a location u , these three regions are distinguished as: 1) Those locations at which another turbine cannot be positioned because of their proximity to the turbine positioned at u ; 2) Those locations at which another turbine can be positioned, but where interference will be experienced between the two turbines; and 3) Those locations at which the positioning of a turbine has no effect.

Let $G = (V, E)$ denote a graph with vertices V and edges $E \subseteq V \times V$. In addition, let the vertices V correspond to the locations where turbines can be positioned, and the set of edges E represents vertex pairs between which there exists a relationship, such as turbine proximity and interference. Where a proximity edge exists between two vertices u and v , then there can be no interference edge between the same two vertices. Thus, the set of edges E can be partitioned into E_p and E_I , which represent the sets of proximity and interference edges respectively. In other words, $E = E_p \cup E_I$ and $E_p \cap E_I = \emptyset$. Let $G_p = G(V, E_p)$ and $G_I = G(V, E_I)$ be the subgraphs of G that are induced by the mutually exclusive edge sets E_p and E_I respectively.

Let W_v denote the amount of power that would be generated if a turbine were positioned at vertex $v \in V$. Let the binary variable $x_v = 1$ if a turbine is positioned at vertex $v \in V$ and $x_v = 0$ otherwise. Let I_{uv} denote the magnitude of the power loss between turbines

being placed at vertices u and v , where $(u, v) \in E_I$. Let the binary variable $y_{uv} = 1$ if turbines are positioned at vertices x_u and x_v , where $(u, v) \in E_I$, and $y_{uv} = 0$ otherwise. Due to the structure of the objective function, the integrality condition on y_{uv} can be relaxed. Let k denote a positive integer that defines the maximum number of turbines to be built in the wind farm. A maximal clique based formulation of the proximity constraints is used as this is known to be stronger than edge based formulations [3]. Let Q denote the set of all maximal proximity cliques in our proximity graph G_P . Each maximal proximity clique $Q \in Q$ is a subset of V , and the number of proximity cliques in G_P is a polynomial function of V .

The *WFLO* model is:

$$\text{Maximise} \quad \sum_{v \in V} W_v x_v - \sum_{(u,v) \in E_I} I_{uv} y_{uv} \quad (1)$$

$$\text{Subject to} \quad \sum_{v \in Q} x_v \leq 1, \quad \forall Q \in Q \quad (2)$$

$$x_u + x_v - 1 \leq y_{uv} \quad (3)$$

$$\sum_{v \in V} x_v \leq k \quad (4)$$

$$x \in \{0, 1\}, v \in V, y_{uv} \geq 0, (u, v) \in E_I \quad (5)$$

The objective function (1) is to maximise net power produced, which equates to gross power less interference experienced. Constraint (2) ensures that the model cannot locate more than one turbine in a maximal proximity clique. Constraint (3) ensures that interference edges between pairs of turbines are accounted for. Constraint (4) ensures that no more than the desired number of turbines is positioned. Constraint (5) defines the decision variables x and y as binary and continuous respectively, and the members of the set of interference edges.

3. Case Study

This paper assesses the efficiency of the *WFLO* model in determining an optimal turbine layout within a wind farm measuring 1.7km by 2.2km with a lattice spacing of 100m by 100m. With this spacing, the wind farm involved $17 \times 22 = 374$ possible wind turbine positions. Although lattice spacing of 25m by 25m is more representative of the level of accuracy that would be required for commercial analysis, 100m by 100m provides an adequate level of detail with which to assess the efficiency of the *WFLO* model.

The positions of turbines within the wind farm are optimised for layouts involving 10 – 20 turbines. A relatively large turbine was chosen, which increases the size of the interference field. This subsequently increases the number of interference variables and interference edge constraints that are necessary to define the problem, thereby increasing the difficulty of the problem being solved.

All problems are formulated and solved using AMPL 10.0 and CPLEX 10.0 respectively, on a PC running a 2.2GHz CPU with 2GB of RAM. A maximum solution time limit of one hour was imposed on all problems and default CPLEX settings are used unless otherwise indicated.

4. Minimum Productivity Requirement

Previous work [1, 2] has identified the potential value of incorporating into the pre-processing of the mixed integer programming (MIP) model a minimum productivity requirement (MPR) on potential turbine locations.

MPR pre-processing would calculate the critical power that justifies investment in a turbine and then eliminate locations that generate less than this critical level. Justification for investment has been considered in terms of the time taken to repay the total capital cost of the turbine. Thus, for any vertex $v \in V$, the value of the power generated by a turbine positioned at v must exceed the capital cost of the turbine within a minimum required pay back period.

Let $W_{Critical}$ denote the critical power that justifies investment [kw]; R denote the revenue earned from electricity [\$/kw.hr]; T denote the minimum pay back period [hr]; and C_T denote the capital cost of a turbine [\$/turbine]. The inequality relating these parameters then becomes $W_{Critical} \cdot R \cdot T \geq C_T$

In the example that follows, R , T , and C_T have been taken to be \$0.06, 76703 hours and \$5,100,000 respectively, with all monetary values in New Zealand dollars. Using these values, $W_{Critical}$ can be calculated to be 1108kW. Eliminating all vertices that generate less than this amount reduces the problem size, as shown in Table 1. The symbols Δ and Δ (%) define the absolute and percentage change in the value of the model attributes respectively. The statistics on LP-Relaxation are given for an example problem involving 20 turbines.

Model attribute	MPR		Δ	Δ (%)
	Without	With		
Variables	23394	11788	11606	-50
Binary	374	305	69	-18
Constraints	23767	12169	11598	-49
Non-zeros	76498	40583	35915	-47
LP-relaxation	29507	29505	2	-0.01

Table 1: Effect of MPR pre-processing on problem size

From Table 1, it can be seen that the MPR pre-processing removes 69 binary variables, or 18%, of the vertices in the problem. This has the subsequent effect of approximately halving the total number of variables, non-zeros, and constraints associated with the problem. However, despite this significant reduction in problem size, the value of the LP-relaxation is relatively unaffected.

The implications of these reductions in problem size on MIP performance are shown in Table 2. In this table, the values of the best bound and the best integer solution are presented for comparison. The column ‘‘MIP Gap’’ gives the relative difference between the best bound and the best integer solution after one hour. A gap of zero indicates that the optimal solution to the problem was found within one hour.

Table 2 reveals that the implementation of MPR pre-processing reduced the gap in all instances of the problem. In addition, MPR pre-processing returned improved best integer solutions in eight of the eleven problems tested.

Number of turbines	Best Bound			Best Integer			MIP Gap (%)		
	Without	With	Δ (%)	Without	With	Δ (%)	Without	With	Δ (%)
10	14994	14923	-0.47	14921	14921	0.00	0.49	0.00	-100
11	16437	16414	-0.14	16297	16309	0.07	0.86	0.64	-25.1
12	17910	17873	-0.21	17618	17626	0.05	1.66	1.40	-15.5
13	19348	19326	-0.11	18957	18953	-0.02	2.06	1.97	-4.58
14	20805	20754	-0.25	20183	20273	0.45	2.08	2.37	-23.0
15	2223	22181	-0.19	21388	21536	0.69	3.90	2.99	-23.3
16	23645	23585	-0.25	22684	22754	0.31	4.24	3.65	-13.8
17	25056	25006	-0.20	23935	23896	-0.16	4.68	4.65	-0.82
18	26475	26385	-0.44	25064	25100	0.14	5.63	5.01	-11.0
19	27836	27732	-0.37	26271	26217	-0.21	5.96	5.78	-3.00
20	29233	29101	-0.45	27436	27443	0.03	6.55	6.04	-7.76

Table 2: Effect of MPR preprocessing

MPR pre-processing is a simple and effective method of improving MIP performance. In light of the above results, all future instances of the *WFLO* model that are presented in this paper incorporate the MPR pre-processing.

5. Power Rank Branching Strategy

The application of a customised branching strategy improved the performance of the *WFLO* model. The branching strategy incorporated additional information on the spatial and power relationships that existed between sets of vertices in the problem.

Let $N_v = \{u \in V : (u, v) \in E_p\}$ denote the neighbouring vertices of vertex $v \in V$, $\bar{P}(N_v) = \left(\frac{1}{|N_v|} \right) \sum_{u \in N_v} P_u$ denote the mean power generated by the neighbours of vertex $v \in V$.

The relative power RP_v for each binary decision variable x_v was then defined to be:

$$RP_v = \left(\frac{\bar{P}(N_v)}{\max_{v \in V} \{\bar{P}(N_v)\}} \right) \times \left(\frac{P_v}{\max_{(u, v) \in E_p} \{P_u\}} \right)$$

The two terms in brackets consider different relative values of the power generated by a turbine positioned at vertex v . The first term considers the relative value of the mean power produced by the neighbours to vertex v in comparison to the maximum mean power of the neighbours to every other vertex in the graph. The second term considers the relative value of the power produced at v as compared to its best neighbour. This

essentially compares the power available at v to the maximum power available at any of the vertices u that would be eliminated as a result of the proximity edges between v and u .

The vector of vertices V was then sorted in descending order according to the value of RP_v . The position of the vertices in this vector determined the relative priority of the corresponding vertices in the model. This means that variables with the higher priority would be branched to one first.

This priority ordering of the variables is referred to as the Power Rank Branching Strategy (PRBS). Table 3 shows the implications of the PRBS on the performance of the MIP model.

Number of turbines	Best Bound			Best Integer			MIP Gap (%)		
	Without	With	Δ (%)	Without	With	Δ (%)	Without	With	Δ (%)
10	14923	14923	0.00	14921	14921	0.00	0.00	0.00	-
11	16414	16311	-0.63	16309	16309	0.00	0.64	0.00	100
12	17873	17819	-0.30	17626	17617	-0.05	1.40	1.15	18.2
13	19326	19234	-0.48	18953	18947	-0.03	1.97	1.52	23.0
14	20751	20655	-0.48	20273	20213	-0.30	2.37	2.19	7.84
15	22181	22062	-0.54	21536	21471	-0.30	2.99	2.75	8.09
16	23585	23444	-0.60	22754	22716	-0.17	3.65	3.21	12.3
17	25006	24828	-0.71	23896	23940	0.18	4.65	3.71	20.2
18	26358	26195	-0.62	25100	25174	0.29	5.01	4.06	19.1
19	27732	27566	-0.60	26217	26326	0.42	5.78	4.71	18.5
20	29101	28925	-0.60	27443	27442	0.00	6.04	5.40	10.6

Table 3: Effect of PRBS on solution performance

Table 3 reveals that the use of PRBS returned reduced best bounds for all of the eleven problems tested. However, PRBS was unable to find improved best integer solutions in five of the problems tested. This suggests that PRBS improves the rate of reduction in best bound at the expense of finding improved integer solutions. In terms of MIP Gap, PRBS was able to significantly reduce the difference between the best bound and the best integer solution.

Using PRBS as outlined above is a simple and effective method of improving MIP performance. However, a more thorough investigation into alternative branching strategies will be undertaken in the future.

6. Mixed Clique Inequalities

It is well known in the literature [3] that clique based formulations for vertex packing problems are stronger than the edge based form. Furthermore, maximal clique inequalities can be derived from edge inequalities by recursively aggregating valid inequalities and applying integer rounding [3]. These results motivate the derivation of mixed clique inequalities (MCI) by applying a similar recursive procedure using mixed integer rounding (MIR) [4].

Consider a complete subgraph K_n of G_I induced by the vertices $\{v_1, \dots, v_n\} \subseteq V$.

Let $X_n = \sum_{i=1}^n x_{v_i}$ where $0 \leq X_n \leq n$ and integer, denote the sum of the variables associated the vertices $\{v_1, \dots, v_n\}$. Let $E_I(K_n)$ denote the set of edges in K_n and $Y_n = \sum_{(u,v) \in E_I(K_n)} y_{uv}$ where $y_{uv} \geq 0$, denote the sum of the variables associated with $E_I(K_n)$.

Consider a K_3 complete subgraph in G_I . The three interference constraints corresponding to the edges are shown below, as well as the aggregated inequality.

$$x_{v_1} + x_{v_2} \leq 1 + y_{12}$$

$$x_{v_1} + x_{v_3} \leq 1 + y_{13} \Rightarrow 2x_{v_1} + 2x_{v_2} + 2x_{v_3} \leq 3 + y_{12} + y_{13} + y_{23}$$

$$x_{v_2} + x_{v_3} \leq 1 + y_{23}$$

The aggregated inequality can be written as $2X_3 \leq 3 + Y_3 \Rightarrow X_3 \leq \frac{3}{2} + \frac{1}{2}Y_3$. Applying MIR to this inequality yields the K_3 MCI $X_3 \leq 1 + Y_3$.

Similarly, aggregating the four K_3 MCI associated with a K_4 and applying MIR yields the K_4 MCI $X_4 \leq 1 + Y_4$. It can be shown for any K_n that the corresponding MCI is $X_n \leq 1 + Y_n$ and that these inequalities induce facets of the convex hull of the set of feasible solutions to the constraints (3) and (4) [5].

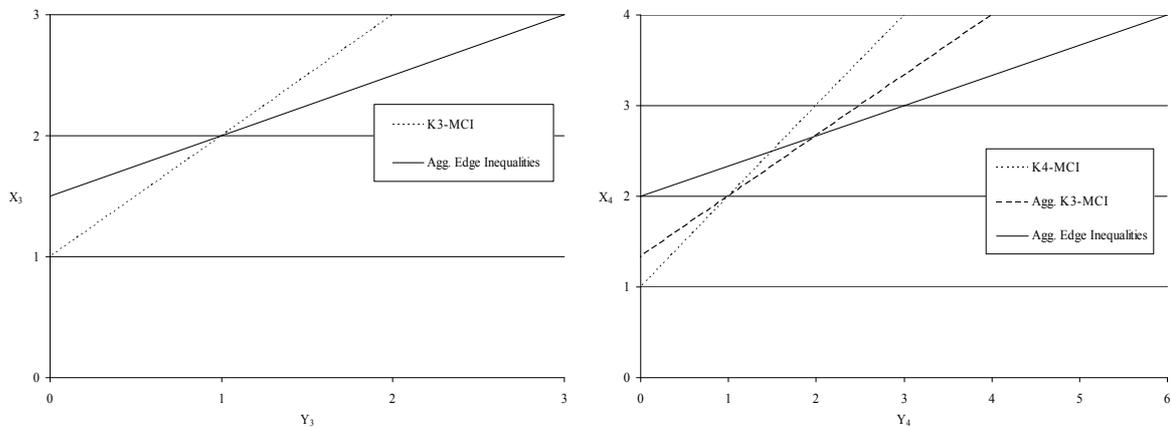


Figure 1: Mixed Clique Inequalities for K_3 and K_4

The left hand plot in Figure 1 indicates that the extreme point of the LP-relaxation of the MIP corresponding to $(Y_3, X_3) = (0, 3/2)$ is cut off by the K_3 MCI. This point corresponds to $(0, 0, 0, 1/2, 1/2, 1/2)$ in $(y_{12}, y_{13}, y_{23}, x_{v_1}, x_{v_2}, x_{v_3})$ space respectively. In addition, the right hand plot in Figure 1 indicates that the K_4 MCI cuts off the point $(0, 0, 0, 0, 0, 0, 1/3, 1/3, 1/3, 1/3)$, which corresponds to $(0, 4/3)$ in $(y_{12}, y_{13}, y_{14}, y_{23}, y_{24}, y_{34}, x_{v_1}, x_{v_2}, x_{v_3}, x_{v_4})$ space respectively. Work continues on characterising the inequality that cuts off the extreme point $(8/3, 2)$.

Unlike the case for clique inequalities in vertex packing, these MCI do not dominate each other, as demonstrated by the aggregated edge constraints, aggregated K_3 MCI, and K_4 MCI in the right hand plot in Figure 1. It is expected that the effect of including MCI on the LP-relaxation (LP-R) of the MIP model diminishes as the number of vertices in K_n increases.

Preliminary investigations into the impact of MCI on MIP performance have revealed only modest results. MCI were enumerated and added for K_3 and K_4 for an instance of the problem for which the value of the optimal integer solution was known to be 14941. The inequalities were added as CPLEX user cuts.

The impact of the MCI was assessed by measuring the improvement to value of the LP-R. As the *WFLO* model seeks to maximize the objective function, improvement is characterised by a reduction in the value of the LP-R.

K_n -MCI	Number of MCI	LP-R	
		Without	With
3	115715	15370	15355
4	438572	15355	15354

Table 4: Effect of MCI on the LP relaxation

Table 4 indicates that both the K_3 and K_4 MCI have a relatively small impact on the size of the gap between the LP-R and the optimal integer solution. It is also interesting to note the increasingly large number of MCI that were required to be added to the problem for K_3 and K_4 respectively.

7. Dynamic Generation of Violated Interference Constraints

The results in [1, 2] showed the *WFLO* model without interference edges solved extremely quickly. This results from the relatively small number of proximity constraints when compared to interference constraints.

The proximity constraints are necessary to identify feasible solutions to the problem. In contrast, the interference constraints identify optimal solutions to the problem. Therefore, the efficiency of the MIP formulation may be improved by using the proximity constraints to identify feasible solutions and then implementing the interference constraints as is necessary to define solution optimality. This was investigated by adding the interference constraints as CPLEX lazy constraints. This simulated the effect of dynamically generating violated interference constraints (DGVIC) by placing them in a cut pool and letting CPLEX separate them. The effect of this modification on the performance of the MIP model for a small number of problems is shown in Table 5.

Number of turbines	Time to solve (seconds)		Lazy Constraints Applied
	Without	With	
10	1334	422	1842
11	3374	1405	2427

Table 5: Effect of DGVIC on solution performance

Table 5 shows that the application of DGVIC significantly reduces the time to prove optimal integer solutions to these problems.

8. Conclusions and Future Work

From these results it is concluded that the minimum productivity requirement (MPR), power rank branching strategy (PRBS), and dynamically generating violated interference constraints lead to significant improvements in the efficiency of the MIP model for determining the optimal turbine layout in a wind farm. The value of the mixed clique inequalities (MCI) is less certain as a result of their large number and modest impact on the LP-relaxation. The opportunity exists, however, for further investigation into the application of separation techniques to identify and add violated MCI to the problem as they are required. As the tractability of the MIP model is largely dependent on the strength of the interference edges, it is considered important that priority be given to investigating alternative vertex packing structures, such as odd holes and webs [6], which exist within the interference subgraph and may suggest valid inequalities that would strengthen the LP-relaxation of the MIP. While further work should also seek to refine the gains made using MPR preprocessing and PRBS, it is considered unlikely that these methods alone will result in the efficient treatment of problems involving more than fifty turbines. A comparison of results from the MIP model with those generated from existing industry software is also necessary, which should primarily focus on the quality of the solution found in terms of power production and solution time.

9. References

- [1] S. Donovan, "Wind Farm Optimization," Proceedings of the 40th Annual Conference of the Operations Research Society, Wellington, 2005.
- [2] S. Donovan, "Wind Farm Optimization," Proceedings of the 7th Triennial Conference of the Asia-Pacific Operations Research Society, Manila, 2006.
- [3] G. L. Nemhauser and L. A. Wolsey, "Integer and combinatorial optimization", Wiley, New York, 1988.
- [4] L. A. Wolsey, "Integer Programming," Wiley, New York, 1998.
- [5] H. Waterer, Personal Communication, September 2006.
- [6] M. Padberg, "On the facial structure of set packing polyhedra," Mathematical Programming, Vol. 5, pp 199-215, 1973.

Peak Shaving and Price Saving: Algorithms for Consumer Generation

David N. Craigie
Department of Engineering Science
University of Auckland
New Zealand
dcra036@ec.auckland.ac.nz

Abstract

Large scale industrial and commercial electricity consumers may have generation capacity, typically in the form of diesel generators, to guarantee security of supply. But self generation can also be a means of reducing energy costs due to lines company tariffs and high spot market prices.

We examine the problem of optimally utilizing a generator to achieve the greatest savings in these areas. While this problem can be formulated easily as a linear programme, it is not easily solved as such due to an intractable number of constraints.

We propose an alternative optimal algorithm for a deterministic scenario that is easily implemented. However, this algorithm is not easily extendable to scenarios with stochastic spot market prices. For such cases we propose using the dual simplex method on a relaxation of the stochastic programme.

We implement these algorithms using demand data from the University of Auckland and spot market prices from the Otahuhu reference node. We find that the optimal allocation of generator fuel reduces the cost of the security of supply.

1 Introduction

There is growing interest in the role played by consumers in electricity markets and in particular large industrial and commercial consumers with loads that account for a significant proportion of the aggregate demand. Decisions made by supply-side participants to invest in new generation and network upgrades are the most publicized and undoubtedly the most influential decisions affecting the future of an electricity market. However, demand-side management is likely to attract further analysis as the problem of meeting growing electricity demands and various environmental and political constraints becomes more pressing both in New Zealand and abroad. One aspect of demand-side management is self-generation by consumers. In this paper we pose a problem for self-generators that can be easily understood yet offers an interesting optimization challenge and we propose a solution that makes use of the problem's unique structure.

Typically a consumer's electricity costs will consist of a charge for the electricity consumed and some additional charge levied by the lines company for the use of their assets. In most electricity markets under normal conditions, it is more economical for most consumers to draw power from the grid than it is to generate their own. Some consumers will opt to have generation capacity to ensure security of supply. Under some price scenarios it may be preferable to utilize this capacity rather than the grid. However, even if such price scenarios don't occur, generator fuel purchased to guarantee security of supply

has a limited useful life and a sunk cost. This fuel should be used before its shelf life is exceeded. In this paper we examine the problem of using such fuel in an optimal manner to reduce costs incurred due to high spot market prices and lines company charges applied to periods of peak consumption.

2 The Deterministic Peak Shaving Problem

2.1 Problem Definition

In the Deterministic Peak Shaving Problem we consider a finite number of periods of equal duration. In each period there is both a price and demand realization. We take each of these to be known, or at least a single point estimate. We assume the consumer is purchasing electricity directly from the spot market, as large consumers are able to do. Note also, that in this paper the terms demand and load are interchangeable. The problem then can be phrased as: “Given a load profile, price profile and a capacity of generation, find the optimal allocation of a *limited* quantity of fuel with a *sunk cost*, in order to minimize the cost of consumption.”

This is the problem definition we will use in this paper. However, some may find the stipulation of a limited quantity of fuel to be unnecessary or unrealistic and might prefer this definition: “Given a load profile, price profile and capacity of generation, find the optimal allocation of an *unlimited* quantity of fuel with a given *fixed cost*, in order to minimize the cost of consumption.”

While the linear programming formulations of these two problem definitions differ, the algorithm we propose for their solution does not, except in its stopping criterion. Note that we refer to a capacity of generation. It is probably easiest to think in terms of a single generator. However, this definition allows for more than one generator provided they are of identical specifications. What we seek is a solution that defines how much fuel to use in our generator(s) in each period to achieve the maximum savings on the cost of our consumption. The algorithm proceeds to allocate fuel in order to maximize savings until the fuel supply is exhausted in the case of the first definition, or until the value of the savings does not exceed the cost of the fuel in the case of the second definition.

The cost of consumption consists of two components. The first is the cost of the electricity itself, which in a given period is equal to the load drawn times the price per unit of electricity. The second component is charged by the lines company. This may take different forms but we will use Vector’s tariff scheme for customers with Time of Use meters in the Auckland region. It is a charge (per kWh) applied to the average of the 10 highest load realizations in a given month (Vector Ltd, 2006). In the formal definition that follows we will generalize the number of load realizations to which this charge applies.

2.2 A Linear Programming Formulation

Indices

i = period: $1, \dots, N$.

j = set of M periods out of a possible N : $1, \dots, {}^N C_M$

k = period in the j th set of M : $1, \dots, M$.

Parameters

N = total number of periods.

M = number of periods to which the line charge will apply.

P_i = electricity price in period i .

D_i = demand in period i .

- S_j = the j th set of M periods
 T = total quantity of fuel available.
 Q = capacity of the generator in a single period.
 $K_i = \min\{D_i, Q\}$ the maximum amount of fuel that can be allocated to period i .
 C = the maximum demand charge applied to the sum of the greatest M loads.

Decision Variables

- x_i = the amount of fuel to allocate to generation in period i .
 md = the sum of the maximum M demand realizations.

Model PkShvLP

- 1) Minimize $\sum_{i=1}^N P_i(D_i - x_i) + Cmd$.
- 2) $\sum_{i=1}^N x_i \leq T$.
- 3) $x_i \leq K_i$ for $i = 1, \dots, N$.
- 4) $md \geq \sum_{k \in S_j} (D_k - x_k)$ for $j = 1, \dots, N C_M$.
- 5) $x_i \geq 0$ for $i = 1, \dots, N$.

Explanation

- 1) Objective is to minimize total consumption cost.
- 2) Cannot generate more than the total fuel supply will allow.
- 3) Cannot generate more than the capacity of the generator in a given period.
- 4) Maximum demand variable must be equal to the sum of the M periods of highest load less self-generation.
- 5) Cannot allocate a negative amount of fuel in any period.

For simplicity we assume a 1:1 ratio for conversion of fuel to electrical energy. An appropriate conversion factor can easily be included.

The objective function given by 1) in PkShvLP can be simplified and written as a maximization objective as follows:

- 6) Maximize $\sum_{i=1}^N P_i x_i - Cmd$

This objective seeks to maximize savings which is equivalent in this context to minimizing costs. When we allocate fuel to a given period i we get a saving from that period in the objective function from the term P_i and if the allocation reduces the value of the variable md through constraints 4) then we also get a saving from the term C .

Thus if a consumer's periods of highest demand do not coincide with the market's periods of highest price, then there is a tradeoff to be made; namely between generating in periods of high price and generating in periods of high demand and thus reducing the load to which the line tariff applies.

It is the combinatorial number of constraints 4) that makes this otherwise appealing model intractable. For example, if we were considering a month of half-hourly periods we would have $N = 1488$ and using Vector's tariff scheme $M = 10$. This gives 1.4×10^{25} constraints. We have to constrain all sets of M from N because we don't know a priori which M periods will have the greatest load following our generation decisions.

2.3 The Peak Shaving Algorithm

The above model is intractable due to a combinatorial number of constraints. However, we have a greedy algorithm that solves the Deterministic Peak Shaving Problem to optimality and in polynomial time. It is able to exploit the problem's structure, in that starting from a solution of zero allocations ($x_i = 0$ for all i) we know at every iteration which of the constraints labeled 4) from PkShvLP will be the next to bind. The algorithm in pseudo code is given below. Indices, parameters and decision variables that appeared in PkShvLP have the same definitions here, only newly introduced parameters are defined.

Parameters (at each iteration)

V_i = the saving obtained from one unit of fuel used for generation in period i for $i = 1, \dots, N$.

MX = the set of the M highest ranking periods by load.

$D_{(i)}$ = the i th largest load.

V_{N+1} = the savings obtained from one unit of fuel used optimally for generation in periods of demand equal to $D_{(M)}$. We label such periods as being in a tie.

l = minimum number of periods in a tie to which fuel must be allocated for any proportion of the saving C to be obtained.

u = size of the tie.

Peak Shaving Algorithm (PSA)

1) Initialise: Begin with an allocation of zero for each period, for all $i = 1, \dots, N$:

$$x_i = 0$$

2) Calculate savings for each individual period, for all $i = 1, \dots, N$:

$$V_i = \begin{cases} P_i + C & \text{if } i \in MX \text{ and } x_i \leq K_i \\ P_i & \text{if } i \notin MX \text{ and } x_i \leq K_i \\ 0 & \text{otherwise} \end{cases}$$

3) If there is a tie (if $D_{(M)} = D_{(M+1)}$) determine the minimum number of periods in that tie, l , which require generation in order to obtain some portion of the saving C . Also determine the size of the tie, u , that is how many periods i have $D_i = D_{(M)}$.

4) Calculate the best periods in the tie in which to generate. These will be the highest priced n periods in the tie where n satisfies:

$$\max_n \left\{ V_{N+1} = \frac{1}{n} \sum_{i=1}^n p_{(i)} + \left(\frac{n-l+1}{n} \right) C \right\} \text{ And } l \leq n \leq u$$

5) Determine the maximum savings:

$$\max_{i=1, \dots, N+1} \{V_i\}$$

6) Allocate fuel to the period(s) that give the maximum savings until:

(a) Fuel supply is exhausted – Stop.

(b) Capacity of one of the periods is reached – return to 2).

(c) The period(s) load(s) have been reduced to the level of the highest ranking period by load outside of the MX set – return to 2).

3 Numerical Illustrations

In this section we introduce two examples. The first is a small example intended to illustrate the above PSA algorithm. The second is an example using real-world data from the University of Auckland’s Engineering School.

3.1 An Explanatory Example

Here we will implement the PSA using a tabular representation on an example with just 5 periods. The initial load and price profile is given in Figure 1. Suppose we have 16 MWh of fuel to allocate amongst these 5 periods with a maximum allowable allocation of 4MWh of fuel to any one period. How should we allocate this fuel in order to reduce the cost of consumption by as much as possible? In addition to the cost paid for the volume consumed in each period suppose we are charged \$15 times the sum of the largest 2 load realizations. Our initial cost is then given by: $(6 \times 72) + (10 \times 68) + (12 \times 60) + (11 \times 65) + (5 \times 85) + 15 \times (12 + 11) = \$3,317$.

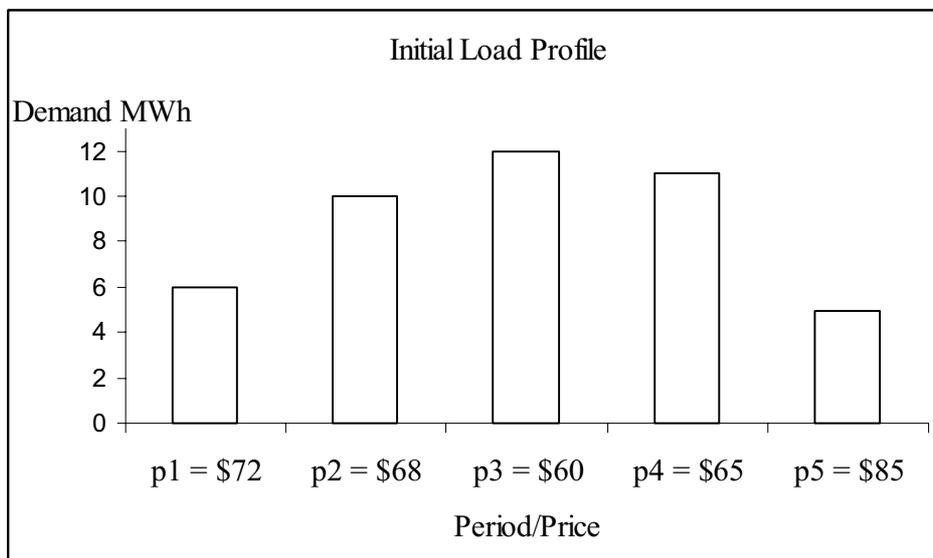


Figure 1. Initial Load Profile.

The PSA is implemented in a tabular representation in Figure 2. The first three column headings are fairly self-explanatory. The column labeled “MX” contains a “Y” if that corresponding period is in the set of maximum demands, which in this example is of cardinality 2 and an “N” if it is not uniquely in this set. By uniquely, we mean that if its load were reduced then the sum of the 2 highest loads would be reduced. The column labeled “Saving” gives the saving that will be obtained by allocating 1 unit of fuel to the corresponding period(s). The column labeled “Range” gives the number of units that may be allocated before the saving per unit will change for that period(s).

In iterations 3, 4 and 5 in this example there is a tie for the period of second largest or even largest demand. In these cases we consider allocating fuel equally to each period in the tie so as to reduce the sum of the largest 2 loads. The saving is given by the average of the prices in the tie and a proportion of the maximum demand charge, which is in this case \$15. The general formula for calculating the value of a tie is given in step 4) of the PSA but

as an example the saving in iteration 3 of allocating to periods 2 and 4 equally is given by the average of the prices of these 2 periods plus half of the maximum demand charge. The allocation decisions made by the algorithm are denoted by the shaded rows.

Iteration 1

Period	Load	Price	MX	Saving	Range	Allocation
1	6	72	N	72	4	0
2	10	68	N	68	4	0
3	12	60	Y	75	2	0
4	11	65	Y	80	1	0
5	5	85	N	85	4	0

Iteration 2

Period	Load	Price	MX	Saving	Range	Allocation
1	6	72	N	72	4	0
2	10	68	N	68	4	0
3	12	60	Y	75	2	0
4	11	65	Y	80	1	0
5	1	85	N	0	0	4

Iteration 3

Period	Load	Price	MX	Saving	Range	Allocation
1	6	72	N	72	4	0
2	10	68	N	68	4	0
3	12	60	Y	75	2	0
4	10	65	N	65	3	1
5	1	85	N	0	0	4
{2,4}				74	3	

Iteration 4

Period	Load	Price	MX	Saving	Range	Allocation
1	6	72	N	72	4	0
2	10	68	N	68	4	0
3	10	60	N	60	2	2
4	10	65	N	65	3	1
5	1	85	N	0	0	4
{2,4}				74	2	
{2,3,4}				74.33	2	

Iteration 5

Period	Load	Price	MX	Saving	Range	Allocation
1	6	72	N	72	4	0
2	8	68	N	68	2	2
3	8	60	N	0	0	4
4	8	65	N	65	1	3

	5	1	85	N	0	0	4
{2,4}					74	1	
<i>Iteration 6</i>							
Period	Load	Price	MX	Saving	Range	Allocation	
1	6	72	N	72	1	0	
2	7	68	N	68	1	3	
3	8	60	Y	0	0	4	
4	7	65	N	0	0	4	
5	1	85	N	0	0	4	

Figure 2. Peak Shaving Algorithm Example in Tabular Form.

The resulting load profile is shown in Figure 3. The new cost is given by: $(5 \times 72) + (7 \times 68) + (8 \times 60) + (7 \times 65) + (1 \times 85) + 15 \times (8 + 7) = \$2,081$. This is the lowest cost attainable using the fuel available.

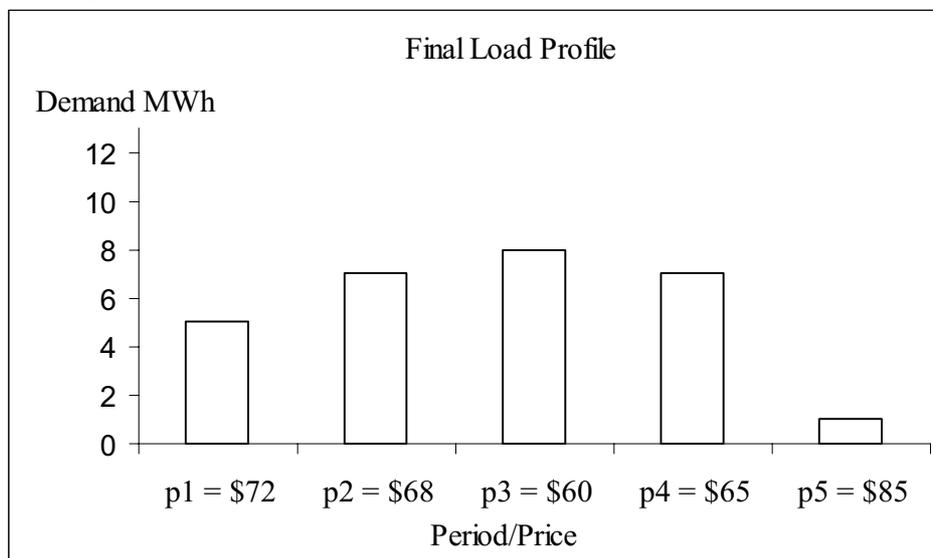


Figure 3. Final Load Profile.

3.2 An Example with Real World Data

We now present an example using demand data for the month of August 2006 from the Time of Use meter at The University of Auckland’s Engineering School. As the month is divided into half hourly trading periods we have 1488 periods. The prices for this example come from the Otahuhu reference node. There is a unique price and demand realization in each period. We apply a maximum demand charge of \$8/kWh to the average of the highest 10 load realizations or equivalently 8c/kWh applied to the sum of the highest 10. We assume we have 1 diesel generator of 100kW capacity and an efficiency of 3.6kWh/L and a total fuel supply of 14000L. The cost of the initial load profile is \$36,164. Following application of the PSA this is reduced to \$32,102. In all likelihood, one would not find 14000L of fuel for approximately \$4000, so this is not a profitably strategy per se. However, if the fuel was necessary for security of supply and was approaching expiration, the allocation given by the PSA would maximize its value given its cost is sunk. Figure 4 gives the highest 14 load realizations before and after the application of the algorithm.

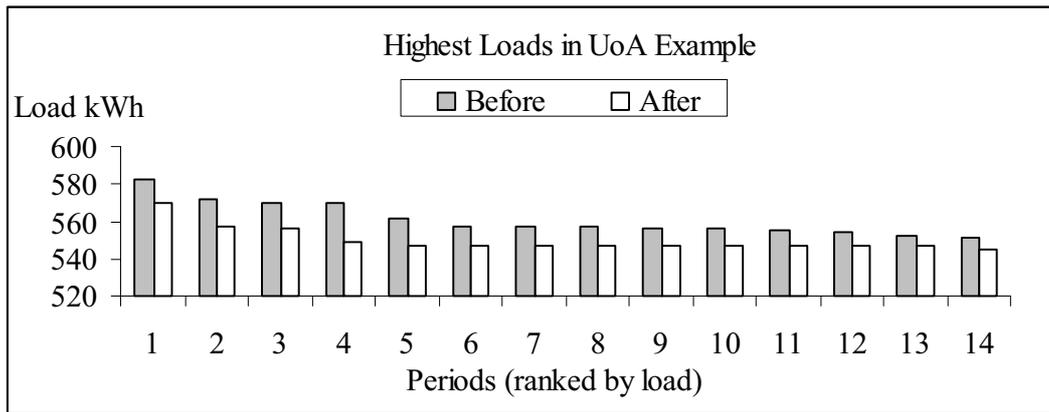


Figure 4. PSA Example Using UoA Data.

4 The Stochastic Peak Shaving Problem

4.1 Problem Definition

In the Stochastic Peak Shaving problem, rather than assume we know the price in a given period, or use a single point estimate of it, we allow for a price process with uncertainty. We can represent this in a scenario tree as in Figure 5. Each node in the tree represents a different price and has some probability of occurrence. The probabilities associated with nodes of a common period sum to 1. Although there are 14 prices, because of the structure of this small example we can say there are effectively 2 price states.

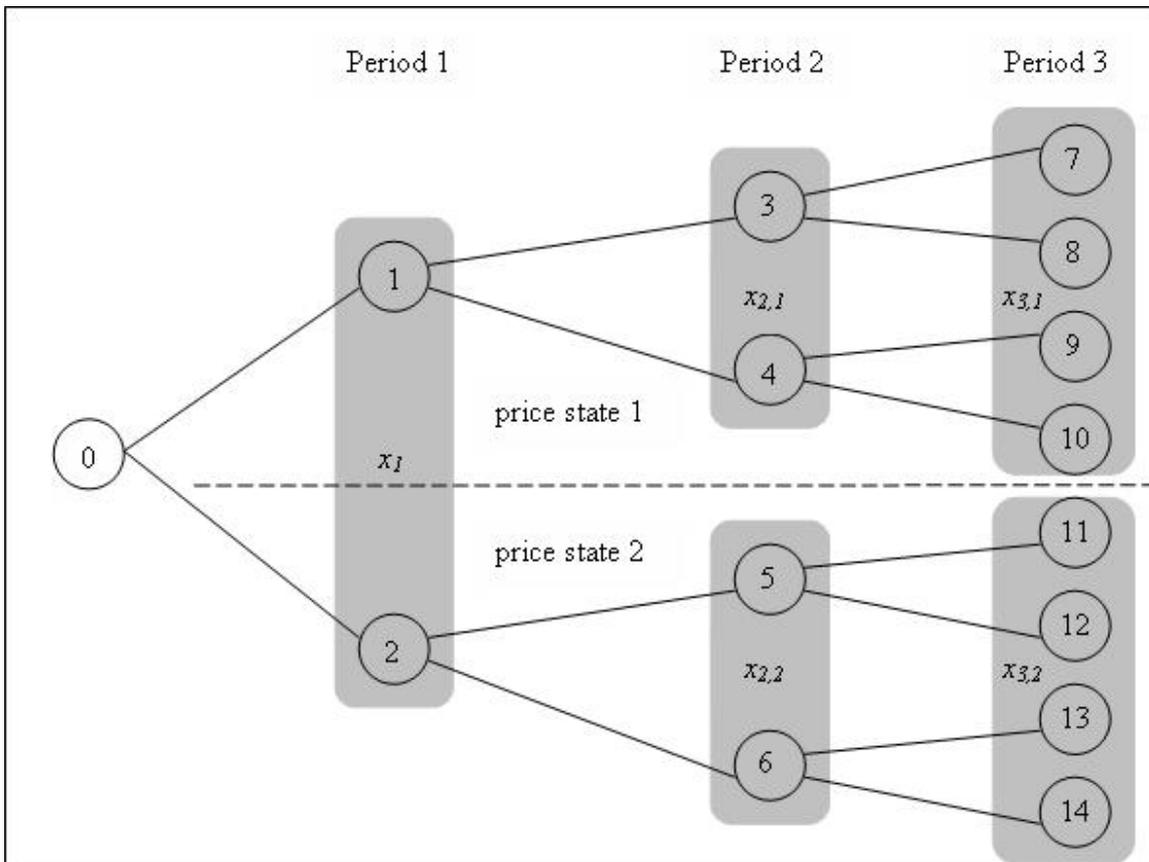


Figure 5. Scenario Tree Representing 3 Stage Stochastic Price Process.

The first decision denoted by x_1 is how much fuel to allocate to Period 1. We make this decision at Node 0 in the scenario tree. As time elapses we will either transition into a price state defined by Node 1 or a different price state defined by Node 2. But our decision must be made before we have this information. If we find we have transitioned to Node 1, we then have a decision to make about how much fuel to allocate in Period 2. This is defined by the variable $x_{2,1}$ and precedes a transition to either Node 3 or 4. Similarly if we find ourselves at Node 2 we make decision $x_{2,2}$ which will apply to Period 3 at either Node 5 or Node 6. The reason decisions $x_{3,1}$ and $x_{3,2}$ each apply to 4 nodes is that having made 2 out of our 3 fuel allocations from a finite supply, the third decision is simply to use whatever is remaining.

Depending on our allocation decisions, we may have 2 distinct resulting load profiles after we have solved the problem depicted in Figure 5. One corresponding to the decisions $x_1, x_{2,1}$ and $x_{3,1}$ and the other corresponding to the decisions $x_1, x_{2,2}$ and $x_{3,2}$. Our expected cost is given by the probabilistically weighted costs of these 2 load profiles. It is this expected cost that we seek to minimize when solving the Stochastic Peak Shaving Problem.

Unfortunately the PSA that was optimal in polynomial time for the deterministic problem cannot be applied to the stochastic problem with similar success. The chief reason for this comes from the problem of determining which periods, out of those tied with the M th highest load, to allocate fuel to. In the deterministic case there was a simple rule given by step 4) of the PSA for solving this problem. In the stochastic case the problem of determining which periods to choose is a combinatorial one. An alternative way to solve this problem, which can also be applied to the deterministic problem, is described below.

4.2 Utilizing the Dual Simplex Method

Below is a formulation of the deterministic equivalent of the stochastic problem depicted in Figure 5 with the maximum demand constraints relaxed. Any indices, parameters or variables that have previously been defined have the same definition here.

Indices

i = period: 1,2,3.

j = price state: 1,2.

Parameters

W_1 = probabilistically weighted price in period 1.

$W_{i,j}$ = probabilistically weighted price in period i under price state j (for $i = 2,3$).

C_j = probabilistically weighted maximum demand charge under price state j .

Variables

x_1 = amount of fuel to allocate to period 1.

$x_{i,j}$ = amount of fuel to allocate to period i under price state j (for $i = 2,3$).

md_j = sum of largest 2 loads under price state j .

Model PkShvStoch

- 1) Maximize $W_1x_1 + W_{2,1}x_{2,1} + W_{2,2}x_{2,2} + W_{3,1}x_{3,1} + W_{3,2}x_{3,2} - C_1md_1 - C_2md_2$.
- 2) $x_1 + x_{2,1} + x_{3,1} \leq T$.
- 3) $x_1 + x_{2,2} + x_{3,2} \leq T$.
- 4) $x_{i,j} \leq K_i$ for $i = 1,2,3$ and $j = 1,2$.
- 5) all vars ≥ 0 .

Explanation

- 1) Objective is to maximize expected savings.
- 2) Cannot use more than total amount of fuel in price state 1.
- 3) Cannot use more than total amount of fuel in price state 2.
- 4) Cannot exceed generator capacity in any period.
- 5) Cannot allocate negative fuel and maximum demand cannot be less than zero.

Without the combinatorial number of maximum demand constraints the PkShvStoch model is easily solved. However, the md variables will each be set to zero and clearly this is not correct. We can observe which two periods under each price state are in fact the maximum 2 in terms of load and add the appropriate md constraints. We can then use the dual simplex method beginning with our solution to the relaxed problem, which is now infeasible, and iterating until we obtain feasibility (or optimality in the dual).

For example suppose in price state 1, periods 1 and 2 have the highest loads following generation, while in price state 2 it is periods 2 and 3. We would then add the following constraints to PkShvStoch:

$$6) \quad md_1 \geq (D_1 - x_1) + (D_2 - x_{2,1})$$

$$7) \quad md_2 \geq (D_2 - x_{2,2}) + (D_3 - x_{3,2})$$

We then apply the dual simplex method until these constraints are satisfied.

This method can be applied to stochastic problems of more stages and states and also to the deterministic problem. We continue to add md constraints and apply the dual simplex method until the value of the md variable(s) is equal to the sum of the highest M loads of the load profile resulting from the optimal solution. At that point we know we have solved the fully constrained problem as any additional md constraints would be non-binding.

5 Conclusions

In this paper we have presented an optimization problem for large consumers of electricity with self-generation capacity. We have observed that self-generation with diesel generators is not an economical strategy in New Zealand under normal market conditions. However, if generator fuel with a limited useful life must be kept for security of supply then the solution to the Peak Shaving Problem gives an optimal way of using it. Furthermore, the algorithm can be applied to any form of self-generation where the fuel cost is known.

The deterministic problem can be solved to optimality in polynomial time using the PSA. The stochastic problem can be solved using the dual simplex method. The time taken to implement this method will depend on the size of the problem and how similar the load and price profiles are. If a consumer's periods of highest demand often coincide with the market's periods of highest price, then the problem is easily solved.

Finally, the peak shaving problem is one of an interesting class of optimization problems where additional costs are applied to variables in accordance with how they rank in terms of magnitude in the final solution. There may be other applications of solution techniques for such problems.

6 References

Vector Ltd. *Vector Electricity Line Charges*. Retrieved on July 12 2006 from http://www.vectorelectricity.co.nz/business/line_commercialindustrial.php

Scheduling Product Pairs subject to Changeover Times and Mould Constraints

Julie Jang
Department of Engineering Science
The University of Auckland
New Zealand
jjan030@ec.auckland.ac.nz

Abstract

This paper examines a production scheduling problem that involves identifying feasible pairs of items and scheduling their production on a single machine to minimise the total production time while satisfying existing orders for each item. The production time of a feasible pair depends on the items in the pair, and the changeover time depends on the current pair and the next. Each item requires a mould specific to that item and it must remain in the mould for a specified time during post processing. The number of moulds for each item are limited.

The problem is modelled as a side constrained symmetric prize-collecting travelling salesman problem in AMPL and solved using the CPLEX solver. A user interface has been created in Microsoft Excel to provide the user with a convenient way to access the model. The model has been verified using a number of test data sets. Presently, the time taken to solve large data sets is too long for the model to be used in practice. However, a number of improvements have been identified that could potentially reduce these solution times significantly.

1 Problem Overview

1.1 Production process

The production process involves a machine which is capable of producing two items at the same time. There are three types of items and each of these item types are manufactured in a range of sizes.

Although the machine allows two items to be produced at the same time, not all pairings of items are possible. This is due to the fact that the size of the one item in the pair determines the maximum size of the other item. If one of the items is large, the other item must be small enough to fit next to it on the machine (Figure 1). Therefore, only feasible pairs of items can be considered.

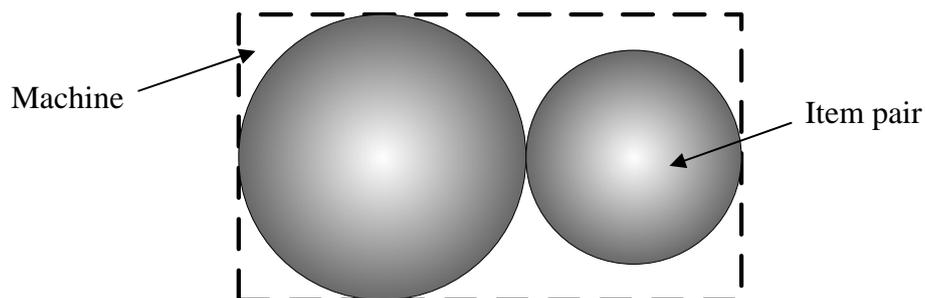


Figure 1 – A diagram showing how the machine is loaded

Each item is produced in a specific mould associated with that item and the number of moulds for each item and size is limited. The item must remain in the mould for a specified amount of time during post processing. Therefore an item can only be scheduled to be placed on the machine, once the associated mould becomes available again for use. The time taken from the moment the mould is taken off the machine up to the point when it can be used again is referred to as the mould turnaround time.

Before the items can be produced, they first need to be placed on the machine. This unloading/loading process involves unloading the pair of items currently on the machine and loading it with the new pair of items. The time to change between the two item pairs is dependent upon the size of the items in the current pair and the new pair and is referred to as the changeover time.

Once the item pair is placed on the machine, the time taken for it to be processed depends on the size of each item in the pair. For each feasible item pair, the minimum processing time for each item is compared and the larger of these two times is used as the processing time.

1.2 Production scheduling

There are many constraints that need to be taken into account when producing a production schedule for this problem. These include hours of operation, the feasible pairings of items on the machine, the available labour resources, the limited number of moulds available for each particular item and the manufacturing rates.

1.3 Model requirements

One of the objectives of this project was to develop a model of the production scheduling problem. The AMPL modelling language was chosen to model the problem and the ILOG CPLEX mixed integer optimiser was chosen as the solver. A user friendly interface was developed in Microsoft Excel to make the process simple and easy to use. This model would be of no use in the industry, if the company employees find it difficult or confusing to use in their everyday operations. Microsoft Visual Basic for Applications (VBA) programming language was used to link AMPL and the user interface.

2 Model formulation

As previously mentioned, only feasible pairs of items must be considered when determining which item pair should be put on the machine next, because not all pairings of items can fit on the machine. The objective is to find a sequence of item pairs to be produced on the machine that will take the least amount of production time, while satisfying the demand constraints.

2.1 Side constrained symmetric prize-collecting TSP

The problem is modelled as a side constrained symmetric prize-collecting travelling salesman problem (Dantzig, Fulkerson & Johnson 1954). Let the set of all items involved in the problem and a dummy item *DummyItem* be denoted by I and let $P \subseteq I \times I$ represent the set of all feasible item pairs for the items in I . A dummy pair, *DummyPair*, represents the start and the end of the production sequence. For each feasible item pair $p \in P$, let there exist a set of instances K_p such that each instance $k \in K_p$ represents one possible production of the item pair p . The number of instances in

each set K_p depends on the maximum possible production of the pair p (i.e. $K_p = \{1, \dots, L_p\}$). The set of vertices in this model is then $K = \bigcup_{p \in P} K_p$.

Let the set of edges E be defined to include all undirected edges between vertices for K_p for all $p \in P$, except that within the set of vertices for each p , only the edges between adjacent (i.e. first and second, second and third etc) vertices in the set are included in E . A production schedule is represented by a subtour in the graph with the above vertices and edges.

It is prize-collecting because the company will not usually produce all item pairs in a single day (i.e. not all vertices will be visited). The model is symmetric, because the time to change from item pair p to q is the same as the time to change from item pair q to p . And the side constraints in the model ensure that the valid production sequence satisfies the demand and does not require the use of unavailable moulds.

The model formulation is:

$$\min \sum_{\substack{k \in K_p \\ p \in P}} M_p x_k + \sum_{\substack{k \in K_p \\ p \in P}} \sum_{\substack{l \in K_q \\ q \in P}} C_{\{p,q\}} w_{\{k,l\}} \quad (1)$$

$$\text{s.t.} \quad \sum_{\substack{k \in K_p \\ p \in P \setminus \{DummyPair\}}} w_{\{DummyPair,k\}} = 2 \quad (2)$$

$$\sum_{\substack{l \in K_q \\ q \in P}} w_{\{k,l\}} = 2x_k \quad \forall k \in K_p, p \in P \setminus \{DummyPair\} \quad (3)$$

$$D_i \leq \sum_{\substack{p \ni i \\ p \in P}} \sum_{k \in K_p} x_k \leq B_i \quad \forall i \in I \setminus \{DummyItem\} \quad (4)$$

$$\sum_{k \in S} \sum_{l \in S} w_{\{k,l\}} \leq |S| - 1 \quad \forall S \in \mathcal{S} \quad (5)$$

$$x_{k_{i+1}} \leq x_{k_i} \quad \forall i \in \{1, \dots, |K_p| - 1\}, k_i \in K_p, p \in P \quad (6)$$

$$\sum_{\substack{k \in K_p \\ p \in P}} \sum_{\substack{l \in K_q \\ q \in P}} \sum_{\{k,l\} \in H} x_{\{k,l\}} \leq |H| - 1 \quad \forall H \in \mathcal{H} \quad (7)$$

$$x_k \in \{0,1\} \quad \forall k \in K_p, p \in P \setminus \{DummyPair\} \quad (8)$$

$$w_{\{k,l\}} \in \{0,1\} \quad \forall k \in K_p, l \in K_q, p \in P, q \in P \quad (9)$$

The variable x_k represents whether vertex k is visited and the variable $w_{\{k,l\}}$ represents whether the edge between vertex k and vertex g is traversed in the production sequence.

The parameter M_p defines the processing time taken by pair p on the machine and $C_{\{p,q\}}$ defines the changeover time between pair p and pair q (i.e. from when p (or q) is taken off the machine until just after q (or p) has been placed on the machine). The parameter D_i defines the demand for item i for a given time period and B_i represents the maximum possible number of each item i that can be produced in a given time period (i.e. A), given by the formula:

$$B_i = \left\lfloor \frac{AG_i}{T_i} \right\rfloor$$

where G_i is the total number of moulds available for each item i and T_i is the mould turnaround time associated with each item i .

The objective function (1) is to minimise the total production time, which is the sum of the processing times for each pair and the changeover times associated with the

production sequence. The constraint (2) and (3) are the degree constraints, where (2) ensures that the tour goes through the dummy node and (3) ensures that if vertex k is used, then there are exactly two edges that have k as one of their end-vertices. Constraint (4) ensures that the demand for each item I is met and the number of item I produced does not exceed the maximum possible number for that item. In constraint (5), \mathcal{S} represents the set of all possible subtour elimination constraints (SECs) and S represents the set of item pair instances (i.e. vertices) corresponding to each SEC. Constraint (6) represents a symmetry reducing constraint which ensures that k_{i+1} , the $(i+1)^{\text{th}}$ vertex of K_p , is only visited if k_i is also being visited in the solution subtour. The index i used for this constraint does not include $|K_p|$, because the last instance vertex of each item pair p does not have any instance vertices after it. In constraint (7), \mathcal{H} represents the set of all possible path elimination constraints (PECs) and H represents the set of edges corresponding to each PEC. This PEC bans all infeasible paths in the set \mathcal{H} from the solution sequence.

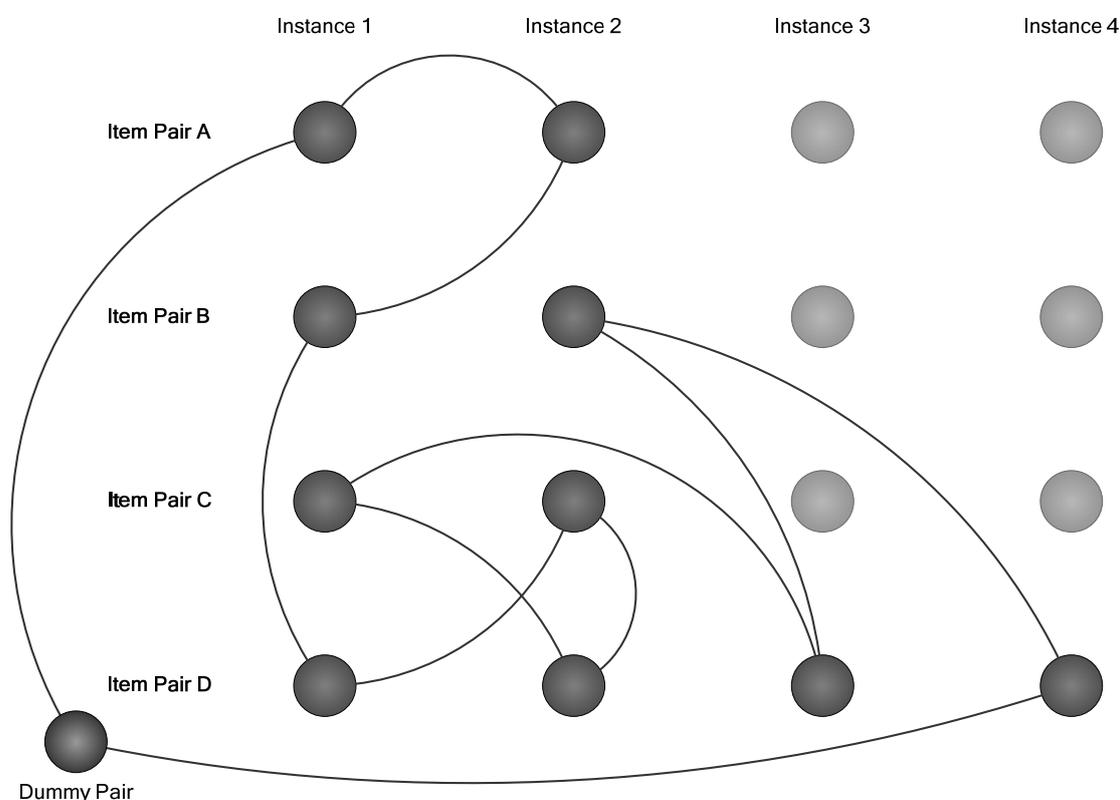


Figure 2 - An example solution

Consider an example case involving four different item pairs A, B, C and D. Figure 2 illustrates a possible solution to the model for this case that represents the production sequence A-A-B-D-C-D-C-D-B-D (or its reverse sequence). In this example, the maximum possible number of each item pair that can be produced is four and the filled circles represent the instances of each pair that were used in the solution. The edges can only be traversed once. For the production sequence A-A-B-C-D-C-D-B-D, the second instance of pair C is not produced before the first instance of pair C. The symmetry reducing constraint (6) does not guarantee strict precedence and so the solution can contain some symmetry. The symmetry reduction constraint (6) removes as much symmetry from a symmetric TSP model as possible. To completely remove the

symmetry, a more complicated precedence constrained asymmetric TSP model would need to be used (Balas, Fischetti & Pulleyblank 1995).

3 Model implementation

For a given set of demands for items and information regarding the feasible pairs of these items, a relaxation of the model was formulated in AMPL. All sets, parameters and variables are declared along with the objective function and constraints in the AMPL model file. The actual data values for the sets and parameters are stored in the AMPL data files.

In the full model shown in the previous section, the SECs (5) and PECs (7) correspond to all possible subtours and infeasible paths in the problem, respectively. However, our solution approach is to only apply a subset of the SECs and PECs as they are needed. This is explained in greater depth later in this section. The motivation for using this approach is that adding all of the constraints to the formulation would limit the size of the problems that we could solve. The SECs and PECs used in the relaxed model formulation are shown below, where $\mathcal{S} = \bigcup_{S \in \mathcal{S}} S$ represents the set of a given number of subtours and $\mathcal{H} = \bigcup_{H \in \mathcal{H}} H$ represents the set of a given number of infeasible paths.

$$\begin{aligned} \sum_{k \in S} \sum_{l \in S} w_{\{k,l\}} &\leq |S| - 1 && \forall S \in \mathcal{S} \\ \sum_{\substack{k \in K \\ p \in P}} \sum_{\substack{l \in K \\ q \in P}} \sum_{\{k,l\} \in H} x_{\{k,l\}} &\leq |H| - 1 && \forall H \in \mathcal{H} \end{aligned}$$

The SECs and PECs are defined as lazy constraints in AMPL. Lazy constraints are used when the modeller knows the constraints are unlikely to be violated and therefore, they can be applied lazily or only as required (ILOG 2006). The SECs and PECs are made into lazy constraints to improve the efficiency of the model.

A user interface was developed in Microsoft Excel and VBA was used to perform the actions of a cutting plane algorithm (Nemhauser & Wolsey 1988). The VBA programming language within Excel is used to link the AMPL model to the user interface and to perform various subtasks involved in solving the complete production planning problem. A flowchart of the entire solver operation is illustrated in Figure 3. This section focuses on the steps taken to add the individual SECs and PECs to the model.

Because the model is formulated to only include SECs and PECs as required, the initial solve of the model will not have any of these two sets of constraints. The production sequence found from this initial solve will satisfy the degree constraints and the demands will be met. However, in most cases there will be more than one subtour present in the solution and unavailable moulds may also be used. Therefore the violated SECs and PECs will need to be added to the new relaxation of the IP formulation and resolved until a solution is found which does not violate any SECs or PECs.

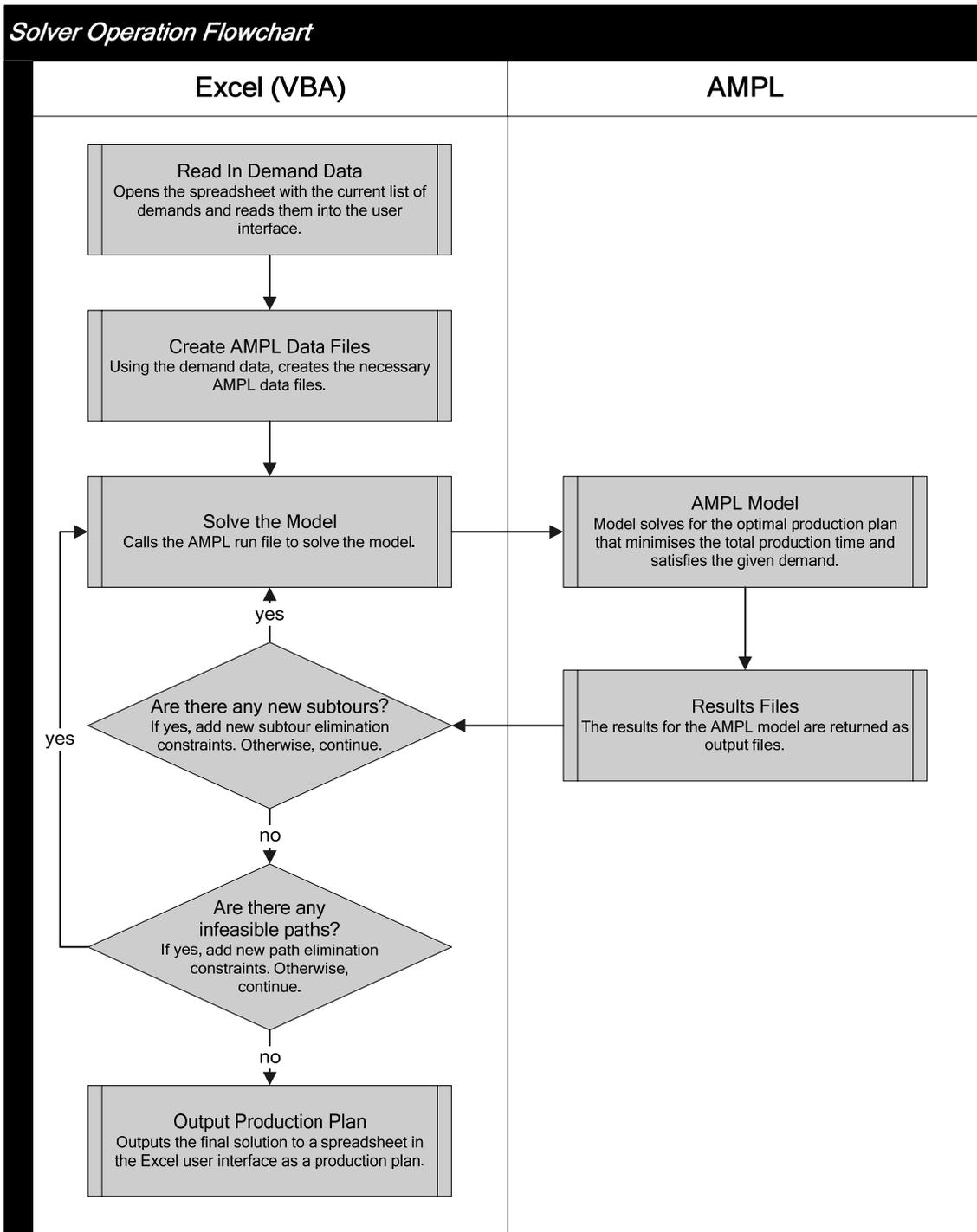


Figure 3 - Flowchart of the solve process using Excel and AMPL

3.1 Subtour elimination constraints (SECs)

After each solve of the AMPL model, VBA checks the solution to see if it contains multiple subtours. It does this by tracing along a path in the solution starting from one of the edges incident to the dummy pair *DummyPair* until the path reaches the other edge incident to the *DummyPair*. The vertices in this path are recorded and if the number of vertices in the path is less than the total number of vertices that are being visited in the solution, then it means that there are multiple subtours in the current solution.

If there is more than one subtour in the current solution, VBA updates a separate AMPL data file to include the vertices that are active but not included in the subtour,

which includes *DummyPair*. Once this is done, the AMPL run file is called again and the model is resolved until no subtours that do not involve the *DummyPair* are detected in the solution. Otherwise, there is one subtour that involves the *DummyPair* and VBA can then carry out a check for infeasible paths.

3.2 Path elimination constraints (PECs)

When the current solution does not contain multiple subtours, the next step in the process is to check for infeasible paths. In this problem, the infeasible paths are parts of the production sequence that are not possible in reality due to the limited number of moulds available for each item. A path in the sequence, which requires the use of an item mould when there are no moulds available, is infeasible and must be banned from the solution using an appropriate path eliminating constraint.

The infeasible paths are detected in VBA by tracing along the production sequence from one of the edges incident to the *DummyPair* and updating a time counter each time an item pair is produced. In each instance, the time counter increases by the sum of the changeover time and processing time associated with the item pair (i.e. $C + M$). The matrix used to keep record of the time when each of the moulds for each item will become available again is also updated. Then for each item pair in the sequence, VBA compares the current value of the time counter variable with the matrix containing the availability times of the moulds for the items in the pair.

If the time counter is less than either or both of the two available times, it means an infeasible path has been found. The infeasible path begins from the point in the production sequence when the unavailable mould of the item was first used (see Figure 4). VBA updates a separate AMPL data file to include this newly detected infeasible path. The model is then resolved with new PECs added and the process of checking for subtours and infeasible paths is repeated again until there are no more new subtours or infeasible paths detected in the AMPL solution.

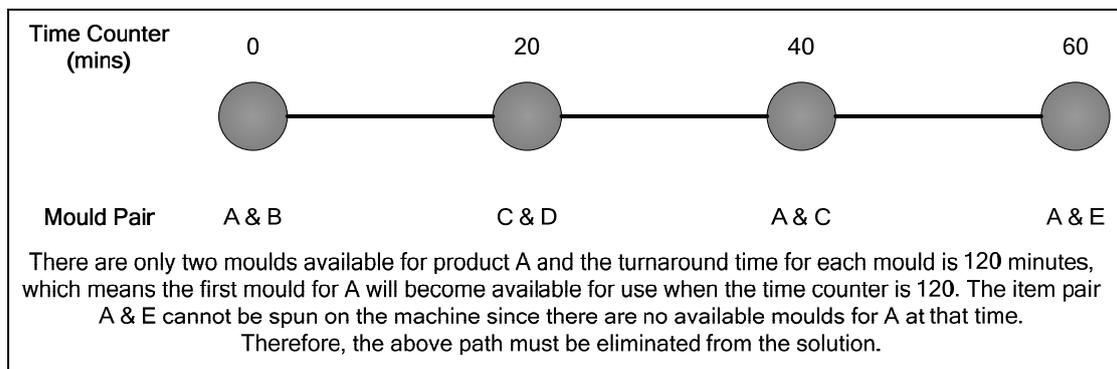


Figure 4 – An example of an infeasible path

However, if infeasible paths are detected, then the current solution is an optimal feasible solution and VBA can carry out the final task of outputting the production sequence to the user interface.

4 Results and discussions

In order to find out whether the model was capable of solving the daily production problem on a typical day at the company, the production data for a typical day is used as the demand input into the user interface (Demand 1). A summary of the problem using Demand 1 is given in Table 1. The problem for the demands in Demand 1 takes 18

hours and 8 minutes to solve. A reason why the problem takes so long to solve is because each relaxation of the model is solved to optimality. CPLEX's heuristics find good near optimal solutions very quickly. Thus, the long running time is likely to be due to the time it takes to prove optimality. The problem is solved using the default CPLEX settings. Tuning the CPLEX solver to the model by modifying some settings is likely to reduce the solve time.

<i>Demand Data #</i>	1	2	3	4	5	6	7
<i>Number of Items</i>	6	7	6	5	5	4	7
<i>Number of Feasible Item Pairs</i>	14	25	15	10	10	6	25
<i>Number of Vertices</i>	161	196	120	49	89	58	125
<i>Total Number of Items in Demand</i>	55	42	34	21	38	31	34
<i>Average "Demand / Number of Moulds" Ratio</i>	1.57	0.93	1.03	0.82	1.17	0.92	0.74
<i>Number of Variables</i>	12472	18846	6981	1204	3807	1561	11341
<i>Number of Constraints</i>	321	382	238	99	179	119	294
<i>Number of Nonzero Constraints</i>	25496	38362	14348	2554	7852	3276	23122
<i>Objective of First LP Relaxation</i>	510.5	380.5	305	209	357	280.5	310
<i>Objective of Optimal Integer Solution</i>	518	381	305	209	357	289	310
<i>Number of SECs</i>	18	8	10	6	3	2	0
<i>Number of Infeasible PECs</i>	66	0	5	3	5	0	0
<i>Total Solve Time</i>	18 hrs 8 mins	8 mins	6 mins	3 mins	3 mins	1 min	< 1 min

Table 1: A summary of the seven data sets

In order to allow comparability between other data sets, a ratio referred to as the average "demand / number of moulds" ratio is calculated by first working out the proportion of the number of each item being demanded to the total available moulds for that particular product, and then averaging these individual ratios. This ratio gives an indication of how many extra moulds are required for the data set. Since the ratio is 1.57 for Demand 1, it means that the data set requires approximately 57% more moulds than there are available for each item.

The next step is to investigate how the model behaves for other test data sets which vary in problem size and demand constraints. These data sets were generated in order to compare the different problem characteristics with the change in the model solve behaviour.

The figures in Table 1 show that generally for slightly smaller data sets, the problem takes a much shorter time to find the optimal production plan. However, it is difficult to find any form of correlation between the solve times of the data sets. The number of SECs and PECs involved in a data set does however show a slight correlation with the solve time. This is reasonable because each time a new subtour is detected, the model is resolved to optimality in AMPL and this takes time.

It is interesting to note that Demands 3, 4, 5 and 7 have no integrality gap (i.e. the objective of the first LP relaxation is the same as the objective of the optimal integer solution). And the integrality gaps for the Demands 2 and 6 are very small. This means that the inclusion of the SECs and PECs did not alter the optimal total production time by much, if at all.

5 Conclusions and future directions

This production scheduling problem involves finding an optimal production sequence of item pairs to be produced such that the demands are satisfied and the limit in the

number of available moulds are taken into account. In this investigation using a side constrained symmetric prize-collecting TSP model to solve the problem, the following conclusions were reached:

- Modelling the problem using a side constrained symmetric prize-collecting TSP model provides an accurate representation of the structure of the actual problem and the constraints involved.
- The model is capable of finding the optimal production sequence to smaller versions of the real demand data set in a reasonably short length of time. However, it takes a lot longer for larger data sets involving greater demand in a wider range of items.

Given these conclusions, it is important to note how the project could be expanded in the future to improve the modelling of this production planning problem.

5.1 Improve implementation

In order to improve the model's efficiency by increasing the speed at which the problem can be solved, the current model could be implemented in a branch-and-cut framework. Also, the model should be modified to allow the passing of solution and bound information from one relaxation to the next, in order to improve the model's efficiency.

5.2 Truncated tree search

Another possible method of improving the current model would be to slightly change the way the problem is being solved in AMPL. Currently, every time AMPL solves a relaxation of the model (with or without the additional SECs or PECs), it solves to optimality. This means that the branch and bound tree is completely explored each time. An advantage of this method is that once we find an optimal solution with no more new subtours or infeasible paths, we can be 100% certain that this solution is the final optimal production sequence. However, solving each relaxation to optimality is very time consuming. An alternative method would be to exit the branch and bound prematurely once a solution to the relaxed problem has been found. When a solution which does not contain any multiple subtours or infeasible paths is found using this new method, it does not necessarily mean that it is optimal. In order to find the optimal solution to the problem, the model will have to be resolved beginning from the current integer feasible solution. If the next integer feasible solution for the relaxed problem contains any subtours or infeasible paths, these will need to be added and the process repeated. Once an integer feasible solution to the relaxed problem is found that is optimal, this solution is also optimal to the complete problem. Therefore, it provides an optimal production sequence to the problem.

This truncated tree search approach normally limits the time, number of nodes explored, branch-and-bound tree depth, the number of integer feasible solutions found, or some combination of these.

6 Acknowledgments

I would like to thank my project supervisors, Dr Stuart Mitchell and Dr Hamish Waterer, for all the guidance and support they provided me with throughout this project. Without their academic insight, this project would not have been possible.

I would also like to thank the company staff for not only giving me the opportunity to work on this project, but also for the time they spent gathering information and discussing the problem with me.

7 References

- Balas, E., Fischetti, M., and Pulleyblank, W.R. (1995) The precedence-constrained traveling salesman polytope. *Mathematical Programming*, **68**, 241-265.
- Dantzig, G.B., Fulkerson, D.R., and Johnson, S.M. (1954) Solution of a large-scale travelling salesman problem. *Operations Research*, **2**, 393-410.
- ILOG. (2006) *ILOG CPLEX 10.0 User's Manual*.
- Nemhauser, G.L., and Wolsey, L.A. (1988) *Integer and Combinatorial Optimization*. John Wiley & Sons Inc.

Rogaining: a Prize-Collecting Orienteering Problem

Samuel P. Gordon
Department of Engineering Science
University of Auckland
New Zealand
sgor029@ec.auckland.ac.nz

Abstract

Rogaines are prize-collecting sporting events, in which competing teams must navigate on foot between a subset of locations where they can collect points (“controls”). The objective in a rogaine is to maximise the total number of points collected within a defined time limit, by choosing which controls to visit and in what order. This is an optimization challenge, but teams are not allowed computer equipment. Instead they must plan out their routes based on some form or combination of race strategies.

This paper establishes strategies that are effective in rogaining events. An enumeration algorithm was created and implemented in C++ to model the rogaining problem, constructing routes by adding one control at a time to a feasible sequence of visited controls. By applying strategies to this process and analysing the quality of the resulting routes, a set of validated strategies was able to be built up. Past rogaine courses from the Auckland region were used as data sets for this problem.

1 Problem Description

Rogaines are prize-collecting sporting events. Originating in Australia in the 1970’s, they are now run internationally and are growing in popularity. Competing teams must navigate on foot between locations that are worth a set amount of points (“controls”), in order to maximise the total points gained within a defined time limit. Controls are spread over many square kilometres, and are worth different amounts of points as indicated by their reference number (e.g. a control labelled 41 is worth 40 points). Rogaines are similar to orienteering events but run over a longer time period: official world rogaine championships are 24 hours long with the shortest recognised rogaines being 3 hours in duration.

The rogaining problem is complicated. It is not possible to visit every control, even for a team of high fitness, and a penalty of 10 points per minute is imposed on teams that finish late. Although teams determine their own route, they must start and finish in the same location (marked by a triangle within a circle as in Figure 1). Some rogaines will have restrictions on areas that are out of bounds, but otherwise any existing track can be used to travel between controls. Each team can only collect the points at any control once; revisiting a control will not result in an increase in points.

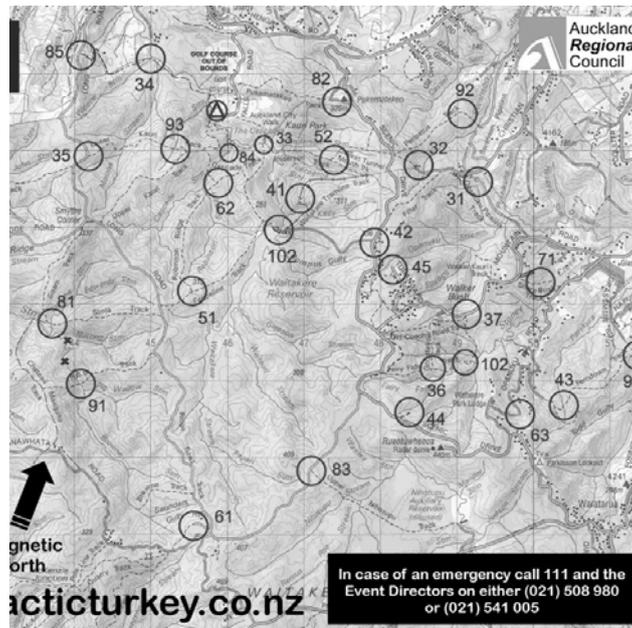


Figure 1. Course map for 2005 Piha Rogaine

Rogaine rules explicitly prevent the use of computers in route planning, as well as other electronic tools – the only aids allowed are a compass and a course map showing the location of tracks and controls (the latter indicated by circles as in Figure 1). Teams must plan out their route by hand, basing their route choice on some form or combination of strategies. Knowing which strategies are worth using can significantly increase a team's chances in the competition.

2 Model-building

2.1 Assumptions

In order to model the rogaining problem the following assumptions were made:

- No route would be considered that took a team over the time limit.
- Teams travel at constant speed throughout the duration of the rogaine, unaffected by changes in terrain and elevation.
- Teams stay on existing tracks. This means the path between each pair of controls is pre-defined.
- A path between two controls takes the same amount of time regardless of the direction of travel.
- Teams lose no time in searching for control markers or taking rest breaks during the race.

These were considered reasonable assumptions and made the problem simpler to model. However, the assumption of constant speed is a considerable one, and was based on lack of information on running pace over varying terrain. This is discussed further in section 8.

2.2 Data Set

Two past rogaine courses in the Auckland region (Piha 2005 rogaine, Piha 2006 rogaine) were used as data sets for this problem. Each of these rogaine courses incorporated both a 3 hour and 6 hour event. Course information was provided through the Optima Corporation.

The visualisation tool Race Reviewer was also provided through the Optima Corporation. This software had been specifically designed to plot rogaine routes onto a

rogaine course map, as well as display the distance covered and total points collected (Figure 2). It was used for visually analysing route characteristics.

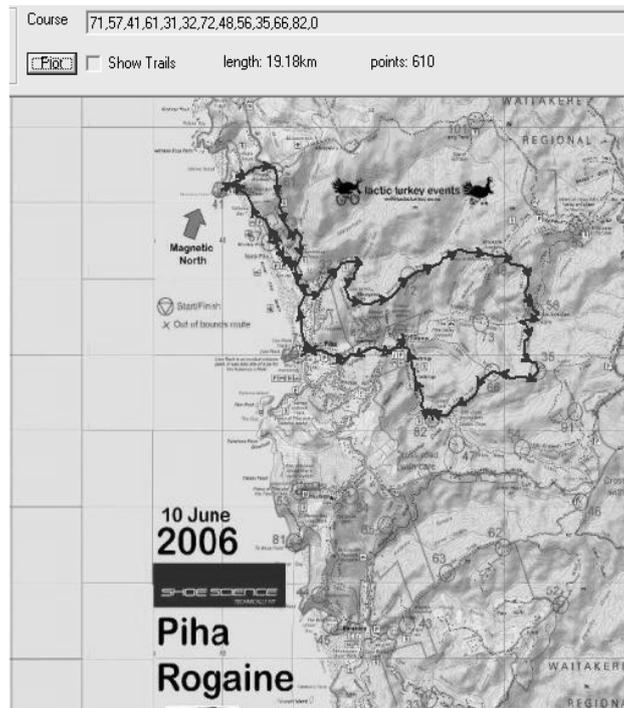


Figure 2. Race Reviewer showing route of a team in the 2006 Piha rogaine

Over 400 team results from these past rogaines were compiled. Using the distribution of speeds from these results, a set of speed intervals was created that represented different fitness levels (Table 1, Table 2). The distance a team is able to travel affects the number of checkpoints they can visit, and so the maximum number of points they can collect.

Designated fitness level	Speed interval (km/hr)
Low	3 - 4.5
Med	4.5 - 6
High	6 - 8

Table 1. Fitness levels based on rogaine results

Distance	Achievable by
15km	Medium fitness team in a 3hr event
20km	High fitness team in a 3hr event
30km	Medium fitness team in a 6hr event
40km	High fitness team in a 6hr event

Table 2. Relating fitness and time limit to achievable distance

3 Approach

3.1 Method choice

Well-established methods exist for solving similar problems in operations research literature. Such problems include the Travelling Salesman Problem (TSP) as introduced by Dantzig, Fulkerson and Johnson (1954), Vehicle Routing Problem as introduced by Dantzig and Ramser (1959), and Orienteering Problem as introduced by Tsiligirides (1984). However, most solution methods for these problems are too complex to allow for testing of simple strategies. Also, these formulations do not allow for locations to be revisited, and in the basic TSP and VRP problems it is not possible to only visit a subset of locations.

Enumeration was chosen as the solution method for the rogaining problem. Of the methods examined, it is the only one that can be implemented by hand and so used by a rogaine competitor. By utilising the fact that routes are defined by a sequence of visited controls, enumeration constructs routes by adding one control at a time to an existing (partial) sequence. Its main advantage is enabling strategies to be tested by controlling how each consecutive control is selected, although it is also easy to implement. A limitation is that it only has information about a small segment of the rogaine course at any one time.

3.2 Use of TSP formulation

Although a TSP formulation cannot be used to test simple strategies, the rogaining problem can still be solved using a modified form of TSP. This enables the routes obtained through the enumeration to be checked for optimality. The altered formulation required the distance between every pair of controls to be known, which was achieved using the Floyd-Warshall algorithm as described by Floyd (1954). The formulation was as follows:

Indices

- v = control
- v' = control

Parameters

- V = set of all controls
- p_v = points available from control v
- T_{max} = time limit
- $T_{v,v'}$ = time to travel from control v to control v'

Decision variables

- X_v = whether control v has been visited (binary variable)
- $x_{v,v'}$ = whether the path between control v and control v' is part of the route (binary variable)

Model TSPcompare

- 1) Maximize $\sum_{v \in V} p_v X_v$
- 2) $\sum_{v, v' \in V} x_{v,v'} = 1$ for all v, v' in V
- 3) $\sum_{v', v \in V} x_{v',v} = 1$ for all v', v in V
- 4) $X_v = 1 - x_{v,v}$ for all v in V
- 5) $\sum_{(v,v'): v \in U, v' \in U, v \neq v'} x_{v,v'} \leq |U| - 1$
- 6) $\sum_{v \in V, v' \in V} t_{v,v'} x_{v,v'} \leq T_{max}$

Explanation

- 1) Objective is to maximize total points collected.
- 2) Each control must be arrived at once as part of a route.
- 3) Each control must be departed from once as part of a route.
- 4) Allow self-loops on controls that are not visited as part of the main route.
- 5) For all subsets U of controls, at least one control is linked to a control outside the subset (no subtours).

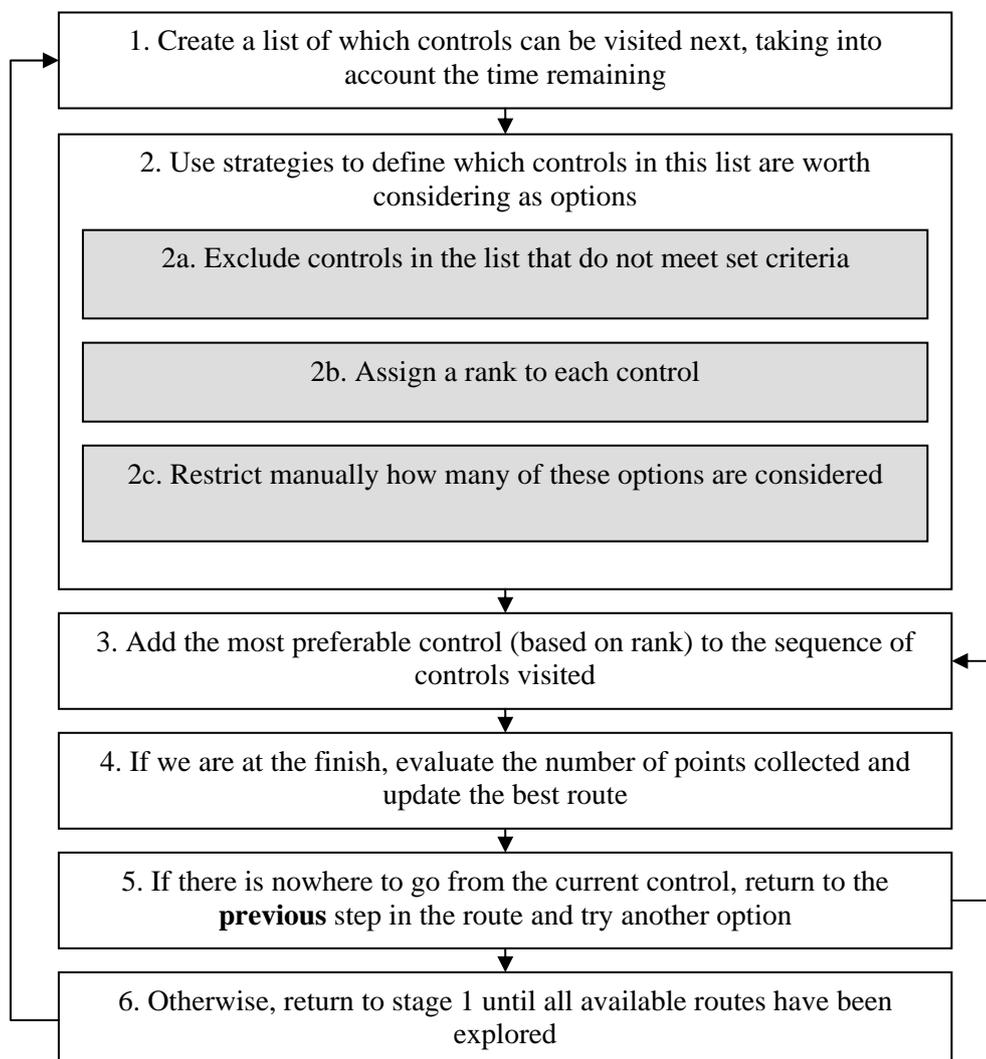
6) The route must not take longer to complete than the time constraint.

4 Enumeration Algorithm

The algorithm to solve the rogaining problem was created in C++ and uses text-based information as input. The section that constructs and compares routes is broken down into six stages that repeat, as in the diagram below, with strategies being implemented in stage 2. Before the route-building section begins, the following information is known:

- Time limit.
- Speed/fitness level of team.
- For each control: the number of controls that can be reached from it directly, the time taken to reach each of these, and the shortest time to return to the finish control.

Route-building section of algorithm



Ranking is an important way of testing strategies, and this is implemented in stage 2b as above. To determine which control is the best one to visit next, each option is assigned a ranking based on some criteria. Controls given a ranking of 1 are seen as the most preferable to visit. The following simple criteria were tested:

- Highest no. of points.

- Longest length of time away.
- Shortest length of time away.
- Highest ratio of points to time away.

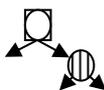
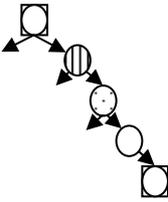
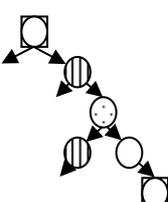
Routes that contain high-ranked controls are explored before routes that contain lower-ranked controls. For example, the very first route explored contains only controls ranked 1. Routes that contain controls ranked 2 are not explored until later, and so on.

Restrictions tested in stage 2a and 2c took the following forms:

- Ignoring controls that will lead to a route taking longer than the time limit. This was done by comparing the time remaining to the shortest time from the current control to the finish.
 - Ignoring controls further away than some specified distance.
 - Limiting revisiting of controls (backtracking).
 - Ignoring routes predicted to collect fewer points than the current 'best' route.
- For any control, the sequence of controls on its shortest path to the finish was known. If the time remaining was close to the shortest time to get back, the total score achievable was able to be calculated and compared to the current best route. Ignoring these routes saved time in the enumeration process.
- Only constructing routes that contained high-ranked controls (e.g. ignoring controls assigned a ranking lower than second). This is a closer reflection of implementing strategies by hand; a rogainer will only be able to consider a limited number of possibilities.

5 Comparing strategies

To compare the effectiveness of various strategies, two key measures were used. The first measure was the total number of 'iterations' performed before the best route was found, as explained in Table 3. If the strategies used are good, the algorithm will not need to construct many routes before it finds a good one, and the number of iterations will be low as a result.

Course	Representation of algorithm trying possibilities	Comment	Iterations performed
		There are two possibilities from the start control. The algorithm selects one of these possibilities as the next step in the route sequence and updates route information accordingly.	1
		This process continues until (in this case) a route is complete.	4
		There were no options available from the control last visited. The algorithm returns to an earlier step in the route to try another possibility, and updates route information for this new (partial) route.	5

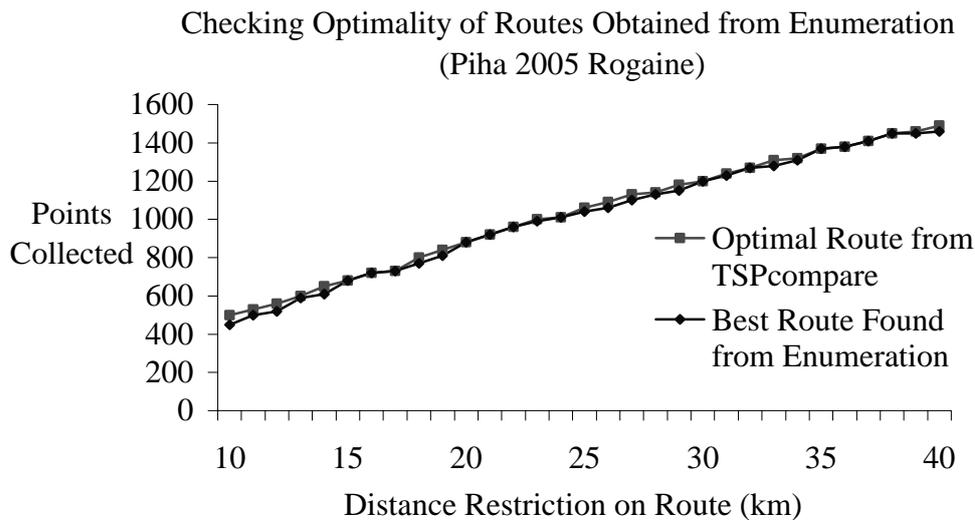


Figure 3. Showing optimality of best routes found (Piha 2005)

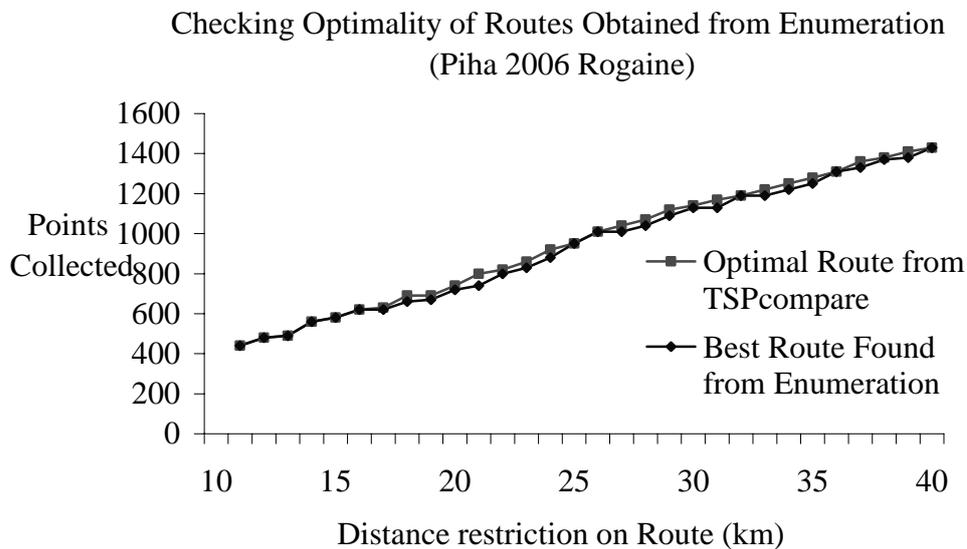


Figure 4. Showing optimality of best routes found (Piha 2006)

If only the top two ranked controls are considered at each step in a route, selectively applying strategies changes the quality of the best route found. Table 5 shows this effect for representative routes.

Rogaine	Distance restriction	All strategies implemented	Considering only 1 st ranked choices	Considering 1 st and 2 nd ranked choices, only sorting by ratio	Considering 1 st and 2 nd ranked choices, with no distance restrictions
Piha '05	20km	880	750	840	880
	40km	1460	1260	1450	1460
Piha '06	20km	720	660	710	720
	40km	1430	1270	1400	1400

Table 5. Points gained through selective application of strategies

6.2 Discussion

The strategies that worked well were not complicated. It is not unexpected that choosing controls based on the ratio of points to time proved to be effective. The overall objective is to maximise the total points gained in the time allocated, so aiming to do this at each step in the route is reasonable. What is more unusual is the effectiveness of choosing the first two controls based on being the closest. Although unexpected, this could be due to the fact that early choices have a large impact on route possibilities later on, and choosing controls that are close leaves more time flexibility later.

One difference between the rogaining problem and other operations research problems is the flexibility to revisit controls (backtracking). However, none of the best routes achieved through the enumeration involved backtracking. This suggests that for the purpose of finding optimal routes, the rogaining problem can be modelled using an existing formulation.

6.3 Application of principles

Using good strategies can make a significant difference. In June 2006 the author competed in the 2006 Piha rogaine as part of a three person team, choosing a route that collected 610 points as in Figure 5. Although the team placed 6th out of 24, adopting the strategies tested in this project would have resulted in a route that collected another 110 points (as shown in Figure 6). This would have brought us into the top three finishers in our division.

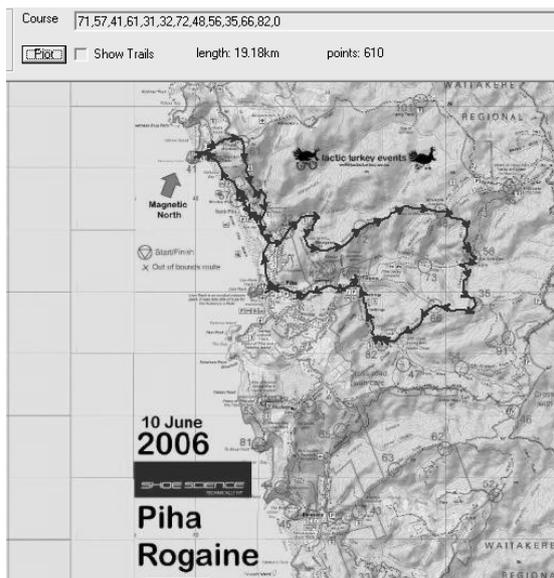


Figure 5. Author's attempt: 610 points

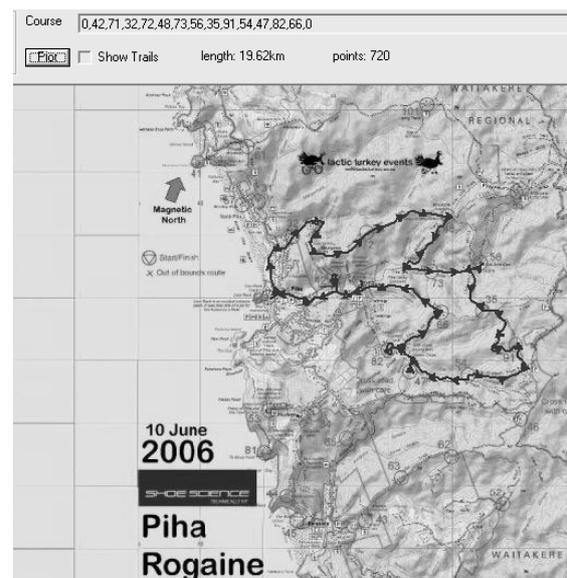


Figure 6. Best route using validated strategies: 720 points

7 Conclusions

This work created an algorithm to compare and validate different route-planning strategies for rogaining events. The set of strategies validated in this project is robust enough to be applied to a new rogaining course, although further modelling work needs to be done to accurately account for the effects of terrain and elevation differences. The following strategies are likely to lead to a good route:

- Base route choice around how far the team is likely to travel (considering their fitness level)
- Ignore controls that will take more than 1/6 of the time limit to get to
- The first two controls in the route should be selected on the basis of being the closest

When choosing the next control in a route, teams should use the following criteria:

- If the current time is less than 40% of the time limit, only consider controls further away from the start or up to 900m back towards it
- If the current time is between 40% and 80% of the time limit, only consider controls that are a similar distance from the finish (within 2km)
- If the time is greater than 80% of the time limit, only consider controls that are within 3.5km of the current control
- Base the choice of where to go next on the controls with the highest ratio of points to distance
- Consider at most the top two preferred options using the above strategy
- Do not re-visit a control unless it is the only way to get back to the finish within the time limit

8 Limitations and future work

Each rogaine has slightly different characteristics, so it is possible that situations exist in other rogaines where the strategies mentioned in this work will fail. This is a limitation of the number of data sets available for the algorithm to be tested on. While the strategies presented here seem to be robust, testing on more data sets is needed in order to confirm their validity.

To test the assumption of constant speed, the effect of terrain was included in the model after a set of strategies had already been validated. Course information was altered by multiplying the distance between two controls by a factor proportional to their difference in elevation. The direction of travel between two controls was now important: going uphill would take longer than going down. Preliminary tests showed existing strategies still producing good routes (within 20% of optimality), although some unusual results were achieved and there was not time to explore these further. Further testing is necessary to confirm how terrain influences strategy.

It is uncertain how well the recommended strategies would perform for longer rogaines. World championship events are run over a 24 hour period and often have no existing tracks between controls. The number of route possibilities for these events would be much larger than the 3 hour events, making enumeration a more time-consuming method to use. Finding effective strategies for these events is an area of possible future work.

9 Acknowledgements

I would like to thank my project supervisors Professor David Ryan and Dr Stuart Mitchell for their guidance and input, as well as Jody Snowdon of the Optima Corporation for providing the data on which the algorithm could be tested. I would also like to thank Martin Peat and Nicholas Falconer, for agreeing to compete in a rogaine with me in June 2006.

10 References

- Dantzig, G.B., D.R. Fulkerson., and S.M. Johnson, 1954. "Solution of a large-scale travelling-salesman problem." *Operations Research* **2**:393-410.
- Dantzig, G., and Ramser R., 1959. "The Truck Dispatching Problem." *Management Science*, **6**:80-91.
- Floyd, R.W., 1962. "Algorithm 97: Shortest Path". *Communications of the ACM* **5**(6):345
- Tsiligirides T., 1984. "Heuristic Methods applied to Orienteering." *Journal of Operational Research Society*, **35**(9):797-809.

Reflections on Forty Years of Researching and Implementing Inventory Management Systems in Commercial Firms

Alan J. Stenger

Ports of Auckland Visiting Professor of Logistics and Supply Chain Management

The School of Business and Economics

The University of Auckland

Private Bag 92019, Auckland Mail Center, Auckland 1142

a.stenger@auckland.ac.nz

Abstract

The author has been researching, as well as helping to implement, inventory management systems in association with variety of firms over the last 40 years. This paper traces some of the interesting changes that have taken place in both the inventory planning tools available and the receptivity of managers in North American firms to using these tools over this period. We also draw conclusions about the challenges that remain in furthering the use of such tools in the future.

Keywords: Inventories; Inventory Planning; Practice of Inventory Management; JIT; History of Inventory Management

1. Introduction

Gradually over the last few years, I have come to realize that I have been doing inventory-related research and consulting for over 40 years. This has happened as part of my involvement in related transportation; distribution; logistics; and now, supply chain issues. I never really have considered myself an “inventory researcher” *per se*—I just kept bumping into inventory problems and began to see issues of application of, and education in, inventory management techniques as critical to designing and operating effective supply chains. In this paper I want to (1) briefly trace my own journey in inventories; (2) relate what I have found to work well in industrial situations and where it came from; and (3) draw some conclusions about the current state of the practice in industry today, offering some suggestions regarding promising directions for future research and educational efforts. My hope is that this may inspire both budding academics in the area, as well as practicing managers to help operate more effective supply chains through good inventory management practices.

2. Journey in inventory theory and practice

I stumbled accidentally into the inventory management field, probably like many others in this discipline. My first taste came in an Industrial Engineering course, taken as part of my MBA degree at the University of Michigan in 1963. We used the text, new at the time, by (Hadley and Whitin 1963) entitled *Analysis of Inventory Systems*. The authors

noted in their preface that the text was the accumulation of 15 years of advances in the field produced by a variety of researchers in academia, as well as in consulting and contract research firms (such as Arthur D. Little and the Rand Corporation). Thus the field was relatively young, aside from the so-called “Wilson” lot-size formula.

2.1. Personal industrial experiences

After receiving my MBA, I took a job with The Dow Chemical Company at the firm’s headquarters in Midland, Michigan. There I once again stumbled on inventories. The company held inventories in numerous distribution centers (for packaged products) and terminals (for bulk products). While operated by independent third parties, these facilities each reported semi-weekly their withdrawals, receipts, and inventory levels by stock-keeping unit (SKU) back to the headquarters, using teletype machines. Dow in Midland received this feedback through special machines that produced punched paper tapes of the data. The firm then fed those tapes into readers, loading the data for computer analysis. The system compared the current inventory positions for each SKU at each location to preset minima and maxima; and generated replenishment orders for the standard replenishment quantities, as well as overstock notices when necessary. Analysts updated the control parameters once per quarter, although they did not use precise, scientific techniques in their calculations.

Later at Dow—1968—I was put in charge of distribution for the consumer products business. As part of this job, I took over supervision of a woman who was using a simple tabular process for generating time-phased distribution center replenishments and production plans (by SKU), driven by a monthly national sales forecast provided by the marketing and sales function.

2.2. Personal academic experiences

In 1969 I left Dow to get a Ph.D. at the University of Minnesota in business logistics and management information systems. Subsequently I was employed by The Pennsylvania State University (Penn State) in 1972 as an assistant professor of business logistics. It was not long before I again faced several inventory problems.

2.2.1. Transportation and inventory trade-offs

Some opportunities arose to study the trade-offs between transportation and inventory involved when choosing the best mode of transportation (rail, highway, air, etc.) to use for an inventory replenishment. With the development of the air mode as a viable means of transporting freight, managers attempted to determine its proper role in logistics. Lewis (1956) proposed the concept of a “total cost analysis” that included measuring the impact and cost of carrying inventories (as transportation modes are changed) as a way to help make this decision. Later, Baumol and Vinod (1970) hypothesized that the demand for various modes of transport was driven by a combination of transportation and inventory costs. They used the Poisson distribution of demand over the transit time for their analysis, which works well for movements of small annual volumes, but not for those of higher volumes.

Recognizing that neither transportation nor inventory decision makers in industry did a very good job of total cost analysis, I began a stream of research on the subject, using the Normal distribution rather than the Poisson (see Stenger, Coyle and Prince 1977; Stenger and Cunningham 1978; Tyworth, Rao and Stenger 1991), and

subsequently developed a personal computer software package for doing the analysis under a variety of conditions and assumptions (Stenger 1985).

2.2.2. Distribution requirements planning

During the 70s I also got involved in a consulting project for the Commonwealth of Pennsylvania's Liquor Control Board (PLCB). This governmental agency had a monopoly on the wholesale and retail distribution of liquor and wine in the state. This began—like many of my consulting projects—as a study of the optimal number and location of warehouses the agency needed. As is usual in such analyses, I endeavored to determine how aggregate inventories might change as the number of stocking points changed. This attempt uncovered the fact that inventories at the warehouse level were managed by intuitive techniques and rather poorly—the agency had both high inventories and high stock-out rates. So we began a new study to determine better ways to manage this inventory. This effort unearthed the fact that the PLCB maintained up to 3 years of demand history by SKU both by store and by warehouse. Being familiar with Forrester (1958) and the concept of demand variance amplification, I recognized the relatively unique opportunity to empirically test the use of downstream data as a means of arriving at better forecasts of upstream demand. The positive results of this analysis led to one of the early distribution requirements planning (DRP) applications (Stenger and Cavinato 1979, and Whybark, 1975).

Of course the DRP initiatives built on the previous work of (Wight 1974) and (Orlicky 1975) in popularizing the concept of materials requirements planning (MRP). However, one of the misunderstandings I have found in industry (and at times in academia) regarding MRP is the assumption that the process manages inventories. In fact it *maintains* inventories by managing/planning the flow of materials given the basic, exogenously determined, inventory parameters supplied for each item (safety stock, lot size, and lead time).

3. My view of the key contributions to the practical management of inventories since 1975

I have already indicated above some of the critical and/or interesting developments in the practical application of inventory management techniques through the 3rd quartile of the 20th Century. Now I would like to point out what I have found to be key developments since that time. Again I will proceed chronologically.

3.1. 1975-80

The latter part of the 1970s saw the stunning introduction of the Toyota Production System in the article by Sugimori et al. (1977). Covering the areas of Just-in-Time (JIT), Kanban, and “respect-for-humans,” this article brought a new paradigm to inventory, as well as production management. Initially disbelieved, then dismissed as applicable only in the Japanese culture, and finally adopted to the point of misuse and abuse; the concepts promoted by Toyota did ultimately lead to the “Lean” approach to manufacturing (see (Womack and Jones 1996)) that has been widely adopted. The critical issues for inventory management addressed by JIT concerned the factors used in computing the inventory control parameters themselves. Previous to JIT, we generally took those factors—setup times and costs, carrying costs, lead times, uncertainties etc, as fixed, exogenous values. JIT forced us to think about aggressively challenging and

mitigating these factors as another way to reduce inventories. In manufacturing this meant radically reducing set-up times; in logistics it meant reducing lead times and/or uncertainties whenever possible. Major inventory reductions resulted when companies implemented these initiatives correctly.

A great deal of progress on software applications also occurred during the later half of the 1970s. IBM evolved the IMPACT system into the Consumer-Oriented Goods System (COGS), which subsequently evolved into the INFOREMS system by the end of the decade. Each took advantage of the rapidly increasing power of computers and their capacities for data storage. Separately R. G. Brown became involved with Scientific Time Sharing Systems (STSC) to implement his models for use by many on a computer-utility basis. (STSC subsequently became Manugistics, which was recently purchased by JDA Software.)

3.2. 1980s

The 1980's brought to fruition an approach often referred to as Vendor-Managed Inventory (VMI) or the Continuous Replenishment Process (CRP). People also first began articulating the concept of a "supply chain" and the management thereof.

3.2.1. Vendor-managed inventories

The idea of VMI is simple: a supplier of products to a company manages the inventories of that supplier's products (purchased and owned by the customer) at the customer's location. Having inventory on consignment had long been a practice used in some industries—the supplier stores and manages inventory at the customer location, but owns the product until the customer uses it. Consignment is basically a cost shifting activity. VMI is somewhat different. The objective in VMI is to reduce the inventories held by the customer, as well as reduce costs for the supplier (vendor)—a win-win situation. One of my former graduate students, Duane Weeks, was the project manager for Procter & Gamble's (P&G's) first foray into this approach, implemented in conjunction with a grocery retailer in St. Louis, Missouri, USA (Koch 2002). The initial effort in 1980 was highly successful in improving the customer's inventory levels and fill rates, but was subsequently discontinued due to the high cost to P&G of faxing data between the two partners and manually entering that data into computers, and the perceived lack of scalability of the process. Later in the 1980s, P&G adopted the approach with mass merchandisers Kmart and Wal-Mart using electronic data exchange (EDI), and subsequently rolled it out with other grocery chains.

3.2.2. Supply chain management

Also in the 1980s, consultants introduced the term "supply chain management." Representatives of the US consulting firm Booz Allen Hamilton, working with Philips Electronics in Europe, sought to gain new economies by looking across and beyond the firm. In their words, the approach "...differs from the traditional approach in four respects: 1. It views the supply chain as a single entity rather than relegating fragmented responsibility for various segments in the supply chain to functional areas, such as purchasing or manufacturing. 2. It depends upon strategic decision making. 3. It provides a different perspective on inventories. 4. It requires an integrative new approach to systems. Supply chain management works to balance the conflicting

objectives of marketing, sales, manufacturing, and distribution.” (Houlihan 1985), p. 51)

Subsequently “supply chain management” (SCM) came to be broader than Houlihan’s original explanation, encompassing both suppliers of suppliers and customers of customers as integral parts of the chain. This broader view provided even more incentive for both academics and practitioners to consider inventories from a multi-echelon point of view. In fact, most uncoordinated supply chains embody significant amounts of redundant inventories, making inventory management activities in firms even more important.

3.3. 1990s

In the decade of the 90’s, efforts to improve the practical management of inventories focused both on refining aspects of what had already been developed (improved forecasting and replenishment, and optimizing the requirements planning process), and on expanding the scope of the analysis to multi-echelon, or supply chain, situations. One major refinement effort resulted in a process known as “Collaborative Planning, Forecasting, and Replenishment” (CPFR). Another produced a set of initiatives that came to be called “Advanced Planning and Scheduling” (APS). Multi-echelon inventory theory was not new, but generally had been applied only in military situations, not in the commercial world. The new supply chain focus put added impetus on making multi-echelon techniques more accessible to inventory managers.

3.3.1. CPFR

The idea of customers and suppliers collaborating to produce better demand forecasts grew out of the VMI efforts in the grocery and mass merchandising industries. VMI was basically passive as far as the customer was concerned—the customer supplier the data and the supplier made the decisions. But obviously the customers had their own marketing and sales initiatives, as well as other objectives that might affect demand and inventories, just as the suppliers did. The CPFR initiative advocated combining input from both parties, which should lead to better plans and forecasts. A working group of consumer-goods manufacturers, retailers, consultants, and software vendors coalesced, and designed a process (very technology-based) for sharing information and developing collaborative plans, forecasts and replenishments—CPFR. The group also sponsored several pilot projects to test the process. Although apparently quite promising and successful, the concept has been slow to catch on. The main problems relate to competitive issues. Often there are only two major suppliers in a given product category, so that competitive moves planned by the second supplier can be inferred by the first supplier when the customer shares its anticipated volume in the market for that first supplier.

3.3.2. APS

Advanced planning and scheduling (APS) initiatives for the most part came out of industry, although these often depended on solution technologies already developed by the academic operations research community. Essentially an expansion of the concept of finite production scheduling, the idea was to develop better, or “optimized,” production schedules—e.g. minimizing costs subject to a variety of constraints. Sanjiv Sidhu, for one, proposed the idea while working at Texas Instruments Corporation. He

subsequently founded I2 Technologies, and became an extremely strong advocate for the approach.

The term, APS, is now used to represent a wide range of logistical and manufacturing planning tools that employ various optimizing technologies. These range from multiple plant/distribution center master planning (network optimization) to detailed finite scheduling, to transportation optimization and vehicle scheduling. The original offerings used heuristic techniques to identify and attempt to work around process constraints and bottlenecks. Subsequently mathematical programming techniques were employed. Good examples of APS systems in practice utilizing mathematical programming solution technology are provided in (Brown et al. 2001) and (Brown et al. 2002).

What do these efforts have to do with inventories? For the most part APS systems, like their MRP/DRP predecessors, generally take the inventory planning parameters as exogenous and seek to plan the optimum flow while meeting the inventory targets supplied. However in certain production scheduling situations, APS systems determine production run sizes, taking into account set-up or changover times. (See (Brown et al. 2002) for an example).

3.3.3. Multi-echelon techniques

Due to the lack of practical techniques and/or software for optimizing inventories in multi-stage operations, Graves and Willems developed a method for applying multi-echelon inventory techniques (see (Graves and Willems 2000; Graves and Willems 2003)) that has fostered commercial use. They started a firm called Optiant to produce and market software for solving these problems. This software has subsequently been used by several major firms to improve inventory management and supply chain operations (for examples at Hewlett-Packard, see (Billington et al. 2004)). While this product appears to have a great deal of potential, it remains unclear how comprehensive the software is in considering relevant supply chain variables and costs other than direct inventory factors in its analysis.

3.4. 21st Century

More recently, in the first decade of the 21st Century, industrial inventory applications and development continue to focus on better use of the tools and techniques already available. Often these efforts encompass multiple levels in the supply chain to a greater extent than in previous decades. In addition, many practitioners are anticipating that technologies such as radio frequency identification (RFID) can bring greater visibility to the level and location of inventories in the supply chain—knowing how much of each SKU is in each location, on-demand, and in real-time.

3.4.1. Continued refinement

In some cases, the mere application of standard techniques—with a focus on the underlying creators of inventories—continues to produce significant improvements for the firm making the effort. For example, Revlon Inc. reduced inventories by 27 percent while maintaining a 97 percent or higher case-fill rate by making a comprehensive effort to fine-tune the basic inventory parameters—computing safety stocks using statistical techniques, reformulating production cycle stocks, reducing supplier lead times; and generally coordinating purchasing, production, distribution, and inventory

decisions across their supply chain (Davis et al. 2005). In other cases, more sophisticated, multi-echelon models have been used as was the case at Hewlett Packard (H-P). H-P redesigned their supply chains for both digital cameras and for ink-jet printer cartridges with the aid of the Optiant software previously cited (Billington et al. 2004).

At least some new technology has been developed to deal with specific problems. For example, researchers at Philips (de Kok et al. 2005) sought to better synchronize decisions regarding capacities and material flows in the highly volatile electronics industry. Unable to find software or best practices that met their needs, they developed their own “collaborative planning process and collaborative planning software” (de Kok et al. 2005, p. 38). In this case they developed software that allowed those involved to operate in an interactive planning environment.

3.4.2. Better information through RFID

Another major initiative of the last few years has been the effort to get better information about the specific location and amount of inventory for each and every SKU in a supply chain. The big hope in many industries is that RFID tags and readers will allow the gathering of the appropriate information on a real-time basis, and that large-scale information and communications networks will make the data available to those with a need to know. Then those parties can take appropriate actions—possibly through multi-firm collaborations—to maintain adequate inventories while reducing out-of-stock situations. (See Angeles, 2005 for more discussion of RFID in the supply chain.)

Obviously, inventory researchers will recognize that more timely and accurate information on inventory quantities and locations alone will not automatically result in better inventory management. Some benefit may come if the more sophisticated supply chain members use this information to identify where inventories (and those in charge of them) in the supply chain could be managed more effectively and take action to insure improvements are made. But major improvements will be made only if firms employ the proper tools—especially multi-echelon techniques that can take an inter-firm, as well as intra-firm, perspective.

4. Issues and Challengers

While there have clearly been numerous interesting and useful developments in industrial inventory management applications, there remain challenges. These relate to teaching and education on the one hand, and research on the other.

4.1. Teaching and education

Unfortunately many of the successes discussed above are exceptions rather than the rule (for some corroboration, see (Frankel 2006)). I continue to find companies that still use intuitive methods for determining the inventory planning parameters—even when those companies own software that has sophisticated inventory management capabilities. The use of the standard deviation in safety stock calculations seems to intimidate users and their managers. In other cases I find companies that are using the available, good techniques, but applying them incorrectly! For example, a common error is to use a “frequency of stockouts” criterion when setting safety stocks when actually the “percentage of demand filled from inventories” criterion is the relevant one to use, or

vice versa. Types of companies such as these should not expect even to consider multi-echelon techniques until they address the more basic issues.

The major problem seems to be a lack of knowledge, which probably derives from a lack of education in good inventory management principles. I think we in business education all know that both undergraduate and MBA students get precious little in their course work with regard to inventories. Either because we are faced with the “mathematics-phobia” of many business students and choose not to get very deep into inventory management, or because senior administrators and faculty leaders do not see the need for more in-depth instruction in the topics of inventory, we are failing our future managers by not insisting on teaching them more inventory theory and application. The burden of changing these facts lies not on others, but on ourselves. We, the teachers and researchers in this field, must have the will to push for more time spent on inventory management in our courses and our graduate programs.

4.2. Research needs

Research needs in inventory management continue to exist, as well. One of the biggest issues relates to the incorporation of inventory issues into supply chain design efforts. As I mentioned earlier in this essay, my work on supply chain (network) design issues often produced evidence of poor inventory management processes. But this was only part of the problem. Even today, the major means of evaluating how inventories might behave, and how much inventory in what form should exist throughout the network is usually computed after the optimization of the network. In other words, inventory issues are not very well considered in the network optimization process itself (see (Melo, Nickel and Saldanha da Gama 2006) for a comprehensive discussion of the issues and possible solution methodologies). Lee and Billington (1993) offer a decentralized approach to the problem; but this is, again, an approximation.

Inventory levels vary with the volume of the flow through a node in a network and the supply/demand uncertainties faced at that node. These factors depend on the flows from demand points assigned to the node, as well as the sources of supply for the demand so assigned. The typical mathematical programming optimization techniques used to determine these flows do not handle the inventory issues well, thus the reason for using the sequential or decentralized processes described above. We need continued research efforts to arrive at better solutions techniques that can be employed by companies to their actual problems in this area.

5. Conclusions

Inventory management, and research about it, continues to be an exciting and fruitful area of opportunity in business. This is especially true as firms take a broader, supply chain point of view. Managers today have many tools and techniques available for improving the planning and management of inventories in ways that enhance the profitability and competitiveness of their firms. As I have shown, most of these tools have been around for a long time. Managers can purchase computer software that embodies these tools in ways that facilitate their routine use in the operation of the business. Unfortunately these tools are not being used as often or as effectively as they should be.

One of the most important challenges for academics and researchers in the field of inventories is to continue to promote the use of good inventory management practices; and to disseminate information about those practices in undergraduate, graduate, and

executive education. The other challenge is to continue to develop new techniques. Especially as we continue to work closely with industry, we find new situations where the tried and true tools of the past are just not adequate (as shown in the examples cited at Philips and Hewlett-Packard). And as the concept of supply chain management (not just local management) becomes more widely accepted, the scope of any inventory optimization effort expands greatly in terms of the number of locations and echelons involved.

References

- Angeles, R. 2005. "RFID technologies: Supply-chain applications and implementation issues." *Information Systems Management* **22**:51-65.
- Baumol, W. J., and H. D. Vinod. 1970. "An inventory theoretic model of freight transport demand." *Management Science (pre-1986)* **16**:413.
- Billington, C., G. Callioni, B. Crane, J. D Ruark, and et al. 2004. "Accelerating the Profitability of Hewlett-Packard's Supply Chains." *Interfaces* **34**:59-72.
- Brown, G., R. F Dell, R. L Davis, and R. H Duff. 2002. "Optimizing plant-line schedules and an application at Hidden Valley Manufacturing Company." *Interfaces* **32**:1-15.
- Brown, G., J. Keegan, B. Vigus, and K. Wood. 2001. "The Kellogg Company optimizes production, inventory, and distribution." *Interfaces* **31**:1-17.
- Davis, D., J. Buckler, A. Mussomeli, and D. Kinzler. 2005. "Inventory transformation: Revlon style." *Supply Chain Management Review* **9**:53-59.
- de Kok, T., F. Janssen, J. van Doremalen, E. van Wachem, M. Clerkx, and W. Peeters. 2005. "Philips Electronics Synchronizes Its Supply Chain to End the Bullwhip Effect." *Interfaces* **35**:37-48.
- Forrester, J. W. 1958. "Industrial dynamics." *Harvard Business Review* **36**:37-66.
- Frankel, R. 2006. "The role and relevance of refocused inventory: Supply chain management solutions." *Business Horizons* **49**:275-286.
- Graves, S. C., and S. P. Willems. 2000. "Optimizing Strategic Safety Stock Placement in Supply Chains." *Manufacturing & Service Operations Management* **2**:68-83.
- . 2003. "Erratum: Optimizing strategic safety stock placement in supply chains." *Manufacturing & Service Operations Management* **5**:176-177.
- Hadley, G., and T. M. Whitin. 1963. *Analysis of inventory systems*. Englewood Cliffs, N.J.: Prentice-Hall.
- Houlihan, J. B. 1985. "International supply chain management." *International Journal of Physical Distribution & Materials Management* **15**:22-38.
- Koch, C. 2002. "It All Began with Drayer ; The world was transformed when Procter & Gamble's Ralph Drayer and Wal-Mart's Sam Walton sat down in 1987 to discuss a better way of keeping Wal-Mart in diapers." *CIO* **15**:1.
- Lee, H. L, and C. Billington. 1993. "Material management in decentralized supply chains." *Operations Research* **41**:835-847.
- Lewis, H. T. 1956. *The role of air freight in physical distribution. : Pt. 1. Characteristics of air freight and its market*. Cambridge, MA: Division of Research, Graduate School of Business Administration, Harvard University.
- Melo, M T, S. Nickel, and F S. da Gama. 2006. "Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning." *Computers & Operations Research* **33**:181-208.

- Orlicky, J. 1975. *Material requirements planning; the new way of life in production and inventory management*. New York,: McGraw-Hill.
- Stenger, A. J. 1985. "The Freight Transportation Analyzer." State College, PA: MAS Associates.
- Stenger, A. J., and J. L. Cavinato. 1979. "Adapting MRP to the outbound side--distribution requirements planning." *Production and Inventory Management* 20:1-14.
- Stenger, A. J., J. J. Coyle, and M. S. Prince. 1977. "Incorporating transportation costs and services into the inventory replenishment decision." Pp. 22-26 in *Applied distribution research*, edited by R. G. House and J. F. Robeson. Columbus, OH: The Ohio State University Transportation Research Fund.
- Stenger, A. J., and W. H. Cunningham. 1978. "Additional insights concerning rail-truck freight competition." *Transportation Journal* 18:14-24.
- Sugimori, Y., K. Kusunoki, F. Cho, and S. Uchikawa. 1977. "Toyota production systems and kanban system materialization of just-in-time and respect-for-human system." *International Journal of Production Research* 15:553-564.
- Tyworth, J. E., K. Rao, and A. J. Stenger. 1991. "A logistics model for purchasing transportation to replenish high demand items." *Journal of the transportation research forum* 22:146-157.
- Whybark, D. C. 1975. "MRP: A profitable concept for distribution." Pp. 82-93 in *Research issues in logistics*. Columbus, OH: Ohio State University.
- Wight, O. W. 1974. *Production and inventory management in the computer age*. Boston,: Cahners Books.
- Womack, J. P., and D. T. Jones. 1996. *Lean thinking : banish waste and create wealth in your corporation*. New York, NY: Simon & Schuster.

Pricing for Variations in Large Loads and Wind Generations

Bhujanga B Chakrabarti

System Operations, Transpower NZ Ltd

Bhujanga.chakrabarti@transpower.co.nz

Abstract

The output of wind generators may fluctuate over a wide range due to variation in wind speed over time. The fluctuations and uncertainty are presently covered, in the New Zealand electricity market (NZEM), by the reserves originally offered for the protection of larger risk units in the system and frequency regulating reserve with no cost to the wind generators. In fact, smaller units benefit from the magnitude of reserves determined by the largest risk units. The frequency of outage of wind generators and variability of wind generation outputs is quite high compared to other smaller units and these frequently need a considerable amount of reserves to cover the generation fluctuations. There are also large loads in the system those fluctuate and cause frequency decay, and need reserve to address the frequency variation issue. While these are true, there have not been a formal procedure or model in place that relates this lack of reliability to a market based pricing mechanism in electricity markets.

This paper puts forward a concept to formally recognise the lack of reliability of wind generators and large loads for critical periods, e.g., during the contingency state. It also integrates the concept to a market clearing model for energy and reserve co-optimisation and explains the pricing implications for such fluctuations. The developed dispatch and pricing model is based on DC-OPF, driven by generator risk contingencies, security margins at the load buses to cover the large variation of loads, and security margins at the wind generator buses to account for the uncertainty of wind generations. These are briefly discussed in the next section. The main objective of this paper is to examine the effect of security margin constraints at the wind generator buses and also at the load buses on the nodal spot prices.

Key words: Contingency state, Electricity market, Fluctuations, Hydro back-up, Large loads, Nodal price, Reserve, Security margins, Wind generation.

1 New Components in the Model

The three major new components which are introduced for the first time in the model are briefly discussed in this section. These are:

- I. Contingency State
- II. Security margin at the load Buses
- III. Security margin at the wind generator (farm) Buses

1.1 Contingency State

The contingency state is modelled using a set of additional equations and constraints in the model in order to maintain feasibility during the contingency state.

1.2 Security margin at the Load buses

We define the Security Margin (SM) as a load margin in terms of MW from the current operating point. The objective of the security constraint is to secure the operating point within a pre-defined margin should such a loading condition occur due to some contingency, c . The margin in the model is maintained using reserve. The SM at each bus i during contingency c is defined as:

$$k_{i,c} = \frac{P_{di,c} - P_{di}}{P_{di,c}}; \forall i, \forall c,$$

Rewriting,

$$P_{di,c} = \frac{P_{di}}{1 - k_{i,c}}; \forall i, \forall c$$

$P_{di,c}$ Demand at bus i during contingency condition, c .

P_{di} Demand at bus i during normal condition.

$k_{i,c}$ Security Margin at bus i during contingency condition, c .

For example,

$$P_{di} = 100 \text{ MW}, k_{i,c} = 0.2 (= 20\%), P_{di,c} = \frac{P_{di}}{1 - k_{i,c}} = \frac{100}{0.8} = 125 \text{ MW}$$

1.3 Security margin at the wind generator (farm) buses

Wind generation (WG) is primarily an energy source with limited ability to provide reliable generation capacity at the peak load periods. Wind variability increases the need for operating reserve and associated generation capacity to manage frequency. The security margin at the wind generator (farm) buses can be thought of as being related to uncertainty and fluctuations in wind generations. The wind generation variability could be different at different buses and the margin at the wind generator buses is assumed to be a function of, among many factors, number of generators in the farm, their locational dispersion in the farm (if measurement interval is of order 10 sec or so), and the standard deviations of wind variations. It is assumed in this paper that the required margin can be statistically estimated. The margin in the model is maintained using reserve.

The security margin at each WG bus i during contingency c is defined as:

$$k_{gi,c} = \frac{P_{gi} - P_{gi,c}}{P_{gi}}; \forall c, i \in \text{wind generators}$$

P_{gi} Generation at bus i during normal condition.

$P_{gi,c}$ Generation at bus i during contingency c

$k_{gi,c}$ Security Margin at the wind generator bus i . It could range between 0 and 1 both inclusive, and represents the reduction in net generation available from a wind generator in the contingency state. The buses i where $k_{gi,c}$ is non-zero are included in the set of WG buses, a sub set of generator buses. Rewriting the above equation, we get,

$$P_{gi,c} = P_{gi}(1 - k_{gi,c}); \forall c, i \in \text{wind generators}$$

The model assumes all reserve is provided by partly loaded generation plant with no contribution from demand side participation (e.g. Interruptible Load) or frequency keeping reserve.

2 Results

The paper demonstrates that the generation, and demand prices at a bus are different in the presence of binding security margin constraints. Generation prices at the wind generator buses get reduced in the presence of binding security margin constraints at these buses. Similarly, the demand price increases due to binding security margins at the load buses.

It should be noted that there are different ways to recover the cost of reserve required to mitigate the fluctuations of wind generations, large loads and the loss of risk generators. For example, in the NZ electricity market, the cost of reserve required due to the loss of risk generators is recovered through a “pool” as stipulated in the Electricity Governance rules (EGR). The idea presented in this research paper is only one of the many possibilities to recover the reserve cost, and is definitely not a proposal or statement from Transpower.

3 Further Works

More work needs to be done to address issues like size of different contingencies (margins), co-optimisation of frequency reserve, and possibility of hydro back-up for large wind generating farms. It is necessary to analyse various hydro scenarios and examine the interaction between hydro and wind.

4 Acknowledgements

I am grateful to my Ph. D supervisors Drs E. Grant Read of University of Canterbury, New Zealand and Deb Chattopadhyay of Charles River Associates, who helped me extensively to develop different dispatch and pricing models during my research period. I sincerely thank Mr Kieran Devine, Mr Doug Goodwin of Transpower NZ Ltd for their interest and encouragement in my research works.

5 References

These references are used while developing the model.

1. M. C. Caramanis, R. E. Bohn and F. C. Schweppe, "Spot Pricing of Electricity: Practice and Theory," IEEE Transactions on Power Apparatus and Systems, 101(9), pp. 3234-3245, 1982.
2. F. C. Schweppe, M. C. Caramanis, R. D. Tabors and R.E. Bohn, "Spot Pricing of Electricity," Kluwer Academic Publishers, Boston, 1988.
3. M. C. Caramanis, R. E. Bohn and F. C. Schweppe, "System Security Control and Optimal Pricing of Electricity," Electrical Power and Energy Systems, 9(4), pp. 217-224, 1987.
4. R. J. Kaye, F. F. Wu and P. Varaiya, "Pricing for System Security", Paper #92-WM-100-8, IEEE Winter Power Meeting, New York, January, 1992.
5. A. W. Berger and F. C. Schwepes, "Real Time Pricing to Assist in Load Frequency Control," IEEE Transactions on Power Systems, 4(3), pp. 920-926, 1989.
6. W. Hogan, "Contract Networks for Electric Power Transmission", Journal of Regulatory Economics, 4(3), pp. 211-242, 1992.
7. M. L. Baughman and S. N. Siddiqi, "Real Time Pricing of Reactive Power: Theory and Case Study Results," IEEE Transactions on Power Systems, 6(1), pp. 23-29, 1991.
8. D. Chattopadhyay, B. B. Chakrabarti and E. G Read, "Pricing for Voltage Stability," proc. IEEE, PICA conference, Sydney, May 2001.
9. E. G. Read. and B. J. Ring, "Dispatch Based Pricing,". Report prepared for Trans Power New Zealand Limited. Wellington, 1995.
10. E. G. Read, G. R. Drayton-Bright and B. J. Ring, "An integrated Energy and Reserve Market for New Zealand," Management Science Department, University of Canterbury, EMRG-WP-98-01, 1998.
11. B. B. Chakrabarti and E.G. Read, "Pricing Implications of Security Constrained Dispatch," in Proc. ORSNZ Conference, Auckland, 2004.
12. N. S. Rao, "Optimal Dispatch of a System based on Offers and bids – A mixed Integer LP Formulation," IEEE Transactions on Power Systems, vol. 14, 1999..
13. T. Alvey, D. Goodwin, X. Ma, D. Steriffert and D. Sun, "A Security Constrained Bid-Clearing System for the New Zealand Wholesale Electricity Market," IEEE Transactions on Power Systems, vol. 13, 1998.
14. B. B. Chakrabarti, "Reactive Power Management and Pricing," Ph. D Thesis, University of Canterbury, 2004.
15. W. Hogan, E. G. Read and B. J. Ring, "Using Mathematical Programming for Electricity Spot Pricing," International Transactions on Operational Research, 3(4), pp. 209-221, 1996.
16. B. B. Chakrabarti, "Modeling of Wind Generation Fluctuations in a Dispatch Model", 2006 IEEE Power India Conf, New Delhi , April 10-12, 2006

Biography

Bhujanga B. Chakrabarti has been working with System Operations Group in Transpower New Zealand Limited. He obtained his BEE degree from Jadavpur University, India in 1975, MS from Northern Illinois University, MEE and Diploma in Business Administration from University of Newcastle, Australia and Ph.D. from University of Canterbury, New Zealand in 2004. He has been working for more than 30 years in power system analysis, control, planning, and pricing areas.

National Instantaneous Reserve Market in the New Zealand Wholesale Electricity Market

Vladimir Krichtal
System Operations
Transpower NZ Ltd
New Zealand

vladimir.krichtal@transpower.co.nz

Abstract

The Scheduling, Pricing and Dispatch (SPD) model is used in the New Zealand Wholesale Electricity Market to produce security constrained optimal dispatch and prices for energy and instantaneous reserves. The energy model is based on a configuration that consists of two separate multinodal AC systems (North and South islands) connected by a DC link. The reserve model has two separate zones based on the two islands.

A likely development in NZ will be the introduction of a national reserve market making use of the available capability on the DC link to share instantaneous reserves across the islands. To do this, reserve transfer decision variables, DC ramp limits, DC modulation limits and reserves availability constraints were introduced into SPD. Simulation experiments with real offers and demand scenarios show that this would make significant saving in the NZ electricity market.

The paper also proposes a new methodology for reserve revenue calculation based on the shadow price for each risk-reserve constraint. This would make some saving compared to existing zonally-based reserve price methodology.

Key words: security constrained optimal dispatch, risk, reserve market, DC link.

1 New Zealand Electricity dispatch model with ability to transfer instantaneous reserves across the islands.

The New Zealand Wholesale Electricity Market (NZWEM) uses the Scheduling, Pricing and Dispatch (SPD) model. The model's network structure consists of two islanded AC networks connected by the Direct Current (DC) line.

The AC power flow is modelled based upon the direct current power flow framework at bus level.

The instantaneous reserve model incorporated in SPD has two separate reserve collecting areas (North and South islands). Instantaneous reserves are maintained in each area to stabilise the system when an under frequency risk event occurs (loss of a generator at risk or the DC's capacity). The amount of instantaneous reserves cleared

in each island is utilised within the same island. The model's objective is to minimise total energy and reserves opportunity cost, given the constraints of network security, capacity and generator ramping capability. The model produces an optimal dispatch for energy, instantaneous reserve, and nodal energy and island reserve prices.

The reserves transfer ability of the DC is not modelled in SPD at the present time. The DC has a limited modulation capability that allows transferring power between the islands to support frequency. As a result, instantaneous reserves in one island can be used to cover risks in the other island. More of this capability could be built if analysis shows an economic efficiency for such investment.

The proposed national reserves market model includes reserves transfer variables and linear constraints: DC ramps, modulation limits and reserve conservation constraints. The formulation of the reserve transfer model can be found in the presentation slides.

2 Reserve revenue calculations of risk constraints with multiple contingencies.

Power System components can be separated into two risk groups. The first group of risks do not produce reserves (Generators without reserve, Interruptible load). The second group can produce reserves (Generators with reserves offers, DC with reserves transfer capacity). Instantaneous reserves revenue in the existing SPD is based on reserve prices in each island, multiplied by the total amount of reserve cleared in the area. A proposed formula for reserves revenue would include only risk shadow prices multiplied by the actual risk. So, a double counting of reserve revenue from risks which have cleared reserves is avoided.

The revenue difference between old and new methods is always non-negative. It is strictly positive in the case of multiple risk-reserve constraints binding with non-negative shadow prices if a non-zero quantity of reserve is cleared from the accompanying risk.

3 Simulation experiments with the DC reserve transfer model.

The DC reserve transfer model is developed and introduced into the SPD code. The experiments consist of two SPD simulations (with and without reserve transfer model) with the same initial data. The first run is a base case without the reserve transfer model. It is used to identify DC configuration, to calculate upper and lower post contingency limits and to calculate DC loss adjustment parameters. In the second run the reserve transfer model is switched on.

The following economic indices are compared: six second reserve revenue, sixty second reserve revenue, demand energy revenue, total system revenue and objective function.

Simulation of the DC reserve transfer model with SPD has shown a comparative reduction of total market costs by 0.39% in March, 1.04% in April, 1.56% in May and 1.3% in June, for the year 2004, which is about \$0.68m overall. A total reduction of energy and reserve revenue for the same period would be around \$9.6m. Simple extrapolation of these results for average year should be done with caution because of different hydrology, load and offer scenarios.

4 Acknowledgement

I am very grateful to Roger Miller from Transpower NZ Ltd for advice and providing technical details of NZ power system. I sincerely thank Kieran Devine, Doug Goodwin, John Clarke, Graeme Ancell, of Transpower NZ Ltd for their interest and encouragement in my research works.

5 References

Scheduling, Pricing and Dispatch software: Mathematical formulation. 2006. Transpower NZ.

<http://www.electricitycommission.govt.nz/pdfs/opdev/servprovinfo/servprovpdfs/spd-formulation-v4.2.pdf>.

Risk-Adjusted Discount Rates and Optimal Plant Mix: A New Formulation for Electricity Market Optimisation

Grant Read
University of Canterbury
grant.read@canterbury.ac.nz

Deb Chattopadhyay

Abstract

Traditional capacity expansion formulations are often still employed to generate forecasts of future investment, even in de-regulated electricity markets. It can be shown that this is appropriate, because commercial incentives align with traditional planning criteria, provided an appropriate commercial discount rate is employed in the analysis. This paper does not attempt to determine an appropriate discount rate, but starts from the observation that the required rate will depend on perceived risk, and risk depends significantly on the role capacity is expected to play in the power system, with dry year backup plant being a particularly risky investment in hydro systems. The problem is that a traditional capacity expansion formulation can not represent this “utilisation risk”, which is not a property of plant type, per se, or of “load class” in the traditional sense. We introduce, and demonstrate, a new formulation which overcomes this problem by dividing the “Load Duration Curve” into (horizontal) “load slices”, rather than (vertical) “load classes”, and applying a utilisation risk adjusted discount rate to investment to meet each load slice.

A mass-balance gas simulation model for assessing the benefits of gas network augmentation options

Andrew Kerr
Concept Consulting
andrew@concept.co.nz

Abstract

VENCorp is the independent system and market operator of the Victorian gas transmission system. One of their responsibilities involves identifying and assessing gas pipeline expansion options to address transportation limitations that can affect gas consumers. System security problems can arise when the linepack capability is combined with uncertainty about daily supply and demand, particularly during winter. Since 2004, VENCorp has forecast a capacity shortfall and identified that in order to meet gas planning standards, a major system augmentation was required before the winter of 2008. In 2005, the formal process for developing an augmentation commenced with the first phase being to perform an assessment of the costs and benefits of the suitable options. A model was required to perform this task, with key requirements being that it needed to be robust enough to withstand the scrutiny of the ACCC and public consultation, needed to be built over a relatively short timeframe to support the regulatory process, and was expected to produce result sets in under an hour. This paper describes the key components of the model and process of arriving at the end product.

Reliability Based Assessment Model for Space Vehicles

Alex J. Ruiz-Torres¹ Carey McCleskey^{2a} Kazuo Nakatani³ Arun Pennathur^{4b}

Russell Rhodes^{2c} Edgar Zapata^{2d} and Jianmei Zhang^{4e}

¹ Blue Frog Technologies, 806 Turney Dr., El Paso, Texas 79902, United States

² NASA- Systems Engineering, KSC, Florida 32899, United States

³ Computer Information Systems and Decision Sciences, Florida Gulf Coast University
Fort Myers, Florida 33965, United States

⁴ Department of Industrial Engineering, University of Texas at El Paso
El Paso, Texas 79968, United States

¹ ruiztorres@blue-frog.biz

² carey.m.mccleskey@nasa.gov

³ knakatan@fgcu.edu

^b apennathur@utep.edu

^c russel.e.rhodes@nasa.gov

^d edgar.zapata-1@nasa.gov

^e jzhang2@utep.edu

Abstract

Next generation space transportation systems must be designed for safety and maintainability. Crew safety is paramount to maintaining human presence in space, while maintainability is directly linked to the cost of accessing space. This paper describes a reliability-based design tool to be used in the assessment of crew safety and maintainability of future space transportation systems. The tool uses basic reliability principles to estimate the probability of a safe mission and the need for repairs/replacement during ground processing before launch and start of mission, based on the characteristics of the vehicle's main systems: the number of subsystems, the mean time to repair, and the per subsystem average reliability. The paper describes how the developed software uses various sensitivity analysis functions to improve design decisions.

Key words: Reliability, Maintainability, Safety, Space Vehicles, Transportation.

1 Background

NASA recently awarded Lockheed Martin with the contract to develop the Orion spacecraft, the space transportation system that will replace the Space Shuttle as the United State's manned vehicle. The Orion spacecraft is a component of the Constellation Program, whose aim is the continued exploration of space, focusing on a

return to the Moon and future travel to Mars and beyond. A key goal of the Constellation Program is to increase the safety, reliability, and cost efficiency of space transportation. This paper describes a top level methodology and design tool developed in support of the Constellation Program.

2 Introduction

New spacecraft must be designed for dramatic improvements in reliability, maintainability and safety (RMS). Crew safety drives many of the design decisions, including the target reliability for the system and its subsystems. Furthermore, the reliability of the subsystems and the number of components for each subsystem will drive the maintainability and therefore the operational costs of the transportation system. The cost of space transportation, in combination with low reliability and operability, are currently the major obstacles to the continued exploration and development of space (Morris 2004, Mankins 2002, Scott 1998)

During the design and development of new spacecraft, subsystems are typically designed independently, which in some cases results in redundancies. Each subsystem has a well defined function, for example life support, propulsion, reaction control, and has design targets, for example weight, volume, reliability, and availability. To accomplish the targeted reliability, designers often depend on backup systems, thus the designers have parallel components in place, sometimes in a cold-standby condition, and sometimes in an active-standby condition, rather than making each component of the subsystem more reliable. By designing in this manner, “mature” technologies can be implemented in the design, which require “small” research and development expenditures. However, what designers often do not recognize is the maintainability burden that these “low” reliability redundant components create.

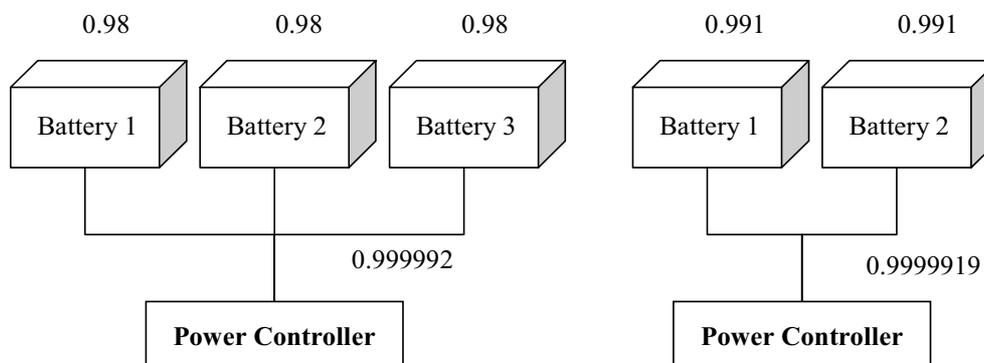


Figure 1. Example subsystem.

For example, Figure 1 illustrates a power subsystem with one main battery and two backups (thus only one needed for subsystem function). Assume the reliability of each battery is 98%, so the probability during the mission of complete battery loss is $(0.02)^3 = 0.000008\%$. However, in terms of maintainability and ground processing, all three batteries must be inspected/ tested, and when failures found, repair/ remove/ replace (R/R/R) must be done (ground processing define). Given this work is due to failures, we refer to it as unplanned R/R/R (U-R/R/R). Based on averages, the expected number of batteries that will fail during processing is $(2\%)(3) = 0.06$. Assuming U-R/R/R work per battery failure is 1,000 hours, the expected U-R/R/R burden is 60 hours. By comparison, if research and development can craft a battery with a reliability of 99.1%, the probability of complete battery loss will be $(0.009)^2 = 0.0000081\%$ (similar level of

subsystem reliability) and the expected U-R/R/R burden is $(0.9\% \times 2 \times 1,000)$ or 18 hours for U-R/R/R. Clearly, the probability that at least one battery will fail during ground processing is much higher for the first case than the second regardless of what random distribution is assumed. Assuming a Poisson random variable, the probability of one failure (which requires 1,000 hours) for the first case is 5.65%, while for the second case, the probability of one failure is 1.77%. This relationship between failures during preparations for launch, and failures that may affect safety once a mission is underway, is exacerbated by low component level reliabilities. This means that the notion of using a redundancy in a system as a means to simply press forward and launch is not allowed. The redundancies are preserved only for during the mission, not to avoid a processing delay or to provide flexibility with launch criteria

This paper presents a methodology being developed to estimate the safety and maintainability burden of future spacecraft systems based on the reliability of the components, the number of components, and the time to repair the subsystems. The paper also describes a software tool where the methodology is being implemented. The goal of the tool is to provide engineers/vehicle developers during the early stages of design with a tool that demonstrates the effect on maintainability of improving component reliability and reducing the number of components. Section 3 explains the RMS modeling approach used to describe the subsystems, while Section 4 presents the RMS modeling approach used to describe the complete vehicle system. Section 5 describes the current software prototype and some of its functions. Section 6 concludes the paper and provides a description of the future work on the model and software.

3 RMS Modeling Approach - Subsystems

Reliability is the probability that a component will function correctly for a given interval of time under stated conditions (Carrese 2006). For a spacecraft, the time under consideration includes both the time on the ground during check out and testing and then the time during flight operations. In our modeling approach, the overall reliability of a subsystem is generated by Equation 1, which simply states that for a subsystem to function properly all of its critical components must work properly. The equation assumes that all components have a common reliability (therefore the term average reliability used).

$$r_j = ar_j^{n_j} \quad (1)$$

Where

r_j	Reliability of subsystem j .
ar_j	Average reliability of the components of subsystem j .
n_j	Number of critical components of subsystem j (failure of any of these components will cause the failure of the subsystem).

Note that the model assumes the reliability of the subsystem is the same at any point of processing, and during flight. The model estimates three measures of performance for each subsystem:

1. Expected U-R/R/R workload.
2. Expected U-R/R/R duration.
3. Probability that U-R/R/R duration is less than a set value X .

3.1 Expected U-R/R/R workload

The maintainability burden of a subsystem is based on the total number of components in the subsystem (which is larger or equal to number of critical components). The expected number of failures for a subsystem is modeled by Equation 2, and the expected time for U-R/R/R by Equation 3.

$$f_j = (1 - ar_j) \times m_j \quad (2)$$

$$w_j = f_j \times t_j \quad (3)$$

Where

f_j	Expected number of failures per processing cycle for subsystem j .
m_j	Total number of components for subsystem j (failure of any of these components will require U-R/R/R activities during ground processing)
t_j	Average time to U-R/R/R when one or more components of the subsystem fail during ground processing.
w_j	Expected U-R/R/R time per processing cycle for subsystem j .

As a simple example, let a sample subsystem be described by 10 critical components (n_j) and 25 total components (m_j). The average component reliability (ar_j) is 99.9% and the time to U-R/R/R the subsystem in the event of a component failure (t_j) is 2,000 hours. Based on Equation 1, the subsystem reliability (r_j) is 99%, based on Equation 2 the expected number of failures (f_j) is 0.025 per processing cycle, and based on Equation 3, the expected U-R/R/R time (w_j) is 50 hours.

3.2 Expected Duration

Based on the expected U-R/R/R time and subsystem workload factor, the estimated duration of the processing is calculated (Equation 4). The compression factor accounts for the amount of parallel work possible and the number of shifts per week.

$$d_j = w_j \times h_j \quad (4)$$

Where

d_j	Duration of the expected work for subsystem j .
h_j	Compression factor for subsystem j . The compression factor is expressed in duration weeks versus workload R/R/R hours.

For example, if the compression factor (h_j) is 1 week for 300 hours of U-R/R/R, the duration of the expected work for the sample subsystem (d_j) is 0.166 weeks.

3.3 Probability of duration less than X

The third measure of performance for a subsystem is the probability that the duration of U-R/R/R is less than or equal to a set value X . We assume a Poisson distribution to model subsystem failures. The probability of y failures for subsystem j is given by Equation 5.

$$P_f(y) = (f_j / y!) e^{-f_j} \quad (5)$$

To determine the probability that the duration for a subsystem j is less than X , we simply add the probability values for y from 0 to the last number of failures that result in a duration less than X . The maximum number of failures that will be below the X limit is

given by Equation 6, and Equation 7 gives the probability that subsystem j meets the X maximum duration (in weeks) for U-R/R/R processing.

$$\theta_j = \lfloor X / (t_j \times h_j) \rfloor \quad (6)$$

$$P_j(\text{duration} \leq X) = \sum_{y=0 \dots \theta_j} P_j(y) \quad (7)$$

Where

θ_j Number of failures for subsystem j that meets the duration value X .

For a target of 12 weeks ($X = 12$), the value of θ is 1, and the probability $P_j(\text{duration} \leq X) = 99.969\%$.

4 RMS Modeling Approach - Vehicle

The RMS analysis of the vehicle combines the measures derived for the subsystems. In addition, the safety during mission is determined, thus a total of four measures of performance.

4.1 Expected U-R/R/R workload and duration

The expected total maintainability burden for the vehicle is the sum of all the U-R/R/R values (Equation 8) and the expected duration is the maximum for all subsystems (Equation 9). The assumption for the duration calculation being that there are no sequence dependencies between subsystems, thus the worst case subsystem will determine the expected duration.

$$W = \sum_{j \in S} w_j \quad (8)$$

$$F = \max_{j \in S} w_j \quad (9)$$

Where

S The set that contains all the subsystems for the defined vehicle architecture.

W The total workload due to U-R/R/R for the vehicle architecture

F The set that contains all the subsystems for the defined vehicle architecture.

4.2 Probability of duration less than X

The overall probability that a transportation architecture will require less than X weeks for ground processing is based on the joint probability that all the subsystems will require less than X weeks for U-R/R/R. This is modeled by Equation 10.

$$P_{system}(\text{duration} \leq X) = \prod_{j \in S} P_j(\text{duration} \leq X) \quad (10)$$

4.3 Mission Safety and Outcomes

To estimate the safety of the vehicle system we assume each subsystem is critical for one of two phases considered (arbitrarily) of the mission, or is not safety critical. The first phase of the mission is that where the option for a launch abort system (LAS) assisted abort is feasible (an abort condition forces the launch abort system to go operational; the abort system has its own flight reliability r_{abort}). The second phase of the mission is that where no LAS assisted abort is possible and failure results in the loss of the crew and the vehicle (although other backups may assist in crew return capability

modes). In general the first phase includes the time before liftoff (when the crew is at the pad and the engines are being started) and lasts for about 163 seconds after liftoff (at about 266,000 ft. altitude). Figure 2 presents a tree illustrating the flow of the safety calculations. Let $S(a)$ be the subset from S that includes all subsystems that must function properly (malfunction indicates abort is needed) during the “abort possible” phase of the flight, $S(b)$ be the subset from S that includes all subsystems not in $S(a)$ where failure will cause loss of vehicle.

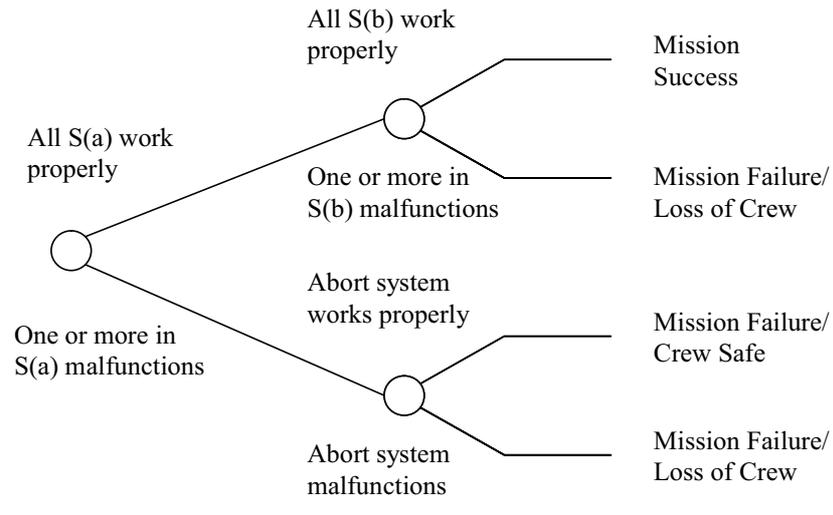


Figure 2. Safety Calculations Tree.

The probability of mission success is determined by Equation 11, the probability of mission failure/ crew safe is determined by Equation 12, and the probability of mission failure/loss of crew is determined by Equation 13.

$$P_{\text{mission success}} = \prod_{j \in S(a)} r_j \times \prod_{j \in S(b)} r_j \quad (11)$$

$$P_{\text{mission failure/ crew safe}} = (1 - \prod_{j \in S(a)} r_j) r_{\text{abort}} \quad (12)$$

$$P_{\text{mission failure/loss of crew}} = (1 - \prod_{j \in S(a)} r_j) \times (1 - r_{\text{abort}}) + \prod_{j \in S(a)} r_j \times (1 - \prod_{j \in S(b)} r_j) \quad (13)$$

5 Software Application

The modeling approach described in Sections 3 and 4 is currently being implemented into a software application. The software uses a Visual Basic for Applications interface and routines, and Microsoft Excel to display graphs and information. The software has typical functions such as save, new, and open which allow users to maintain records of their work, return to previous work, and to share input files (these are text files that contain the names of the subsystems, the reliability, etc.). Figure 3 presents a snapshot of the user interface with a “sample vehicle” loaded into the software. Note that the display to the left includes all the subsystems, grouped by major systems of a space vehicle (called architecture elements). The area to the right provides all the input details for the selected subsystem, in this case the main engines of the “Rocket” System.

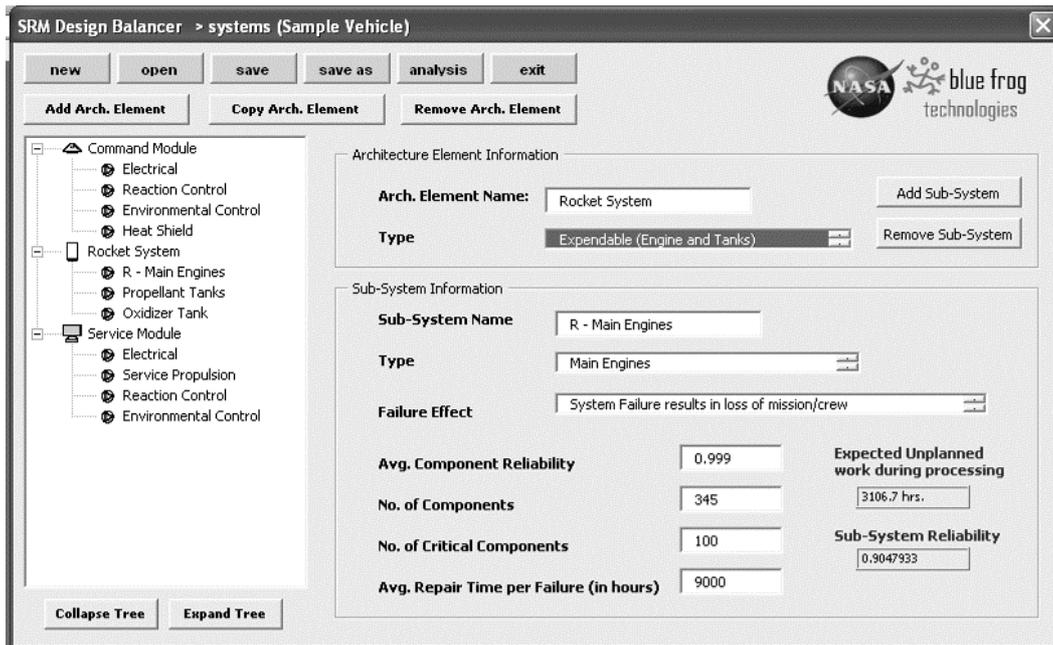


Figure 3. Snapshot of User Interface.

The software determines the measures of performance for the subsystems and the vehicle, providing information into the subsystems that contribute the most to the workload. Figure 4 presents a Pareto chart which shows the ten subsystems that contribute the most to the total U-R/R/R. In this example, the *Electrical* subsystem of the *Service Module* has the highest contribution to the work content with more than 12,000 hours (about 32%).

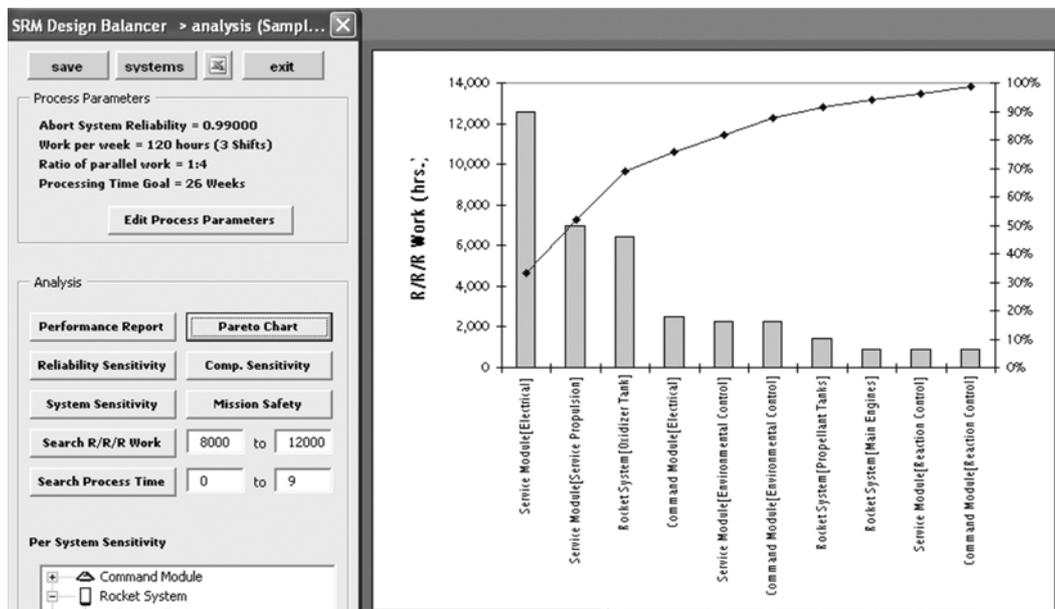


Figure 4. Snapshot of an Output, Pareto Analysis.

The software performs sensitivity analysis by subsystem, illustrating the workload reduction resulting if the average component reliability is improved or the number of components reduced. This is presented in Figure 5. In this case, the *main engines* of the *Rocket system* have a baseline U-R/R/R of 3,107 hours. As the average component reliability is improved (from a current level of 0.999 to 0.99971) the U-R/R/R level is reduced to 888 hours, while as the number of components is reduced (from 345 components to 173) the U-R/R/R level changes to 1,553 hours (half components results in half the time).

The software is also designed to automatically perform multiple types of sensitivity analysis for the overall vehicle. For example, the software estimates the mission outcome probabilities as all the subsystems undergo reliability improvements, or all subsystems undergo a reduction in the number of components. Such an output is presented in Figure 6, where improvements in component reliability are plotted against the 3 mission outcomes. In this case, the current value for mission success changes from a baseline probability of 66% to a probability of 89% when the mean time between failures is increased by 350%.

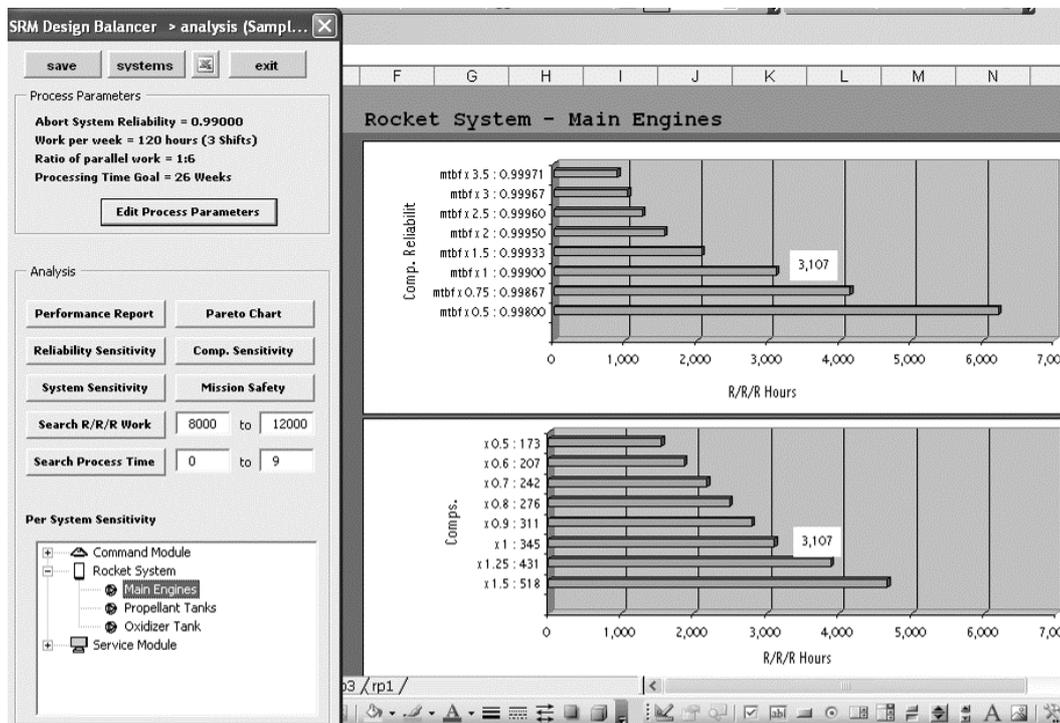


Figure 5. Snapshot of the Output, Subsystem sensitivity.

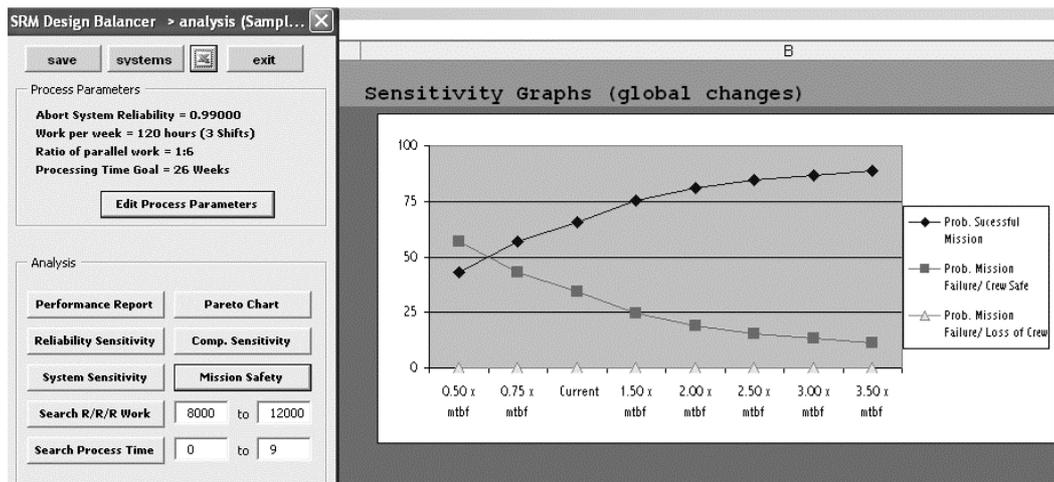


Figure 6. Snapshot of the Output, Mission Success sensitivity.

6 Conclusions and Future Work

This paper presents a simple RMS model being developed to support the top level design process of future space transportation vehicles. The model is being implemented in a software tool which will allow designers and engineers to quickly evaluate their designs in terms of safety and maintainability. The software is still at the “prototype” stage and improvements to the interface and its functions are being discussed.

The long term goal is to increase awareness by quantifiable means of the relation between two high level figures of merit in space transportation system design – that between safety in flight and maintainability on the ground in the months (or hopefully one day – days) before flight. Hardware that is not easily maintainable due to lack of maturity for the challenging environment of space flight is also un-safe hardware one day, and co-relations exist and can be quantifiably understood.

Acknowledgements

This research is funded by NASA contract NNK06MB68C to Blue Frog Technologies in support of the Constellation Program.

References

- Carrese, S. and G. Ottone. 2006. “A model for the management of a tram fleet.” *European Journal of Operational Research* **175**:1628-1651.
- Mankins, J.C. 2002. “Highly Reusable Space Transportation: Advanced Concepts and the opening of the space frontier.” *Acta Astronautica* **51**: 727-742.
- Morris, J. 2004. “Lawmakers express cautious support for Bush space vision.” *Aerospace Daily*, **209**: 1.
- Scott, W. B. 1998. “Airline Ops Offer Paradigm for Reducing Spacelift Costs.” *Aviation Week and Space Technology* (June 15): 64-67.

Application of Two-Dimensional Renewal Processes in Modelling Product Warranties

Dinu Corbu

Ministry of Social Development, IAP Team,
Wellington, New Zealand
dinu.corbu001@msd.govt.nz

Stefanka Chukova

School of Mathematics, Statistics and Computer Science
Victoria University of Wellington, New Zealand
stefanka@mcs.vuw.ac.nz

Jason O'Sullivan

School of Mathematics, Statistics and Computer Science
Victoria University of Wellington, New Zealand
Jason.O.Sullivan@mcs.vuw.ac.nz

Abstract

The study of product warranty is of interest to manufacturers for several reasons. Paid claims are a cost of doing business and a liability incurred by the manufacturer at the time of sale. For these reasons, estimating and forecasting warranty expenses is of interest. Also, warranty data provides information about the durability of products in the field and, therefore, is of interest to engineers. Here we emphasise two-dimensional warranties, like automotive warranties, that guarantee free repairs subject to both age and mileage limits. We aim to evaluate the average warranty expenses under a two-dimensional non-renewing free replacement warranty policy using real-world warranty data. We derive a numerical procedure for the evaluation of the two-dimensional renewal function and use it for the warranty cost analysis. In addition, we test whether the available data can be considered as coming from a two-dimensional renewal process.

1 Two-Dimensional Warranties

Most products are sold with a one-dimensional warranty. The most common one is the non-renewing free replacement warranty, i.e., a newly sold item is warranted for some calendar time of duration T , called warranty period, and the warranter assumes all expenses associated with the failure of the product during this period of time.

Lately, two-dimensional warranties, like automotive warranties, have attracted the attention of many researchers (Blischke, W.R. and Murthy, D.N.P. 1996), (Murthy, D.N.P., Iskandar, B.P., and Wilson R. J. 1995). In automotive warranties, the first warranty parameter is the age of the car, usually depicted on the X -axis, and the second warranty parameter is the mileage of the car, depicted on the Y -axis. An example of an automotive warranty is the following: the automobile is covered by warranty as long as its age is no more than 36 months and the accumulated mileage does not exceeds 36 000 miles. The warranty expires if either the age exceeds 36 months or the mileage exceeds 36 000 miles, whichever occurs first.

Assuming that the warranty repairs rectify the product to its initial reliability level, i.e., after the repair the product is “as good as new”, and the repairs are instantaneous, we can model the process of the failures of the product by a two-dimensional renewal process.

2 Renewal Theory in Two Dimensions

In what follows we provide a brief review of two-dimensional renewal theory and derive a numerical procedure for solving the two-dimensional renewal equation.

Bivariate Renewal Processes. A bivariate renewal process (Hunter, J. J. 1974) is a sequence of i.i.d. non-negative bivariate random variables $\{(X_n, Y_n)\}$, $n = 1, 2, \dots$, with a joint c.d.f. $F(x, y) = P\{X_n \leq x, Y_n \leq y\}$. Let us consider the bivariate sum

$$S_n = (S_n^{(1)}, S_n^{(2)}) = \left(\sum_{i=1}^n X_i, \sum_{i=1}^n Y_i \right), \quad (1)$$

and define the corresponding counting process

$$N_{x,y} = \max\{n : S_n^{(1)} \leq x, S_n^{(2)} \leq y\}. \quad (2)$$

The marginal sequences $\{X_n\}$ and $\{Y_n\}$, related to $\{(X_n, Y_n)\}$, form a univariate renewal processes. Denote by

$$\begin{cases} N_x^{(1)} = \max\{n : S_n^{(1)} \leq x\} \\ N_y^{(2)} = \max\{n : S_n^{(2)} \leq y\} \end{cases} \quad (3)$$

their counting processes. It follows from (2) and (3) that

$$N_{x,y} = \min\{N_x^{(1)}, N_y^{(2)}\}. \quad (4)$$

In the automotive warranty context, the point process $N_x^{(1)}$ is the age-wise renewal process, generated by the X_i 's, say with c.d.f. $F_X(x)$. Similarly, $N_y^{(2)}$ is the mileage-wise renewal process, generated by the Y_i 's, say with c.d.f. $F_Y(y)$. It is well known that in the univariate case

$$\begin{cases} N_x^{(1)} \geq n \iff S_n^{(1)} \leq x \\ N_y^{(2)} \geq n \iff S_n^{(2)} \leq y. \end{cases} \quad (5)$$

Therefore, in the two-dimensional case we have

$$N_{x,y} \geq n \iff (N_x^{(1)} \geq n) \cap (N_y^{(2)} \geq n) \iff (S_n^{(1)} \leq x, S_n^{(2)} \leq y). \quad (6)$$

Bivariate convolution. Recall that (Hunter, J. J. 1974), given two Stieltjes integrable c.d.f.'s $F(\cdot, \cdot)$ and $G(\cdot, \cdot)$ of non-negative random variables, their convolution is defined as

$$F ** G(x, y) = \int_0^x \int_0^y F(x-u, y-v) dG(x, y). \quad (7)$$

Let us denote by

$$F_{n+1}(x, y) = F ** F_n(x, y), \quad (8)$$

where $n = 1, 2, \dots$ and $F_1(x, y) = F(x, y)$. Then, the c.d.f. of $S_n = (S_n^{(1)}, S_n^{(2)})$, is $F_n(\cdot, \cdot)$.

The distribution of $N_{x,y}$. Using (6), (7) and the notation (8), it follows that

$$P\{N_{x,y} \geq n\} = F_n(x, y).$$

Therefore, the distribution of $N_{x,y}$ is given by

$$P\{N_{x,y} = n\} = F_n(x, y) - F_{n+1}(x, y). \quad (9)$$

The two-dimensional renewal function. The expectation of the bivariate counting process $N_{x,y}$

$$M(x, y) = EN_{x,y} \quad (10)$$

is called the two-dimensional renewal function and it can be expressed as follows:

$$M(x, y) = EN_{x,y} = \sum_{n=1}^{\infty} nP\{N_{x,y} = n\} = \sum_{n=1}^{\infty} P\{N_{x,y} \geq n\} = \sum_{n=1}^{\infty} F_n(x, y). \quad (11)$$

Two-dimensional renewal equation. The renewal function $M(x, y)$ satisfies the two-dimensional renewal equation

$$M(x, y) = F(x, y) + \int_0^x \int_0^y M(x-u, y-v) dF(u, v) \quad (12)$$

or

$$M(x, y) = F(x, y) + \int_0^x \int_0^y F(x-u, y-v) dM(u, v). \quad (13)$$

It is easy to derive the renewal equation (12). Indeed, using (11) and (8), we have

$$\begin{aligned} M(x, y) ** F(x, y) &= \sum_{n=1}^{\infty} F_{n+1}(x, y) = F(x, y) - F(x, y) + \sum_{n=1}^{\infty} F_{n+1}(x, y) \\ &= M(x, y) - F(x, y), \end{aligned} \quad (14)$$

which in view of (7) leads to (12).

For more details on the two-dimensional renewal function, see (Hunter, J. J. 1974).

3 Numerical Approximation of the Renewal Function $M(x, y)$

In most situations it is not possible to find an analytical solution to $M(x, y)$. A possible numerical solution for the evaluation of the renewal function in the one

dimensional case, is the Riemann Stieltjes Method, proposed by (Xie, M. 1989). We shall now provide an extension of this method for two dimensions.

We start with the renewal equation as in (13). First, let us consider the standard midpoint approximation for a definite integral being

$$\int_a^b f(x)dg(x) \approx \sum_{i=1}^n f(x_{i-1/2})(g(x_i) - g(x_{i-1}))$$

where

$$x_{i-1/2} = (x_i + x_{i-1})/2.$$

Now, if we apply the same idea in two dimensions, we get the approximation

$$\begin{aligned} \int_a^b \int_c^d f(x, y) dg(x, y) \approx \\ \sum_{i=1}^n \sum_{j=1}^m f(x_{i-1/2}, y_{j-1/2})(g(x_i, y_j) - g(x_{i-1}, y_j) - g(x_i, y_{j-1}) + g(x_{i-1}, y_{j-1})). \end{aligned} \quad (15)$$

Next, we consider two partitions being $x_i, i = 0, 1, \dots, n$ and $y_j, j = 0, 1, \dots, m$ where $0 = x_0 < x_1 < \dots < x_n = x$ and $0 = y_0 < y_1 < \dots < y_m = y$. Using (13) we have

$$M(x_i, y_j) = F(x_i, y_j) + \int_0^{x_i} \int_0^{y_j} F(x_i - u, y_j - v) dM(u, v) \quad (16)$$

Applying the midpoint approximation (15), we get

$$\begin{aligned} M(x_i, y_j) \approx F(x_i, y_j) + \sum_{k=1}^i \sum_{l=1}^j \\ F(x_i - x_{k-1/2}, y_j - y_{l-1/2})(M(x_k, y_l) - M(x_{k-1}, y_l) - M(x_k, y_{l-1}) + M(x_{k-1}, y_{l-1})) \end{aligned}$$

Hence, if we rearrange this formula and solve for $M(x_i, y_j)$, we get

$$\begin{aligned} M(x_i, y_j) = \\ \frac{F(x_i, y_j) + S_{ij} - F(x_i - x_{i-1/2}, y_j - y_{j-1/2})[M(x_i, y_{j-1}) + M(x_{i-1}, y_{j-1}) - M(x_{i-1}, y_{j-1})]}{1 - F(x_i - x_{i-1/2}, y_j - y_{j-1/2})} \end{aligned} \quad (17)$$

where we define

$$\begin{aligned} S_{ij} = \sum_{k=1}^{i-1} \sum_{l=1}^{j-1} \{F(x_i - x_{k-1/2}, y_i - y_{l-1/2}) \\ [M(x_k, y_l) - M(x_{k-1}, y_l) - M(x_k, y_{l-1}) + M(x_{k-1}, y_{l-1})]\} \end{aligned}$$

for $i = 2, 3, \dots, n$, $j = 2, 3, \dots, m$ and $S_{ij} = 0$ otherwise. Therefore, if we set $M(0, 0) = 0$ and assume that $M(0, y)$ and $M(x, 0)$ are known, we can compute $M(x, y)$ recursively using (17). In addition, if we set the x -partition to be regular, and do likewise with the y -partition, then the above numerical procedure becomes a relatively easy algorithm to implement.

In the one dimensional case, this form of the procedure is fairly satisfactory as in most cases the function $F(x)$, that is the cumulative distribution function, is readily

available. However this is not the case for many two dimensional distributions, where often only $f(x, y)$, the probability density function of the distribution, is available. Hence, how we use $f(x, y)$ to approximate $F(x, y)$ becomes very important. As a very simple approximation we suggest using

$$F(x_i, y_j) = f(x_{i-1/2}, y_{j-1/2}) \cdot (x_i - x_{i-1}) \cdot (y_j - y_{j-1}) + F(x_{i-1}, y_j) + F(x_i, y_{j-1}) - F(x_{i-1}, y_{j-1})$$

the first term being an estimate of the probability in the square with top right corner at point (x_i, y_j) and bottom left corner at (x_{i-1}, y_{j-1}) . This is a fairly basic approximation. However, if the x - and y -partitions are fine enough, this should be a reasonable estimate to use.

This numerical procedure can be extended for the case when only a sample from the renewal process is available, i.e., none of $F(x, y)$ or $f(x, y)$ is available. In this case, the empirical distribution function, computed over the data, can be used to approximate for $F(x, y)$.

4 Testing for a Two-Dimensional Renewal Process

We used the two-dimensional renewal process to model the failure process of a product, which in our illustrative example is an automobile. The sequence $\{(X_n, Y_n)\}$, $n = 1, 2, \dots$, represents the virtual age and the virtual mileage of the car at the time of a warranty claim, assuming that after the repair the car is “as good as new”, i.e., X_n and Y_n are the age and the mileage accumulated between the n^{th} and the $(n + 1)^{\text{th}}$ repairs, $n = 1, 2, \dots$, and $(X_0, Y_0) = (0, 0)$. We assume that these are i.i.d. random variables with a joint c.d.f. $F(x, y) = P\{X_n \leq x, Y_n \leq y\}$, i.e., they are stationary and independent. We have to test whether our data satisfy these assumptions. Testing for stationarity means to show that no trend can be identified and testing for independence means to show that there is no correlation between successive values of the two-dimensional failure data.

To the best of our knowledge, no general test for a two-dimensional renewal processes is available. We test for the 2D-renewal process by using the univariate tests on the marginal sequences $\{X_n\}_1^\infty$ and $\{Y_n\}_1^\infty$. If these univariate tests do not reject the stationarity of the univariate processes, we conclude that there is no evidence to reject the stationarity of the two-dimensional sequence $\{(X_n, Y_n)\}_1^\infty$. We test the independence of $\{(X_n, Y_n)\}$ using a multivariate test, developed by Puri and Sen (Gieser, J. J. and R. H. Randles 1997).

We illustrate the tests discussed in section 4.1 on a sample of warranty records for a particular year 2000 model vehicle. The sample size is $n = 21$.

4.1 Univariate Tests for Stationarity

Graphical method. Plotting the cumulative number of failures against the cumulative age and mileage respectively, a non-linear pattern of the plots shows a variation of the failure rate. A concave down graph corresponds to a deteriorating product, while a concave up graph corresponds to an improving product. The plots of our failure data have fairly linear patterns, as shown in Figure 1. Therefore, no trend is detected.

Fisher-Gnedenko test. Ascher and Feingold (Ascher, H and H. Feingold 1984) provide a test for Homogeneous Poisson Process (HPP) for univariate data that has

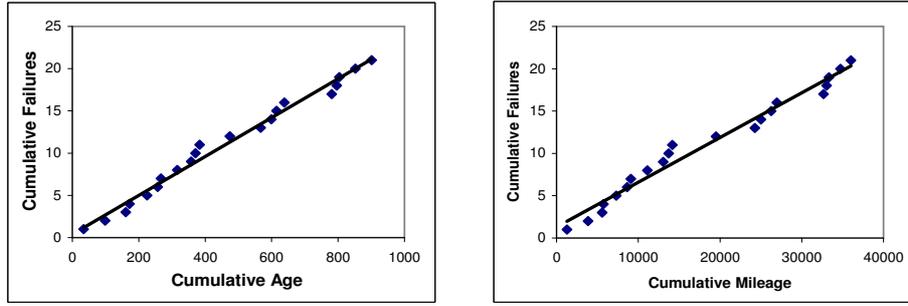


Figure 1: Graphical Method

been developed by Fisher and adapted by Gnedenko et al. (Gnedenko, Belyayev, and Solovyev 1969). To express the likelihood of a very long interarrival time X_i , as an alternative of the exponential distribution, we compute

$$\eta_n = \frac{\max_{1 \leq i \leq n} X_i}{S_n}. \quad (18)$$

The probability that the statistic η_n exceeds the critical value $g_\alpha(n)$ for certain α is

$$P\{\eta_n > g_\alpha(n)\} = \alpha. \quad (19)$$

The values of $g_\alpha(n)$, for $\alpha = 0.05$ and different n 's are tabulated in Gnedenko and al. (Gnedenko, Belyayev, and Solovyev 1969). If the computed value of η_n is larger than $g_\alpha(n)$, we reject HPP hypothesis.

Laplace test. It tests the null hypothesis of HPP against the alternative hypothesis of monotonic trend (Ascher, H and H. Feingold 1984). Under H_0 , the failure rate is a constant and the times of the failures S_1, S_2, \dots, S_{n-1} are the order statistics of a sample from an uniform distribution on interval $(0, S_n]$. Thus, the statistic

$$U = \frac{\frac{\sum_{i=1}^{n-1} S_i}{n-1} - \frac{S_n}{2}}{S_n \sqrt{\frac{1}{12(n-1)}}} \quad (20)$$

is approximately standard normal.

The Military Handbook Test. This test (Ascher, H and H. Feingold 1984), often referred to as the MIL-HDBK test, is for a HPP against a Non-Homogeneous Poisson Process (NHPP). The test statistic is

$$\mathcal{U} = 2 \sum_{i=1}^{n-1} \ln \frac{X_n}{X_i}, \quad (21)$$

and $\mathcal{U} \sim \chi_{2(n-1)}^2$ distributed.

The Mann-Whitney test. This is a nonparametric rank test (Ascher, H and H. Feingold 1984), having no other assumption but that the interarrival times X_1, X_2, \dots, X_n are i.i.d random variables. Let us call *reverse arrangement* an arrangement such that $X_i < X_j$ for $i < j$. Comparing each X_i to later interarrival times, $\frac{n(n-1)}{2}$ comparisons are possible. If no trend is present, then the total number of reverse arrangements, say \mathcal{R}_n , should have the expected value

$$E[\mathcal{R}_n | \text{renewal}] = \frac{n(n-1)}{4}, \quad (22)$$

i.e., half of the number of possible reverse arrangements. The variance of \mathcal{R}_n is

$$\text{Var} [\mathcal{R}_n | \text{renewal}] = \frac{2n^3 + 3n^2 - 5n}{72}. \quad (23)$$

Using (22) and (23), the test statistic is

$$Z = \frac{\mathcal{R}_n - E [\mathcal{R}_n | \text{renewal}]}{(\text{Var} [\mathcal{R}_n | \text{renewal}])^{1/2}}, \quad (24)$$

which is approximately standard normal.

The Lewis-Robinson test. This is a generalization (Ascher, H and H. Feingold 1984) of the Laplace test, under the same null hypothesis of HPP. It is obtained by dividing the statistic (20) of the Laplace test by the estimator of the coefficient of variation of the random variable X

$$\widehat{CV}[X] = \frac{(\widehat{\text{Var}}[X])^{1/2}}{\widehat{E}[X]}, \quad (25)$$

where X is distributed as X_i . The Lewis-Robinson statistic is

$$U_{LR} = \frac{U}{\widehat{CV}[X]}. \quad (26)$$

If X is exponentially distributed, then $\widehat{CV}[X] = 1$ and statistics (20) and (26) are asymptotically equivalent, i.e., U_{LR} is approximately standard normal.

4.2 Testing the Independence

For univariate data, computing the correlation coefficient of lag 1 is the most handy technique for assessing the independence. We applied this method to the marginal sequences of the age X_n and mileage Y_n and found that there is no correlation of lag 1 between the “intervals” between failures.

To test for independence of the bivariate data we applied a test proposed by Puri and Sen (Gieser, J. J. and R. H. Randles 1997). That is, if $W_i^{(k)}$, $k = 1, 2$ are continuous $r_k \times 1$ random vectors, let $\{W_i \equiv (W_i^{(1)}, W_i^{(2)})', i = 1, \dots, m\}$ be a sample of m pairs. The test uses the matrix $T = \begin{pmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} \end{pmatrix}$ with

$$T_{s_1 s_2} = \frac{1}{m} \sum_{\alpha} \text{sgn} \left(W_{s_1 \alpha}^{(1)} - \widetilde{W}_{s_1}^{(1)} \right) \text{sgn} \left(W_{s_2 \alpha}^{(2)} - \widetilde{W}_{s_2}^{(2)} \right), \quad (27)$$

where $\widetilde{W}_{s_1}^{(k)}$ is the median of $W_{s_k 1}^{(k)}, \dots, W_{s_k m}^{(k)}$, for computing the test statistic

$$S^J = \frac{|\mathbf{T}|}{|\mathbf{T}_{11}| |\mathbf{T}_{22}|}. \quad (28)$$

Under H_0 , we have that

$$-m \log(S^J) \xrightarrow{d} \chi_{r_1 r_2}^2.$$

5 An Example

5.1 Testing for stationarity and independence

The cumulative ages and mileages corresponding to 21 warranty claims are given in Table 1. We applied the tests listed in Section 4 to this sample of warranty records. For the univariate tests the results are summarized in Table 2.

Age:	34	99	161	172	225	257	267	316	358	371	
	383	474	567	599	615	638	781	796	804	852	901
Mileage:	1272	3864	5593	5747	7315	8645	9093	11102	13055	13703	
	14176	19519	24257	25000	26267	26942	32659	33030	33321	34731	36006

Table 1: Example: Two-dimensional warranty data for a single vehicle

All of the univariate tests, except Mann-Whitney's test for the age data, do not reject the assumption of stationarity of the data. These univariate results gave us a reason not to reject the assumption of stationarity of the bivariate data.

As we mentioned earlier, for univariate data, computing the correlation coefficient of lag 1 is the most handy technique for assessing the independence. We applied this method to the marginal sequences of the age X_n and mileage Y_n and found that there is no correlation of lag 1 between the "intervals" between the failures.

Therefore, the two univariate processes, formed by the age data and mileage data in Table 1 can be assumed to be renewal processes.

Next, we test the bivariate data for independence using Puri's and Sen's non-parametric test. The value of the test statistic is

$$-m \log(S^J) = 4.007$$

and it is based on χ_4^2 -distribution with 4 degrees of freedom. If the significance level of the test is chosen to be $\alpha = 0.05$, then the rejection region is $R : \{\chi_4^2 > 9.488\}$ and we do not reject the H_0 hypothesis claiming the independence of the bivariate data.

Therefore, there is no evidence that the observed failure process is not a 2D-renewal process and we utilize results from the 2D-renewal theory for our simple warranty cost analysis.

Test	H_0	Critical region	Marginal results		Reject H_0 ?
			X	Y	
Plot	No trend		linear	pattern	No
Fisher-Gnedenko	HPP	> 0.2704	0.1587	0.1588	No
Laplace	HPP	outside of $[-1.96, 1.96]$	-0.207	-0.232	No
MIL-HDBK	HPP	> 55.76	38.23	40.07	No
Mann-Whitney	i.i.d.	outside of $[-1.96, 1.96]$	2.753	1.781	Yes/No
Lewis-Robinson	HPP	outside of $[-1.96, 1.96]$	-0.265	-0.2448	No

Table 2: Summary of the univariate tests

5.2 Estimation of the renewal function $M(x,y)$

Next, we use the numerical procedure proposed in Section 3 to estimate the renewal function of the renewal process generated by our sample. The algorithm is set up

so that if $M(x, y)$ is the targeted value of the renewal function, partitions of size 100 over the intervals $[0, x]$ and $[0, y]$ are used for the computations. The results are displayed in Table 3.

Miles \ Age	100	200	500	901
5000	2.14575	2.75538	2.75544	2.75544
10000	2.20272	4.56256	5.66326	5.66326
20000	2.20272	4.58983	11.14162	11.41601
36006	2.20272	4.58983	11.8316	20.34957

Table 3: Estimated values of the renewal function

These results can be used to estimate, from the producer's point of view, the expected warranty expenses associated with this particular vehicle. If the average cost to the manufacturer to produce this vehicle is, say in average c_M dollars, then, under our warranty model, the average warranty expenses of this particular vehicle will accumulate to

$$(20.34957 \times c_M) \text{ dollars.}$$

It is easy to see, that this vehicle will leave the warranty coverage due to mileage accumulation. Indeed, its last claim is at 36006 miles, which is just above the warranty mileage limit, whereas its age at the time of this claim is less than the warranty age limit.

6 Conclusion

We have discussed an application of the two-dimensional renewal theory in warranty analysis. We considered the warranty records of a particular vehicle and tested whether the assumption, that the warranty claim process is a renewal process, is acceptable. We used univariate tests to check for stationarity of the data and a multivariate test to check for independence. We propose a numerical procedure to evaluate the renewal function of the renewal process generated by our data and concluded the study with brief warranty cost analysis.

There are many open problems related to statistical inferences for bivariate data. In order to study the issues related to two-dimensional warranty and be able to use two-dimensional stochastic processes to model the warranty claim process, we need to address in future research, some of the following questions: how to test a bivariate stationarity, how to test for a bivariate Non-Homogeneous Poisson Process and so on.

References

- Ascher, H., and H. Feingold. 1984. *Repairable System Reliability*. Marcel Dekker.
- Blischke, W.R., and Murthy, D.N.P. 1996. *Product Warranty Handbook*. Marcel Dekker.
- Gieser, J. J., and R. H. Randles. 1997. "A Nonparametric Test of Independence Between Two Vectors." *Journal of the American Statistical Association* 92:561–567.

- Gnedenko, Belyayev, and Solovyev. 1969. *Mathematical Methods of Reliability Theory*. Academic Press.
- Hunter, J. J. 1974. "Renewal Theory in Two Dimensions: Basic Results." *Adv. Appl. Prob* 6:376–391.
- Murthy, D.N.P., Iskandar, B.P., and Wilson R. J. 1995. "Two-Dimensional Failure-Free Warranty Policies: Two-Dimensional Point Process Models." *Operations Research* 43 (2): 356–366.
- Xie, M. 1989. "On the solution of renewal-type integral equations." *Communications in Statistics - Simulation* 18:281–293.

Power to the people - improving the quality of information in the census

John Paynter and Gabrielle Peko
Department of Information Systems and Operations Management
University of Auckland, New Zealand
j.paynter@auckland.ac.nz; g.peko@auckland.ac.nz

Abstract

Like the 2005 General Election that preceded it, Census 2006 was promoted as the largest logistics exercise in the country. A complex timetable for finance, suppliers, personnel, processes and training was devised. In this paper we analyse how to identify processes and the quality of information collected might be improved.

For the most part, although the technical side of the electronic census worked, its uptake by citizens was low. The eforms themselves worked but the supporting information systems failed. The census was completed on the back of considerable goodwill on the part of the field staff.

Information and services provided on-line via by governments is constantly undergoing evolution partly driven by innovations in Information Technology (IT), partly by government wishing to leverage this tool and to a minor extent by user demand (citizen power). Although the uptake by citizens to undertake the census online option was poor, lessons were learnt that will improve the participation in the next census. Given the problems in contacting households, in particular in high rise apartments and walled communities, the Internet offers a viable option.

1 Introduction

A census is an official count. It can be contrasted with sampling in which information is only obtained from a subset of a population. As such it is a method used for accumulating statistical data, and it is also vital to democracy (voting). Census data is also commonly used for research, business marketing, and planning purposes. In New Zealand a census is held every five years. It is a snapshot on the chosen day when the number of people and dwellings (houses, flats, apartments) counted. Everyone in the country on that day is asked to complete census forms. There are two census forms. The blue Individual Form must be completed by everyone in your household on census day. The brown Dwelling Form must be completed by one person in your household. For the 2006 census an option was introduced to complete the forms on the Internet. Other initiatives included sending text messages about this process, amongst other things to the enumerators (collectors) whose job it is to collate the information in the field.

Information technology, especially the Internet, opens possibilities of using methods to distribute information and deliver services on a much grander scale (Paynter and Fung, 2006). It can deliver government services and encourage greater democracy and engagement from citizens. Governments around the world are exploring the use of web-based information technology (Grönlund, 2002).

Since the mid-1990s governments have been tapping the potential of the Internet to improve and governance and service provision. "In 2001, it was estimated that globally there were well over 50,000 official government web sites with more coming on-line daily. In 1996 less than 50 official government homepages could be found on the world-wide-web." (Ronaghan, 2002).

Along with the rapid growth of technological developments, people demand high quality services that reflect their lifestyles and are accessible after normal office hours from home or work. Thus, the goals of delivering electronic government services are to simplify procedures and documentation; eliminate interactions that fail to yield outcomes; extend contact opportunities (i.e., access) beyond office hours and improve relationships with the public (Grönlund, 2002).

2 Background

Census-taking began in China and the Middle East. One of the earliest recorded censuses took place in the Babylonian Empire nearly 6,000 years ago. Early censuses are mentioned widely in early Middle Eastern literature, with references to them in a number of places in the Bible.

Censuses of population were first taken in England and Scotland in March 1801, Ireland in 1811 and Australia in 1828. In the USA the census is undertaken every 10 years. The US Census 2000 project spanned 13 years at a cost of \$65 billion (Gido and Clements, 2006 p147). It is largely based on mailbacks with census employees personally visiting non-respondents. The first New Zealand census was undertaken in 1851, although this census excluded Māori (Statistics New Zealand, 2006a).

In New Zealand several acts of parliament have formed the legal basis for the collection of statistical data and census taking that has developed over the years. The most recent of which is the Statistics Act 1975. It clarified that the information contained in returns is to be used for statistical purposes only. It also specified which particulars it is mandatory to collect in the census and which particulars are able to be collected if the Government Statistician considers it in the public interest to do so. It also guaranteed the census to be free of government influence.

2.1 The use of technology

The 1921 Census marked the first occasion on which automatic sorting and counting machines were employed in New Zealand, enabling the major portion of census compilation to be carried out mechanically. The system installed for this census was purchased from the United States, which had been employing mechanical tabulation for census work since 1870.

For the 1966 Census, sorting machines were replaced by computers. Statistical tables were also produced by computer for the first time and results became available much earlier with a large number of additional cross-classifications of the census data being possible. The use of punchcards for each individual and dwelling was continued until 1976 when an automatic, electronically-based system was introduced. Mechanical tabulation has been replaced by electronic data capture and handling as the speed and capacity of computing technology has improved. In 1996 the scanning and imaging of census forms was introduced, further demonstrating that Statistics New Zealand was now fully immersed in the era of information technology, with analytical tools and information at a level incomprehensible to the department of earlier years.

2.2 Enumeration

In the 2001 census though the process of distributing and collecting forms (enumeration) had hardly been changed. Enumerators within each district would hand deliver forms to each household (one dwelling form and one individual form for each person expected to be present on census night). Each form would be coded with an identifier made up of District, Subdistrict, Meshblock and Dwelling (the individual forms had Person ID added on collection). This ID was recorded in a field book along with any comments including the address and best pick up time. After Census night the households would be visited again to collect the completed forms. Up to three visits would be made in each of the delivery and collection phases. On the third unsuccessful visit prior to census night a default number of forms (one dwelling, three individual) would be left. After census night an envelope would be left on the third unsuccessful collection visit. At the end of the enumeration phase the District Supervisors would send follow up letters and / or visit non-respondents. Once District offices were closed five weeks after the census, the central census office would follow up non-respondents.

3 Method

The authors had worked in the 2002 and 2005 General Election as Polling Place Managers, trainers and team-leaders in the official count, both having participated in other roles in previous (pre-MMP) elections. In addition they had undertaken research on the potential for electronic elections (Paynter and Peko, 2005).

For the 2006 census one author worked as the Grey Lynn District Supervisor. Grey Lynn represents a typical inner city district. It has a few high density apartments, areas of flatting and typical single family dwellings as well as a range of non-private Dwellings (NPDs). The 15 subdistricts represent a cross-section of these, from traditional suburban houses having a low difficulty rating (1) to high density difficult enumerations (difficulty, 3). Thus the observations made are taken from a detailed analysis of a single urban district (one of over 400) and the overall patterns from the country as a whole.

Census 2006 would see the introduction of two technological innovations. The first was the adoption of Internet-based census forms (Statistics NZ, 2004). This would enable the dwelling and individual forms to be submitted electronically via the Internet by the householder and individuals. The second, in part necessitated by the first, was to automate the flow of information about the forms submitted either electronically or via post to the enumerators (collectors). On the basis of the Internet ID on the census form the enumerators were texted via webmail to their census cellphones with the details of forms submitted.

3.1 On-Line Census Forms

On delivery of the census forms to the household the enumerator would ask the “Hi-Five” questions. These included whether or not members of the household would like to submit their forms on-line. If one or more individuals indicated that they might want to do this, then they were given an Internet PIN for everyone in the household to use. The household ID (District, Subdistrict, Meshblock and Dwelling) forms the Internet ID for the entire household. A sealed slip was given out to the household. This contains the PIN. Although everyone within the household used this one combination of ID and password, their information would still be secure as each individual’s had to be entered within the one session. That is, a session could not be

saved and resumed. However the different individuals could enter their information at different times. Once each dwelling or individual form was completed the system would batch the submissions and text the enumerator on a trice-daily schedule. They would get a cumulative report of the number of forms of each type submitted for each household.

3.2 Text Messaging

The computer Field management system (FMS) records the contract and contact details of each of the enumerators. With the exception of some of the high-density apartment buildings in the CBDs each subdistrict is assigned a single enumerator. The census cellphone number of each enumerator is recorded whether it be their own phone (for which an allowance is paid) or one provided for the census. During the census period FMS can text messages to the enumerators. This is done on the basis of the Internet ID (District + Subdistrict) where the message is sent in reference to a pre-coded form or the address when initiated by an individual who perhaps does not have a form. These ACTION messages comprise one of four types. An instance of each type is shown below.

0010498,6320201001W,3 Bombay,NOTIFY,(I:1e),dispatched more forms
 0010499,6320201,5 Bombay,ACTION,,housesitters staying here; deliver forms
 0010503,6320201006F,11 Bomba,INFO,,Destiny Child will need assistance
 6320201004P,9 Bombay,OFFICE,(D:1,I:3,A:1)

The first part is the unique message number; this is followed by the Internet ID (632=District, 02 = Subdistrict, 01 = Meshblock, 001 = Dwelling, W = Checkletter; this is followed by the address; the message type (NOTIFY, ACTION, INFO, OFFICE); the type (D = Dwelling, I = Individual) and number of forms; and lastly any textual information. The forms may be either in English (e) or Maori (m) – the two official languages in New Zealand.

The enumerators were required to enter any messages in their field book against the line for that particular dwelling. In the case of ACTION messages they were to deliver the required number of forms to the dwelling. OFFICE messages denote that the forms have been received via the Internet (online submission) or have been mailed. The enumerator would update the forms received in the OFFICE column of the field book and could check this against those delivered to see if any further forms needed collection for that address.

These messages were also available in the action log. The district supervisor could print these to check against the enumerator's field book when they made field checks and when the enumerator brought the field book in with the boxes containing the forms they had picked up at the end of the field phase of the census.

4 Results

4.1 District collection statistics

The collection statistics for different sub-districts can be compared to see if the appropriate amount of effort is being put into dissimilar (in terms of difficulty) sub-districts. The collectors (enumerators) are recompensed based on the expected number

of dwellings and the degree of difficulty. One way to gauge if the effort allocation is correct is by looking at the collection percentage (the number of dwelling forms collected divided by the number expected, itself being determined by the number of dwelling forms delivered).

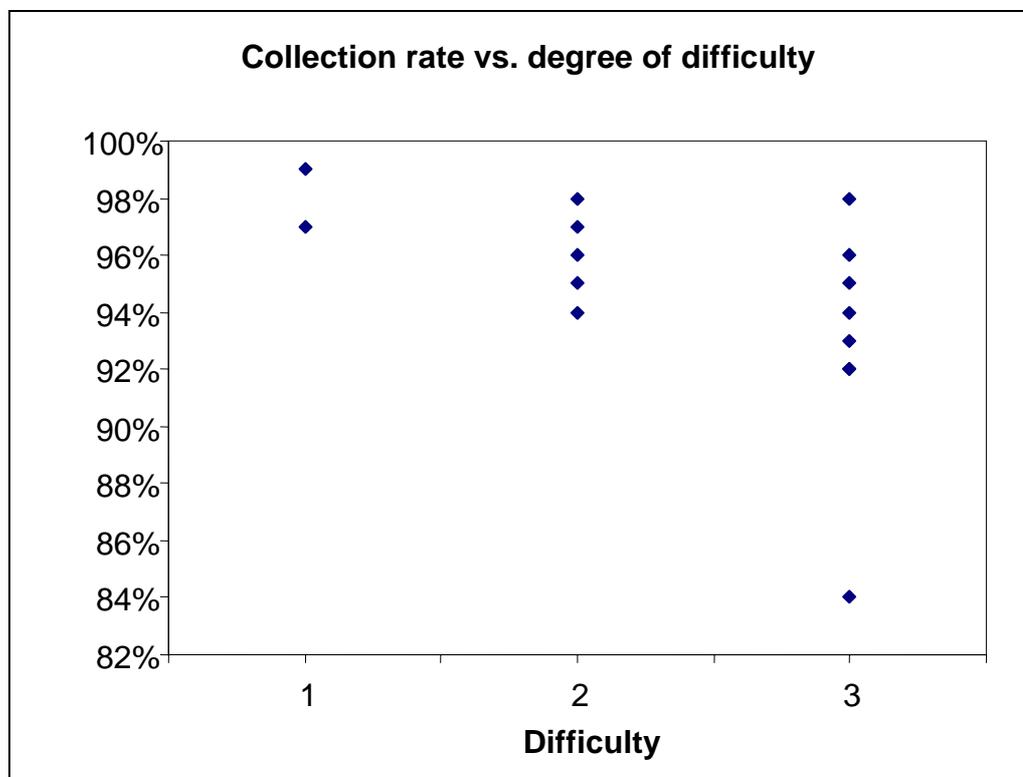


Figure 1: Subdistrict collection statistics based on difficulty

The sample size is too low to test for significance but it would appear that the collection rate depends on the degree of difficulty. That is, the more difficult subdistricts achieve a lower rate. From this it can be seen that at the micro (district) level appropriate levels of effort are being applied. One subdistrict clearly has the worst result and within that one meshblock (75%). This reflects the district's demographics.

4.2 Fieldbooks

The use of the field books was error prone. Enumerators could wander out of a meshblock and even into another subdistrict or district. Such recording errors would have to be corrected in later phases of the census. This would be very hard to do if it did not occur at the district level. For instance if the online option was taken there was no mechanism to the District Supervisor to correct the return – it could only be logged as an enumeration event.

4.3 e-forms

Trials conducted a year before suggested a 20% uptake of the electronic census form option (Statistics New Zealand, 2006). Field staff were instructed not to oversell this option as there were fears that 30% uptake would be too much for the computer system. During the delivery phase staff reported a high uptake in some areas (as much as 80%, particularly in inner city suburbs and the CBD). Although the forms were to be completed as of census night (March 7th), the system was available from 20

February (the beginning of the delivery phase) till the end of March. The peak period though was on census night and few problems were encountered with the use of the Internet forms.

At the end of the field phase only 8% of the completed forms had been received via the Internet option. The author questioned people during the collection phase of the census (when doing doorstep checks). Many of those who had requested PINs so that they could do the census online stated that as they had the paper-based form (given out both to record the Internet ID, and as a back up in case the electronic version failed) they found it easy to complete the paper form, particularly with family members sitting around the table. Conversely in the large flatting situations typified in some suburbs, only having a single PIN made it difficult for those who would elect to complete the forms at work and other places where they had Internet access. Another situation again arose in the case of Non-Private Dwellings (NPDs) such as hostel, hospitals and hotels where each individual was given a separate PIN. However the uptake in such places tended to be low.

4.4 Field Management System (FMS)

However some of the OFFICE messages to the enumerators were not received during the peak census period. It should be emphasised that the forms themselves were received in the census system. There were also performance and other problems with the Field Management System (FMS), especially considering that access to it from the district offices scattered throughout the country was via dial-up lines (Jackson, 2006). The enumerators texted, emailed or phoned their daily delivery and collection statistics to the district supervisors. Although these were to be entered in FMS problems arose and a largely manual system was used. This resulted in further time delays in receiving information. Thus it would be hard say to realise more staff were needed when delivery and collection were encountering problems.

4.5 mailbacks

In contrast over 10% of census forms had been mailed in. These and the ones collected by the enumerators and district supervisors had to be dispatched to Christchurch to be scanned. Some preliminary results for the census (e.g. the overall population count, based on analysis of the field books) were available at the end of May (Statistics, 2006b), but it would take three months for the forms to be scanned. Full results are now scheduled to be released on December 6.

Mailback forms were received at the central office in Christchurch and the district offices. Each of these was to be entered in the system so that FMS could text the enumerator with the corresponding OFFICE message. This was not feasible in the District offices and the information entered at the central office lacked addresses (omitted to cut down the time it took), so it was difficult to determine the correct dwelling when the forms had been miscoded.

5 Discussion

5.1 E-census

Lessons were learnt that will improve the participation in the next census. The risks of the online version were recognized in the trial held a year earlier. Emphasis was made on training of the field supervisors and the collectors. The collectors were given access to the online census facility, to enable them to become familiar with the online option before it went live. This was invaluable as an aid to their understanding of

how the option would be viewed by respondents prior to standing at their doors and offering that option (Statistics, 2006) Additional training was also introduced for the field collectors to ensure they understood what is required of the respondent, regardless of which option is chosen. This was to ensure, firstly, that the online option is communicated correctly to the respondent on the doorstep; and, secondly, that the field communication systems integrating the online option with the paper option were correctly performed. The training theme continued with field supervisors, who need to be able to use the field operations monitoring systems. These are the mechanisms used to monitor how the team in the field responds to the text messages they receive about the lodgement of Internet forms and related helpline actions.

The complexity of these systems and the need to integrate with existing systems for the paper form collection process posed a real challenge to ensure seamless operations. Unfortunately the failure of FMS (Jackson, 2006) meant that this integration was not a success. The field team became distrustful of the accuracy of the information provided about forms collected by the online and mailback options. This made their jobs harder in the collection phase and then when the collectors returned the forms. Information about mailbacks, privacy envelopes, e-forms and hand-collected forms had to be combined. Subsequent to this follow-up letters were sent to apparent non-respondents but again the possibility existed that the forms had been received by mailback or electronically but that this was not communicated to field staff

5.2 E-government

It has been claimed that the low penetration of broadband technology in New Zealand is a limitation to the spread of electronic services. However the unbundling of the Telecom local loop monopoly was released prior to the 2006 Budget (Budget, 2006). Clearly the infrastructure is in place for the participation of New Zealand citizens in e-government. However the uptake by both the citizens, as shown in the census, and government agencies is low. For instance, only one council, Auckland Regional Council, provides an on-line forum for discussion and sharing of ideas (Paynter and Fung, 2006). None of the local government sites provide any e-democracy although some sites have put up information about Elections 2004. Dunayev (2005) used an automated tool to analyse all the local government web sites. He concluded that the sites did not appear to have matured sufficiently to meet the goal of online local government elections in the next cycle (2007). Some of the obstacles to e-voting, such as trust, are outlined in Sharkey and Paynter (2003) and steps towards an e-voting transition in Paynter and Peko (2005). This included the use of e-services in the census and in the local body elections – both less potentially risky and lower profile than a general election.

Other countries that have an e-census option include Switzerland where 4.2% of the population took part in the E-Census. 11% who began to fill out the form online interrupted the process before reaching the end. The E-Census homepage received 238,000 visits. Of those households that took a look at the Web site, only half actually filled out and submitted their questionnaires online. This shows the wide gap between simply searching for information over the Internet and carrying out a complex transaction online (Swiss Statistics).

5.3 Future Trends

The information and services provided on-line via by governments is constantly undergoing evolution partly driven by innovations in Information Technology (IT),

partly by government wishing to leverage this tool and to a minor extent by user demand. Although the uptake by citizens to undertake the census online option was poor, this can be improved as the Internet becomes more pervasive and lessons learnt from the 2006 census incorporated into the training and testing in 2011. Given the problems in contacting households, in particular in high rise apartments and walled communities, the Internet offers a viable option.

Use of the field books is cumbersome and error prone. There are also delays in communicating such information captured on hard copy paper. It would be advantageous to use an electronic notepad allied with a Global Positioning System (GPS) to minimise recording errors (such as failing to give the complete internet ID or moving out of the correct meshblock) and maximize the responsiveness to the delivery and collection statistics recorded. Surveylab (www.surveylab.co.nz) is an example of such a system used for GIS information (Stuff, 2006).

6 Conclusions

For the most part, although the technical side of the electronic census worked, its uptake by citizens was low. This would suggest that further moves towards e-democracy in the form of electronic voting would be premature in terms of public acceptance and uptake.

Acknowledgements

The authors wish to acknowledge the perseverance of the census staff, particularly those in the field in carrying out the enumeration process despite the failure of computer the support systems.

7 References

- Australian Bureau of Statistics (2006) Welcome to eCensus, Retrieved 26 September 2006 from <http://www.census.abs.gov.au/eCensusWeb/>
- Budget (2006) Government moves fast to improve Broadband. Retrieved 3 May 2006 from <http://www.beehive.govt.nz/ViewDocument.aspx?DocumentID=25636>
- Dunayev, A (2005) Electronic Local Government Elections in New Zealand. Unpublished BCom (Hons) dissertation, The University of Auckland, New Zealand.
- Gido, J. & Clements, J. (2006) Successful project Management, Thomson, Mason, OH (3rd Ed), p147.
- Grönlund, A. (2002). Electronic Government: Design, Applications & Management. Hershey: Idea Group Publishing.
- Jackson, R. (2006) Census system overload but no data lost, Computerworld, 20 March, 2006. Retrieved 3 May 2006 from <http://computerworld.co.nz/news.nsf/UNID/0A98536DAE494843CC257133007AEE8B?OpenDocument&Highlight=2,census>
- Paynter, J & Fung, M. (2006) 'E-Service Provision by New Zealand Local Government', In: Ari-Veikko Anttiroiko, & Matti Malkia, (ed.), *Encyclopedia of Digital Government.*, Hersey, Idea Group Publishing, p.-(forthcoming), 2006
- Paynter, J & Peko, G. (2005) e-elections and the price of democracy, Proceedings of the 40th Operations Research Conference of New Zealand, Wellington Dec 1-3, 2005, 145-154
- Sharkey, E. & Paynter, J. (2003) Factors influencing the uptake of Online voting in NZ. CHINZ '03: the 4th annual conference of the ACM Special Interest Group on

Computer-Human Interaction, New Zealand Chapter Dunedin. 3-4 July 2003, 121-122

Statistics New Zealand (2004) On-line Option for 2006. Census Retrieved 22 June 2006 from

<http://www2.stats.govt.nz/domino/external/pasfull/pasfull.nsf/web/Media+Release+2006+Census:+On-line+forms+October+2004?open>

Statistics New Zealand (2006a) Introduction to the Census (2001) - reference report. Retrieved 20 June 2006 from <http://www.stats.govt.nz/census/2001-census-technical-info/2001-introduction/default.htm>

Statistics New Zealand (2006b) 2006 Census of Population and Dwellings – Provisional Counts. Retrieved 29 May 2006 from

<http://www.stats.govt.nz/products-and-services/hot-off-the-press/2006-census/2006-census-provisional-counts-2006-hotp.htm>

Surveylab (2006) Customizing applications for ike, Retrieved 27 September 2006 from <http://www.surveylab.co.nz/~downloads/CUSTOMIZING%20IKE.pdf>

Swiss Statistics (2001) Experience with the E-Census Retrieved 26 September 2006 from

http://www.bfs.admin.ch/bfs/portal/en/index/themen/volkszaehlung/uebersicht/blank/zur_erhebung0/erfahrungen_mit_e-census.html

ⁱ If pop ups were disabled the forms could not be completed as there was a pop up confirmation message at the end of the submission stage. Some collectors had either not coded the forms at all or had omitted the check letter. Where there were more dwellings in a meshblock than expected, although an extra 20% loading was catered for, an overflow book had to be used and there was no corresponding Internet ID. i.e. all the check letters were coded as 'X'. In these circumstances the respondents could not use the Internet option for completing the forms.

Forecasting telecommunications demand for services with short-history data

Fernando Beltrán
Information Systems and Operations Management Department
The University of Auckland
Auckland, New Zealand
f.beltran@auckland.ac.nz

Lina María Gómez
Centro de Investigación de las Telecomunicaciones - CINTEL
www.cintel.org.co
Bogotá, Colombia
lmgomez@cintel.org.co

Pablo Maya
Centro de Investigación de las Telecomunicaciones - CINTEL
www.cintel.org.co
Bogotá, Colombia

Abstract

In this paper we use a mixed set of techniques to forecast changes in demand for telecommunications services over a short-term horizon. Because competition in telecommunications markets is present and strongly growing in almost every country, and the history of consumption of specific products may be short, techniques such as Hidden Markov Models (HMM) and Classification Trees have been adapted to help network planners and marketing departments meet their duties, even though products whose demand is to be forecasted may not have been around long enough. We adapt a two-level method, which recently appeared in the literature, to predict probabilities of short-term individual customer behaviour (the lower level, using HMM), and to feed such probabilities into a classification tree algorithm (the upper level) that exploits high-order interactions among error patterns. Results of simulations that compare the use of classification trees with and without HMM are presented.

Key words: Telecommunication forecasts, Hidden Markov Models, Classification Trees.

1 Introduction

When the business climate is competitive with new providers and technologies constantly arriving into the market, telecommunications providers face the challenge of having to plan their future operation in spite of not having reliable and enough information.

Providers of telecommunications services must forecast demand for access to and usage of their services as well as the required network updates to satisfy them. In addition, the increasing presence of new competitors, the introduction of new services and the fact that different technologies are able to support different uses and applications have multiplied the challenges to be met in the telecommunications sector.

A key work in the concerned literature, (Fildes 2002), classifies telecommunications forecasting models in two: models of established markets and model of newly created markets.

Models of established markets identify two stages. First stage consists of modelling usage (amount of consumed service) as a function of price and income conditional on access. Second stage is the modelling of access, which is modelled through individual choices made between alternatives. These models were extensively used to forecast access and demand for traditional public switched telephone network services. A whole range of models is presented in (Taylor, 1994).

Models for new markets encompass access and usage of new services, new uses of already established services and changes in usage patterns for established services. Such models are mostly concerned with estimating the potential for market development and user adoption and service diffusion. Fildes (2002) points at the inferior degree of development of new market demand models with respect to the highly developed models for more traditional telecommunications markets. Mouchart and Rombouts (2005) confirm the latter by asserting that only very few research questions on new telecommunication service demand have been answered.

Forecasting new service demand can be done of the history of the service in other markets where the service was introduced is used. Such is the approach by Islam, Fiebig y Meade (2002) who explore alternative ways to combine information from different markets (countries) to develop a diffusion model. In (Mouchart and Rombouts, 2005) a similar approach is found that uses cluster analysis.

The hybrid model used here is an adaptation of the model originally presented in (Cox and Popken, 2002). Short samples of historical data about service sales (connections and disconnections to services) are combined with other observable and non observable user characteristics in a two-level approach to demand forecasting. The model quantifies transition rates of individuals between states, which are defined to be combinations of service subscriptions. Classification trees are subsequently used to address three serious and related problems: combinatorial explosion of the number of states, mixed variable types and conditional independence.

A main feature of the hybrid approach is its use of short-term data – usually 6 o 18 months, which makes it an attractive alternative to methods that require long time series.

Section 2 describes the basic elements of the model, from the use of classification trees and Markovian models up to the introduction of Hidden Markov Models (HMM) for forecasting refinement. In Section 3 results obtained through the application of the model using a randomly generated database are presented and discussed. Section 4 concludes.

2 A hybrid model of forecasting

Based on information about the individual history of connections and disconnections to services, the model uses classification trees to define groups of users. A sample of the

type of historical data used is exhibited in Table 1. First two columns identify our basic data unit: an individually identified consumer at a given time period. Starting with column 3, each column refers to a service. For each data unit a value of 1 indicates the individual was a subscriber to the service (column) at the given time period. 0 indicates the individual was not a subscriber.

User	Period	Call Waiting	Voice Messaging	Caller ID	Call Forwarding	Internet Access
1	1	1	0	0	0	1
1	2	1	0	0	0	1
1	3	1	0	0	0	1
1	4	1	0	0	0	1
1	5	1	0	0	0	1
1	6	1	0	0	0	1
1	7	1	0	0	0	1
1	8	1	0	0	1	1
1	9	1	0	0	1	1
2	1	1	0	1	0	1
2	2	1	0	1	0	1
2	3	0	0	1	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
3	1	0	1	0	0	1
3	2	0	1	0	1	1
3	3	0	1	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 1. Service subscription history

A certain combination of 1s and 0s will put the user in a given group. Such combination is known as ‘user profile’. A Markov chain is used to model the evolution of the distribution of users across the combinations of service subscription, also known as states. Transition probabilities - the probability that the user changes his profile one period later - are estimated using the history of connections and disconnections.

A Markovian model and Classification Trees are employed in a two-level approach to generate a hybrid forecasting algorithm. Classification trees are used for multiple purposes; when several services whose consumption is not exclusive are modelled, the combinatorial nature of our definition of a state is likely to produce a large number states. Classification Trees are used to reduce the size of the state space so that only statistically significant states are considered. Also, the use of a Markov chain to model the evolution of the subscription levels assumes that transitions between states follow the Markovian property. Tests are then performed that refine the initial definition of states until the new, refined states become as richly described as to verify the conditional independence property.

Additional information that relates to a user’s behaviour and other characteristics might improve the quality of the forecast but cannot be directly observed. Such information is modelled as variables that describe several features linked to the consumption behaviour of the user. The provider is not usually able to obtain it directly from the user or from its database. A Hidden Markov Model can then be used to estimate the value of such variables to improve the quality of predictions.

2.1 Service classification

The hybrid model distinguishes between two main service groups: core and peripheral. A core service will have a greater relative importance than a peripheral service in the

estimation of demand forecasting. Classification trees are used to discriminate among services according to their relative importance. Core services are then used to define states. If services can be jointly consumed or purchased the combinations produced by “being subscribed” (1) or “being unsubscribed” (0) are the initial candidates to represent chain states. Classification trees can once more be used to reduced the potentially large number of states.

The use of classification trees for distinguishing between core and peripheral services is illustrated with an example. For instance the tree shown in Figure 1 is used to inquire about the likelihood of cancelling a subscription to Call Waiting. The tree starts at a root node that classifies the data units into those who did not cancel their Call Waiting service (1) during the observed historical period and those who did (0). The former are the 71.4% of the data observed and the latter are the 0.5%. 28.1% of the data correspond to situations when Call Waiting was actually added. Refining questions are asked such as whether who cancelled their Call Waiting subscription are subscribers to Caller ID or not. Figure 1 illustrates the outcomes of the process for three services.

From one of the bottom leaves of the tree in Figure 1 we can infer that 0.4% of those who would be likely to drop their Call Waiting subscription (lowest right leaf) are also subscribers to Voice Messaging (1), but have neither Three-way calling (0), nor Caller ID (0). So would 1.2% of those who do have Caller ID.

The example also illustrates an important feature of the use of classification trees: not all attributes are necessary to identify what makes a user likely to change his profile. In Figure 1 the fact that this classification displays questions about three services means that identifying those likely to drop their subscription to Call Waiting only requires information about their demand for the three additional services.

A seeming advantage of the use of classification trees is that users are fully identified (discriminated) into richly described classification groups.

The values obtained, such as the ones in Figure 1, are used to build connection and disconnection indicators. It is the transitions between time periods and not the state of the connection at a particular time that is used to define an indicator. Any connection indicator will take on a value of 1 if the user was no subscriber to a service at a time t , becoming a subscriber at $t+1$. In case there is no change in the user's connection status the indicator is 0. If a disconnection happens then the indicator will take a “don't care” value. Indicators for service disconnection are built in a similar way.

Core services are thus defined as those services that appear most frequently in classification trees of the selected indicators. If m services were considered core services, then there will be 2^m states. When services do not complement each other –as it would be the case of having to choose among different Internet access options – the number of states reduces to just $m+1$, where m is the number of mutually exclusive service options. Thus, either a user is a subscriber to one of the m options or he does not consume the service at all.

The Markovian assumption requires that, conditional on the knowledge of the present state, the probability that the process jumps to any other state in the next time period is independent of the state of the process at all past time periods. Therefore we need to verify the extent to which the Markovian assumption is a good approximation to the actual behaviour of the system. Classification trees can also be used for such purpose.

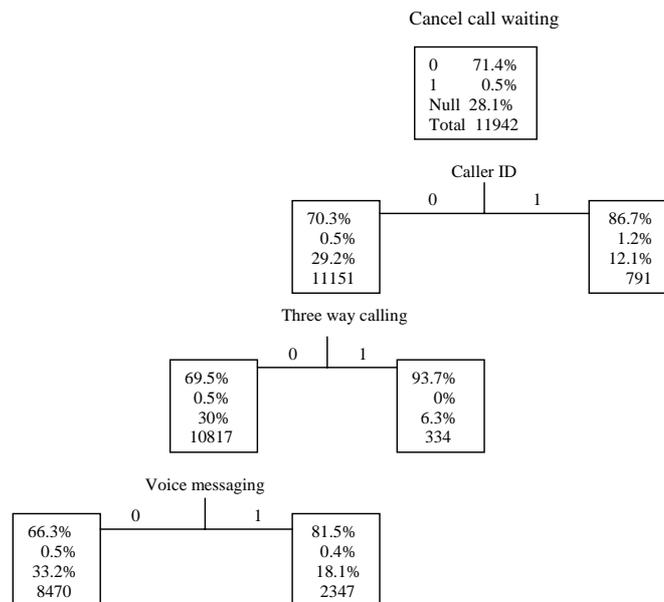


Figure 1. Using a classification tree to determine the likely of service subscription drop out of additional historical information about other services subscription

The root of the tree is a variable called Next State. If we are observing the process at a given time t , Next State will be the state of the process at $t + 1$. A classification tree is used to understand which of the variables at hand, for instance Current State or other observed variables, will explain the transitions occurring in the process. When applying the technique one of two results will be observed: 1) either the tree has one only branch, which means Next State is fully explained by Current State, and no other action is required so the initial Markovian assumption is met; 2) or, some or all of the considered variables are necessary to explain the transition of the process into the next period, which means the set of states currently defined must be refined to involve the information that the tree technique has unveiled.

The latter step may be iteratively applied until no improvement can be achieved.

2.2 Forecasting core services

Forecasting the demand for core services uses matrix A and the vector of initial conditions. $x(0)$ represents how the users are distributed among the states at the time when the most recent period of historical information was recorded. Estimating the evolution of such distribution over time ($t = 1, 2, \dots$) is done by using the equation:

$$E[x(t+1) / x(t)] = A^T x(t)$$

$E[.|.]$ is the conditional expectation, which represents the expectation on the distribution of users at $t+1$ given that the distribution at t is known. Because the forecast is done on the states, and not actually on the amount of connections to a given service, the projected number of connection for a service must be obtained by adding up all through those states whose entry representing such service is 1.

Estimation of peripheral status is described in the work of Cox and Popken (2002).

2.3 Using HMM

In previous sections we have described how to use observable information for demand forecasting. Observable information refers mainly to the actual dates at which a user

subscribed to or was disconnected from a given service. Other historical information about the users available to the provider should also be used. Despite the relative success in building a model around observable information, short-term forecasting can be sensibly improved if we include the treatment of user information that cannot be directly observed. For instance, personal income may be an important factor in the estimation of demand forecasts but many difficulties arise if the provider attempts to obtain such information.

HMM can be used to deal with non observable information. Originally used for signal processing, HMM have become popular in many different settings for estimation of probabilities describing the behaviour of variables that cannot be directly observed.

Formally, we will assume that x_k is a vector of state variables at times $k = 1, 2, \dots$; an observer of the process is able to record the process values. Nevertheless the values of such variables may be polluted due to noisy signals present in the experimentation environment and so the observer actually records a noisy sequence we will call y_k , $k = 1, 2, \dots$

The state equations relating the state variables and the observations made at times $k = 1, 2, \dots$, are:

$$\begin{aligned}x_{k+1} &= A^T x_k + v_{k+1} \\ y_{k+1} &= C^T x_k + w_{k+1}\end{aligned}$$

Equation (1) says that values of state variables at $k+1$ depend on x_k through a parameter matrix and additive noise v_{k+1} ; equation (2) relates the values of the observed variables at $k + 1$ to the state variables through a parameter matrix C ; however such relation is also additively disturbed by a noisy signal w_{k+1} .

In our approach, A is the transition probability matrix between non observable states, that is, an entry a_{ij} in A represents the one-step probability of a transition from state i to state j . Any entry c_{sr} in C represents of a transition from state i to state j . C is the transition probability matrix between non observable states, that is, c_{sr} is the probability that the observed state r has been reached from a transition from non observable state s .

$x(t)$ and $y(t)$ can then be estimated from

$$\begin{aligned}E[x(t+1)|x(t)] &= A^T x(t) \\ E[y(t+1)|x(t)] &= C^T x(t)\end{aligned}$$

Both $A^T x(t)$ y $C^T x(t)$ yield the expected value of $x(t+1)$ and $y(t+1)$ given our knowledge of $x(t)$ at time t .

3 Results

The application was written in Microsoft .NET and uses Statistica® and Microsoft Access. Sales history, from a small telecommunications provider, for 4 Internet access options was used. The options are Dial-up, BroadBand 96K, BroadBand 128K, and BroadBand 256K. Figure 2 shows number of subscribers for each service during past 18 months.

Demand for each service was estimated using the historical data provided. In order to test the accuracy of the method, we used the first 12 months of data. Then forecast estimates were produced for the next 6 months. Estimation was done in two stages: in stage 1, only historical information was used to calculate the forecasted number of users using the Markovian model described above. Then a refining of the data was performed

in which additional information – mainly from a survey done at the time by the provider – was used to verify the extent to which the Markovian property could be realistically assumed.

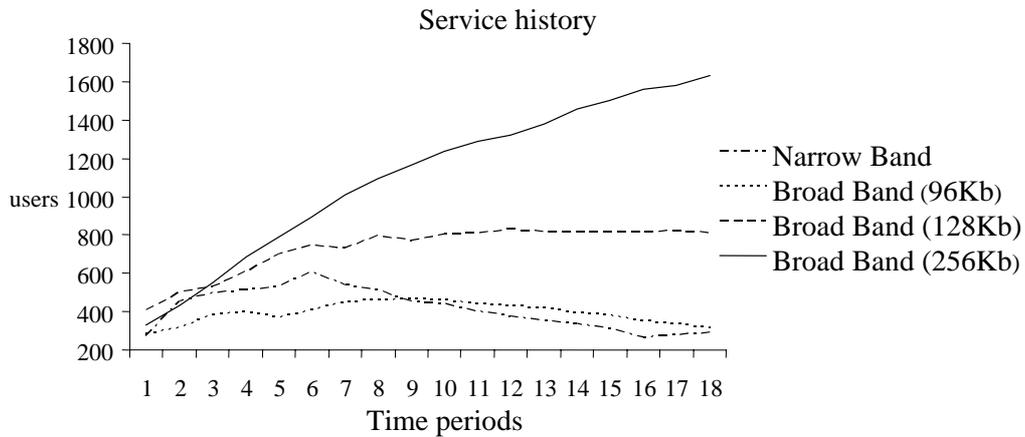


Figure 2. Number of subscribers to each Internet access option over an 18-month period

Figure 3 shows the estimated forecasts. For each service the number of users over 18 months is depicted. Period 12 is forecast period 0, which means that the first forecasted value is found at F1. The example illustrates a frequently found situation whereby the forecasts with and without refinement do not exhibit significantly different values.

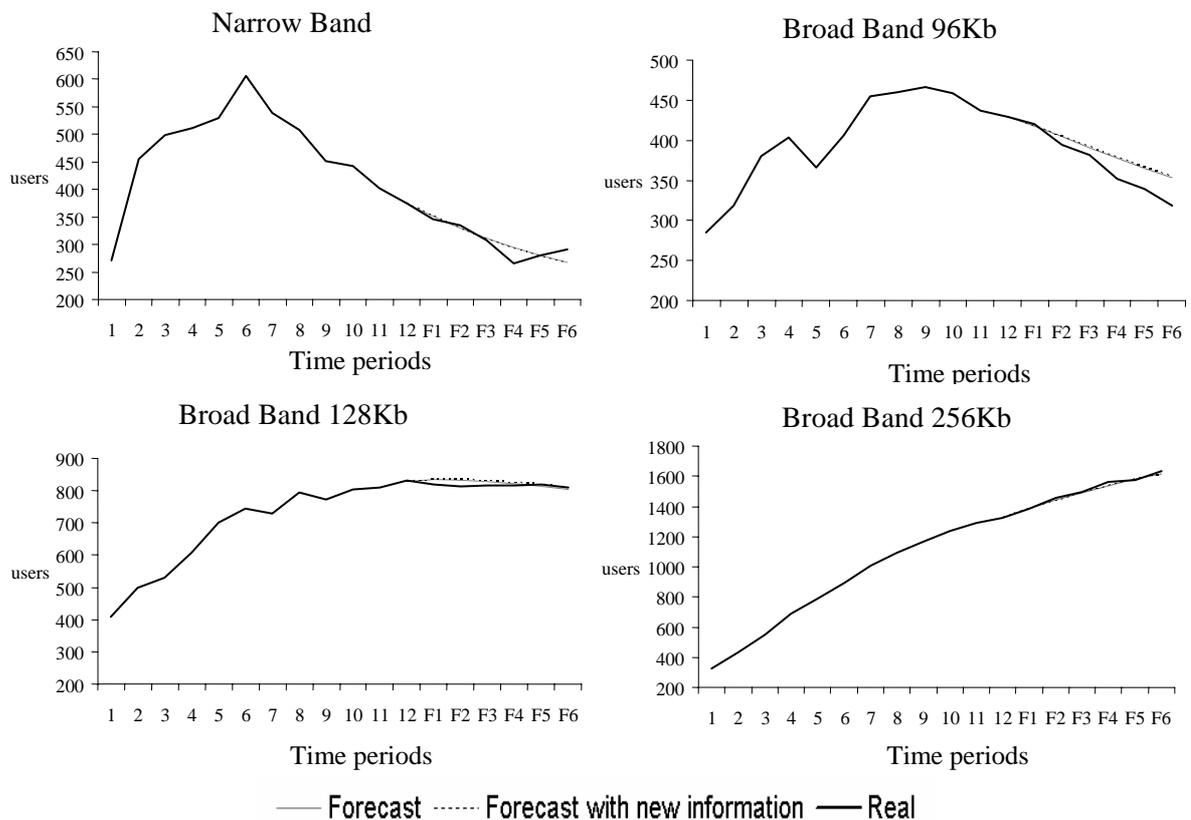


Figure 3. Forecasts of the number of users in each service option

Even though differences in values yielded by the two methods are not notoriously significant, forecasts produced with refined states – which implies a larger size Markov model – fit themselves slightly better with the real values.

Good fitness of the forecasts is observed in most cases. However there may be situations where neither type of forecast (with or without refined states) is able to make a good approximation to the real historical data. Figure 4 is an example of such case. In Figure 4 we can see a strong shift in the demand level for BroadBand 512K about period F2, which may have been brought about by the introduction of a special offer or a price discount. Without such information, as was the case here, our model is definitely not able to track such sudden changes.

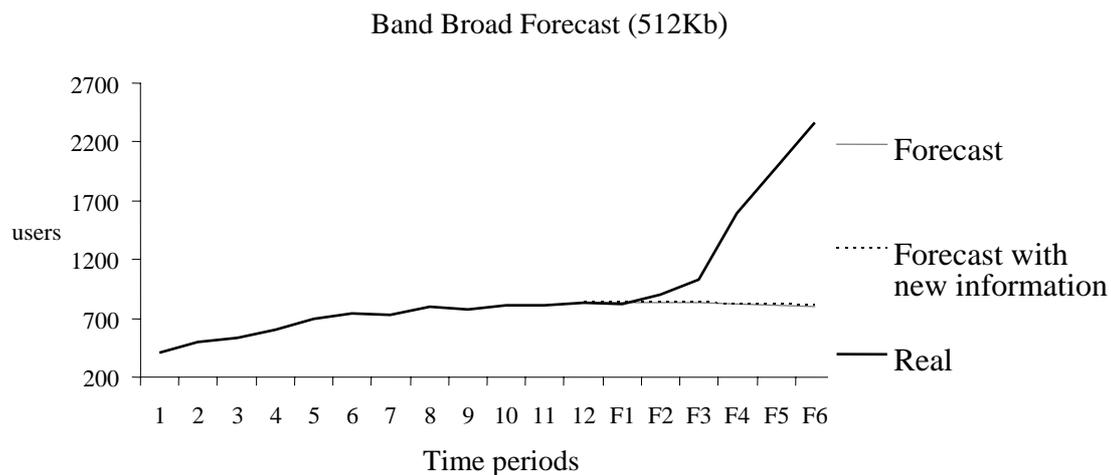


Figure 4. An illustration of the inability of the hybrid approach to follow sudden changes in historical demand

Non observable variables – such as personal income – are introduced into the model in order to assess their impact on the accuracy of estimations. When personal income was considered and segmented into several levels, the model provides a higher degree of discrimination between consumers and forecasts for specific groups are obtained. Figure 5 shows results of the application of a HMM to forecast the number of subscribers to each of the four Internet access options. In Figure 5 the segmented forecasts have been aggregated to portrait the aggregate forecast for each service. One striking feature of our procedure is the dramatic decrease in the number of consumers at period F1 – the first forecasted period. This feature has appeared consistently throughout the experiments we carried out.

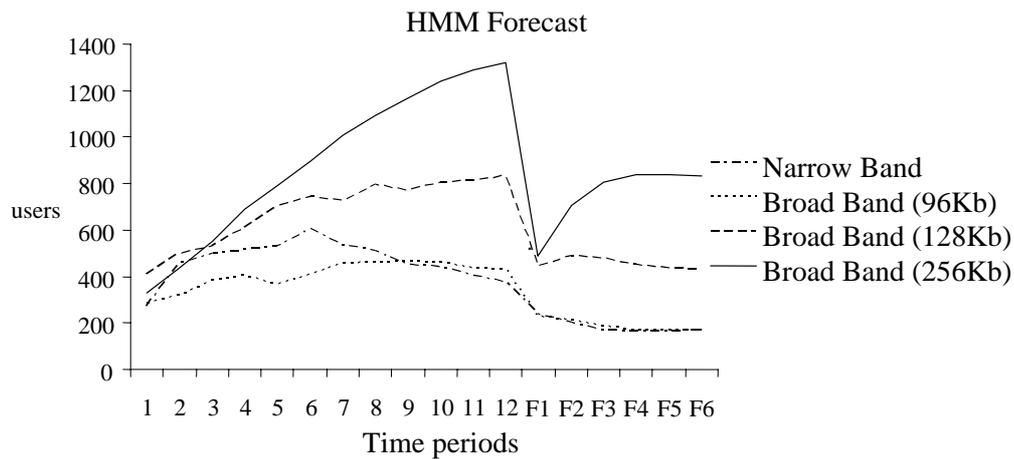


Figure 5. Using HMM to forecast subscriber levels for 4 Internet access options

A final illustration of results obtained can be seen in Figure 6. Herein forecasts are for services which may be jointly acquired – or combinations of some of them – by consumers. The purchaser is required to already have a telephone connection; services are then activated, or deactivated as desired. The algorithm described in Section 2 is used to define and refine states, and define the core services.

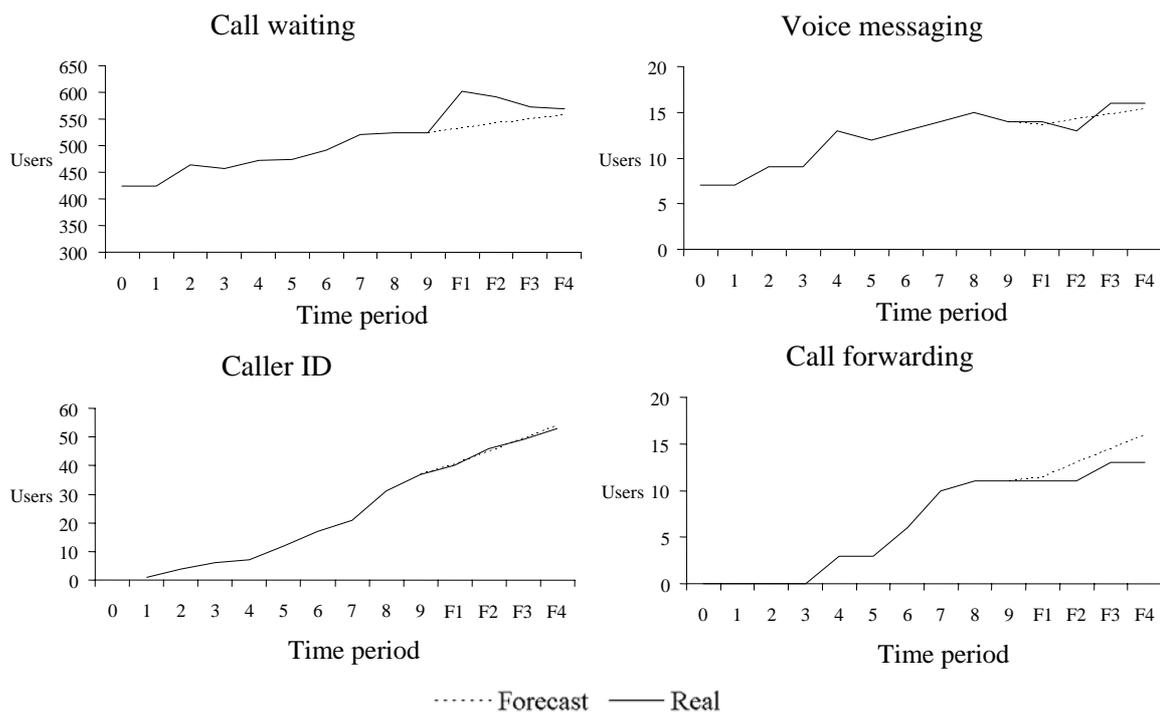


Figure 6. Forecasts for services that may be jointly purchased (complementary service)

4 Conclusions

Our experimentation was based on the use of partial historical information on service subscription. Forecasts were estimated choosing the ending time period of the partial history as period zero for forecasts.

Results turned out to be appealing under certain specific characteristics of the data. One of those is the absence of extreme changes in demand. As observed, extreme

demand increases or decreases cannot be predicted by the hybrid model herein used, regardless of whether the model includes non observable variables or not.

When states were refined, results did not change considerably. Nevertheless, refining states provides the researcher with a higher degree of assurance about the use of Markovian models. Refining can also help in segmenting users according to their profiles when the simple use of connections and disconnections is not enough to segment users.

Using the hybrid we have estimated demand forecasts that do not follow a strictly increasing or decreasing trend unlike other methods. The coupling of classification methods and Markovian models provides a non-smoothing procedure that seems to capture some behaviour patterns that smoothing methods tend to disregard. Aggregating individual consumption patterns is key to the approach presented in this paper and can be realistically done when a service provider keeps individual records of its subscribers. Such is the case of most telecommunications providers.

5 Future Work

The use of a hybrid approach followed in this paper is a novel area of research. We have highlighted some of the difficulties faced when introducing HMM aiming at improving the accuracy of predictions. In particular, we could not identify the cause for the unexpected decreases in the forecasted number of users.

One of the weaknesses revealed by our use of HMM is the estimation of the initial distribution of customers across observable and non observable variables. Forecasts realized with the inclusion and estimation of on observable variables (HMM) commonly present an initial, dramatic reduction in the forecasted numbers; we believe this is due to a poor quality of information regarding the distribution of users across the segments or groups defined by the non observable variables. Our approach consisted of the use of surveyed information obtained by the provider from a sample of its customers; a more reliable procedure is needed to more accurately estimate initial distribution. However, surveys are expensive procedures that search for information the provider cannot access from its history records.

6 References

- Cox, L. and Popken, D. (2002). A hybrid system-identification method for forecasting telecommunications product demands. *International Journal of Forecasting*, 18, 647-671.
- Fildes, R. (2002). Telecommunications demand forecasting-a review. *International Journal of Forecasting*, 18, 489-522
- Islam, T., Fiebig, D. G., and Meade, N. (2002). Modelling multinational telecommunications demand with limited data. *International Journal of Forecasting*, 18, 605-624.
- Mouchart, M. and Rombouts, J. (2005) Clustered panel data models: An efficient approach for nowcasting from poor data. *International Journal of Forecasting*, 21, 577-594
- Taylor, L.D. (1994) *Telecommunications Demand in Theory and Practice*. Dordrecht: Kluwer Academic.

If I'm doing it, it must be O.R. - An example of the use of Operational Research in evaluating the effectiveness of Special Education provision.

Nicola Ward Petty
Department of Management
University of Canterbury
New Zealand
nicola.petty@canterbury.ac.nz

Abstract

The analysis of resource allocation for learners with vision impairment has taken an interesting path. An early qualitative study helped to identify potential outcome measures, based on the agreed purpose of the itinerant service. An instrument was then developed to measure opportunity-to-learn, that drew on the ideas of the intended and implemented curriculum. Statistical modelling, including multi-level modelling was used to explore the provision and effectiveness of services to high-school students with vision impairment. Unexpected quantitative results were interpreted in the light of the qualitative data.

This presentation summarises the process and suggests how the Operations Research process has illuminated the analysis. The following provides an overview of the analysis.

Key words: Operational Research, Special Education.

1 Defining Operational Research

The definition of Operations Research given on the Min and Max website states: "O.R. seeks to improve a problem situation by supplying decision makers with information and insights gained through problem analysis, often involving mathematical models and computers." The "Science of Better" website provides a shorter definition: "Operations research (O.R.) is the discipline of applying advanced analytical methods to help make better decisions." These definitions have three elements in common: improvement, decision-making and analysis. The first definition also includes the element of modelling.

2 The problem

Existing case-load guidelines for the education of learners with vision impairment within New Zealand, and internationally, are not based on empirical research. The motivation for this research was to develop a way to clarify the relationship between the

inputs and outcomes of educational endeavours for learners with vision impairment, with the aim of informing caseload policy.

The increasing emphasis on providing educational opportunity for all students, regardless of need, has resulted in increasing Special Education provision. Depending on viewpoint, this can be interpreted as a basic human right, a necessary cost to society, an investment in the future or an almost endless drain on resources.

This research was begun with the intention of providing information regarding ideal caseloads, and developing research-based caseload formulas. A subsequent study of the nature of school effectiveness research and attempts to develop education production functions for regular school children indicated that even for the general population, issues such as class-size or school-day length are difficult to establish and the source of ongoing research and debate. In the face of this difficulty in quantifying much of the education process for the regular school population, it became clear that a more realistic aim for this study was to provide information that would inform caseload decision-making.

3 The process

Before embarking on quantitative data collection, a preliminary qualitative study was undertaken in order to inform the research process. The main outcomes of the preliminary study were the identification of the purposes of the service predominantly provided by RTVs (Resource Teachers : Vision), a potential outcome measure, and many potential indicators of need.

The preliminary study gave rise to the idea of measuring opportunity-to-learn and this was explored in the research literature. It was proposed that the students could be asked about their perceptions, and an instrument, the Essential Skills Access test (ESA) was developed, based on the Essential Skills of the New Zealand Curriculum, to measure the access that students have to the curriculum, and in particular the development of skills. The ESA test was piloted and then trialled on a baseline sample of 1300 students from twenty diverse schools in three regions of New Zealand. The analysis of this data suggested that student perceptions could be used to indicate differences between schools, and between boys and girls, with respect to opportunity-to-learn skills.

The focus of the research then returned to the original population of interest, the learners with vision impairment, and in particular those in the first three years of secondary school. Fifty students with vision impairment were surveyed using the instrument, which had been adapted into formats accessible for this population. In addition, data was collected from the school, mainstream teachers, RTV and parents of each of the students regarding service provision and the level of need.

4 Outcomes and contributions

This research has covered several fields of study and there are several areas of contribution. These include

- Measurement of educational opportunity as perceived by students - a process indicator. (To be published in "School Effectiveness and School Improvement")
- Results from the baseline study indicating different opportunity-to-learn scores for boys and girls. (Presented to New Zealand Association of Research in Education.)
- Clarification of the purpose of provision to learners with vision impairment.

- An examination of the efficacy of measuring opportunity-to-learn to evaluate the education of learners with vision impairment.
- An analysis of the strengths and weaknesses of vision education in New Zealand for years 9 to 11. (This was provided to the key New Zealand service provider.)

5 Is it Operational Research?

This research crosses boundaries and touches on several disciplines. It has been reported in Education and Educational Effectiveness conferences and has a place in the literature of Vision Impairment and Blindness. The methods used are also used in an Educational Evaluation paradigm, which frequently entertains mixed-methods approaches. However, this does not preclude it from also having a place in Operational Research literature.

The four elements of Operational Research identified in the beginning of the article were improvement, decision-making, analysis and modelling. The research is firmly seated in a philosophy of attempting to improve a situation, in particular the provision of resources for the education of students with vision impairment. The outcomes and results can be used to inform decision-making in this area. Extensive statistical analysis formed the heart of two of the three sections of work. The paradigm of modelling pervaded the whole analysis. The qualitative and quantitative phases of the work all worked together to build a model of the system in question.

6 Website addresses

www.minandmax.org.nz (November 2006)

<http://www.scienceofbetter.org/what/index.htm> (November 2006)

Finding Representative Nondominated Points in Multiobjective Radiotherapy Planning

Lizhen Shao

Department of Engineering Science

University of Auckland

New Zealand

l.shao@auckland.ac.nz

Abstract

We formulate the beam intensity optimization problem of radiotherapy planning as a multiple objective linear programme (MOLP) and we address the problem of finding a set of representative points on the nondominated surface. We use an algorithm which is based on normal boundary intersection to produce the nondominated points. Discrepancy analysis of the nondominated points shows that they are evenly distributed. Therefore this method is acceptable for our MOLP model. Finally we apply this algorithm to the beam intensity optimization problem and we show the results for some clinical cases.

Keywords: MOLP, radiotherapy, nondominated points.

1 Introduction

The aim of radiation therapy is to kill tumor cells while at the same time protecting the surrounding tissue and organs from the damaging effect of radiation. Currently these goals are achieved by using computerized inverse planning systems. Given the number of beams and beam directions, computerized inverse planning systems need to produce beam intensity profiles that yield the best dose distribution under consideration of clinical and physical constraints. This is called beam intensity optimization problem.

In the past, the beam intensity optimization problem has been formulated as a linear or nonlinear programming model, see (Shao 2005) for a survey of these models. In these models, the conflicting objectives – effective treatment of the tumor and limiting the radiation dose to the surrounding tissue and organs at risk are summed up into one using a weight or “importance factor” for the tumor, each organ at risk and the normal tissue. However, selecting weights is decision making before the optimization, which is problematic, and leads to a trial-and-error process of getting the “correct” weights.

Recently, multiobjective optimization has been introduced into radiation therapy planning problem. For example, Hamacher and Küfer (2002) and Küfer et al. (2003) formulate the beam intensity optimization problem as a multiobjective linear programming model, and Cotrutz et al. (2001) and Lahanas, Schreibmann, and Baltas (2003) formulate it as a multiobjective nonlinear programming model. The purpose of multiobjective optimization is to obtain a representative subset of the nondominated set. The set of all the nondominated points forms the nondominated surface in the objective space. For a multiple objective problem, a nondominated point in the objective space corresponds to an efficient solution which is defined as a solution in which an improvement in one objective will always lead to a worse result in at least one of the other objectives. Given a representative set on the nondominated surface, the treatment planner can interactively navigate through these plans. Therefore, the trade offs between different objectives, such as overdosing the organs at risk and underdosing the tumor can be explored. This helps the planner to choose the most preferred plan from the representative set of nondominated solutions.

Sayin (2000) defines coverage, uniformity and cardinality as the three attributes of quality of discrete representations of the nondominated set for a multiple objective programme (MOP). According to these three attributes, a good representation needs to contain a reasonable number of points, should not miss large portions of the nondominated set, and should not contain points that are very close to each other.

At present, the most popular way to obtain a representative set of the nondominated set for radiation therapy planning problem is the weighted sum method, see (Cotrutz et al. 2001; Lahanas, Schreibmann, and Baltas 2003). For a given weight combination we can obtain a single point on the nondominated surface. Thus, in order to produce a set of nondominated points, we need to choose a set of weights. However, it is difficult to choose the weights to make the nondominated points evenly distributed. Even if we use an evenly distributed set of weights it is possible that the points which we obtain on the nondominated surface are not uniformly distributed (Das and Dennis 1997). Therefore, the quality of the representative set produced by the weighted sum method can not be guaranteed.

The purpose of this paper is to describe a method to find a good representative set of the nondominated surface for the radiation therapy planning problem.

In this paper, we formulate the radiation therapy planning problem as a multiple objective linear programme and use a method which is based on the normal boundary intersection (NBI) method (Das and Dennis 1998) to produce a representative set of the nondominated points. These representative nondominated points are shown to be evenly distributed according to the discrepancy analysis. The work is organized as follows. Section 2 describes the mathematical model we use. Section 3 presents the NBI method, the revised NBI method for our MOLP model and the discrepancy analysis of the nondominated points. Three simplified clinical cases are solved and some results are given in Section 4.

2 Mathematical Model

In order to calculate dose in the beam intensity optimization problem, the patient's 3D volume is divided into m small voxels, and the beam is discretized into n small bixels. Then we have

$$d = Ax, \tag{1}$$

where $d \in \mathbb{R}^m$ is a dose vector and its element d_i corresponds to the dose in voxel i . $x \in \mathbb{R}^n$ is a beam intensity vector, x_j represents the intensity of bixel j . $A \in \mathbb{R}^{m \times n}$ is a dose deposition matrix, we assume that it is given. The elements of A , a_{ij} , represent the dose deposited in voxel i due to unit intensity in bixel j . A can be partitioned and reordered into sub-matrices $A_T \in \mathbb{R}^{m_T \times n}$, $A_C \in \mathbb{R}^{m_C \times n}$ and $A_G \in \mathbb{R}^{m_G \times n}$ ($m_T + m_C + m_G = m$) according to the rows corresponding to tumor, critical organs and normal tissue voxels, respectively.

For treatment planning, the planner needs to specify a ‘‘prescription dose’’ for the tumor, each organ at risk and normal tissue. Then we use the ‘‘prescription dose’’ to construct $TLB \in \mathbb{R}^{m_T}$, $TUB \in \mathbb{R}^{m_T}$, $CUB \in \mathbb{R}^{m_C}$ and $NUB \in \mathbb{R}^{m_G}$ corresponding to tumor lower bounds, tumor upper bounds, critical organ upper bounds and normal tissue upper bounds.

Based on Holder’s linear programming formulation (Holder 2003), we formulate the beam intensity optimization problem as a multiple objective linear programme (MOLP).

In this model, we minimize the maximum deviation from tumor lower bounds (α), critical organ upper bounds (β) and normal tissue upper bounds (γ) at the same time. The model can be described as follows:

$$\begin{aligned}
\min \quad & \{\alpha, \beta, \gamma\} \\
\text{s.t.} \quad & TLB - \alpha e \leq A_T x \leq TUB \\
& A_C x \leq CUB + \beta e \\
& A_N x \leq NUB + \gamma e \\
& 0 \leq \alpha \leq \alpha UB \\
& -\min CUB \leq \beta \leq \beta UB \\
& 0 \leq \gamma \leq \gamma UB \\
& 0 \leq x,
\end{aligned} \tag{2}$$

where $\alpha UB \in \mathbb{R}$, $\beta UB \in \mathbb{R}$ and $\gamma UB \in \mathbb{R}$ are the upper bounds for α , β , and γ , respectively. They are specified by the planner and restrict the search to clinically relevant values.

We can see that the three objectives α , β and γ in (2) are bounded by their upper and lower bounds. Moreover, we need to point out that due to the nonnegativity of beam intensity, this MOLP problem is always feasible as long as appropriate lower bounds and upper bounds for tumor, critical organ and normal tissue were set.

3 Computation Method

3.1 Preliminaries

Consider a multiple objective linear programming problem,

$$\min \{Cx : x \in X\}, \tag{3}$$

where $C \in \mathbb{R}^{p \times n}$ is the $p \times n$ matrix whose rows c_k , $k = 1, 2, \dots, p$, are the coefficients of p linear functions $\langle c_k, x \rangle$, $k = 1, 2, \dots, p$ and $X \subseteq \mathbb{R}^n$ is a nonempty polyhedral set. The objective set for this problem Y is defined by

$$Y = \{Cx : x \in X\}. \tag{4}$$

Rockafellar (1970) has shown that such an image Y of a convex polyhedron X by a linear map C is also a convex polyhedron.

Definition 3.1. x^0 is an efficient solution for problem (3), if $x^0 \in X$ and there exists no $x \in X$ such that $Cx \leq Cx^0$ and $Cx \neq Cx^0$. The set of all efficient solutions of problem (3) will be denoted by X_E , it is called the efficient set in decision space. Correspondingly, $y^0 = Cx^0$ is called a nondominated point and $Y_N = \{Cx : x \in X_E\}$ is the nondominated set in objective space for problem (3).

Definition 3.2. A feasible solution $\hat{x} \in X$ is called weakly efficient if there is no $x \in X$ such that $Cx < C\hat{x}$, i.e. $c_i x < c_i \hat{x}$, for all $i = 1, 2, \dots, p$. The point $\hat{y} = C\hat{x}$ is then called weakly nondominated.

Theorem 3.3. A feasible solution $x^0 \in X$ is an efficient solution of the MOLP (3) if and only if there exists a $\lambda \in \mathbb{R}_{>}^p$ such that

$$\lambda^T Cx^0 \leq \lambda^T Cx \quad (5)$$

for all $x \in X$.

The reader is referred to (Ehrgott 2005) for a proof.

Definition 3.4. Let $F \subset Y$ be a face of Y . F is called nondominated face, if $F \subset Y_N$.

3.2 Normal Boundary Intersection

The normal boundary intersection method was developed by Das and Dennis (1998) for finding uniformly spread nondominated points for a general nonlinear multi-objective optimization problem. The method is independent of the relative scales of the functions and is successful in producing an evenly distributed set of nondominated points on the nondominated surface.

Since our model of beam intensity optimization problem is an MOLP model, we will use an MOLP to illustrate how NBI works.

Consider the MOLP problem (3). Let $y^I = (c_1 x^1, c_2 x^2, \dots, c_p x^p)^T$ be the ideal point and the individual minimum of the functions y^1, y^2, \dots, y^p be attained at x^k for each $k = 1, 2, \dots, p$, i.e., $y^k = Cx^k$. Then the convex hull of the individual minima (CHIM) is obtained. A set of equidistant reference points on the convex hull of the individual minima (CHIM) is generated and, for each of them, a NBI subproblem is solved for finding the farthest point on the boundary of Y along the normal \hat{n} of the CHIM pointing toward the origin. The NBI subproblem for a given reference point q is as follows:

$$\begin{aligned} \max \quad & t \\ \text{s.t.} \quad & q + t\hat{n} \in Y \\ & t \geq 0 \end{aligned} \quad (6)$$

Figure 1 shows how the NBI method works for an MOLP example with two objectives. The limitation of the NBI method is that the solution method may overlook a portion of the nondominated surface for $p > 2$. For details, the reader is referred to (Das and Dennis 1998). Moreover, Das and Dennis (1998) do not provide bounds on the spacing of the resulting points.

3.3 Revised Normal Boundary Intersection for our MOLP Model

For our MOLP model of the beam intensity optimization problem, we know that the feasible region Y in the objective space is bounded due to the lower and upper bound for our three objectives, α , β and γ . Based on this, we revise the NBI method. In the revised NBI method, instead of CHIM, we use a supporting hyperplane of Y_N as the reference plane to put the equidistant points. By doing this, we overcome the limitations of the NBI method.

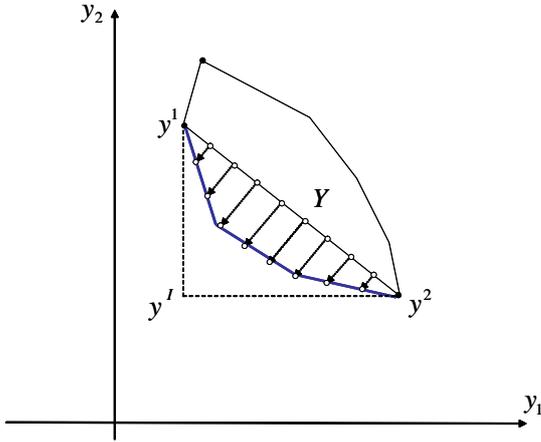


Figure 1: Solutions obtained in the NBI method.

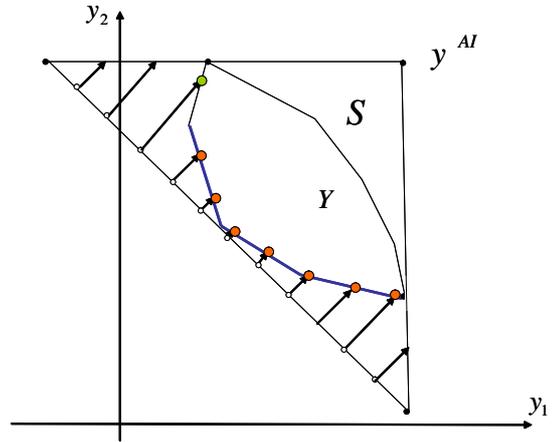


Figure 2: Solutions obtained in the revised NBI method.

The revised normal boundary intersection involves choosing a reference plane, putting the equidistant reference points on the plane and computing the intersection point of the normal and the boundary of Y . At last, we need to check if the intersection point is nondominated or not because not every intersection point is nondominated. In the following paragraphs, we show the details of the revised NBI method.

Reference Plane. Here we use the subsimplex of the simplex which was used in (Benson and Sayin 1997) as the reference plane.

Define

$$y_k = \max\{y_k : y \in Y\}. \tag{7}$$

$y^{AI} \in \mathbb{R}^p$ is called the anti-ideal point for problem (3).

Let

$$\beta = \min\{\langle e, y \rangle : y \in Y\}, \tag{8}$$

where $e \in \mathbb{R}^p$ is the vector in which each entry is 1.

Define $p + 1$ points $v^l \in \mathbb{R}^p$, $l = 0, 1, \dots, p$. Let $v^0 = y^{AI}$ and, for $l = 1, 2, \dots, p$,

$$v_k^l = \begin{cases} y_k, & \text{if } k \neq l, \\ \beta + y_l - \langle e, v^0 \rangle, & \text{if } k = l, \end{cases} \tag{9}$$

$k = 1, 2, \dots, p$. Then the convex hull S of $\{v^l : l = 0, 1, \dots, p\}$ is a p -dimensional simplex, and S contains Y , as shown by Benson and Sayin (1997).

For our MOLP model, S is a three dimensional simplex. The subsimplex of S given by the convex hull R of $\{v^j : j = 1, 2, \dots, p\}$ is the reference plane. It is a supporting hyperplane of Y_N .

Equidistant Points on the Reference Plane. We use R as the region to put equidistant points. For $p = 2$, R is a line segment, while for $p > 2$, R is a $p - 1$ dimensional simplex with equal edge length and with the normal direction e according to the construction of S .

For our MOLP model, R is an equilateral triangle in the three dimensional objective space. Therefore, we use a triangular lattice to produce the equidistant points, see Figure 3.

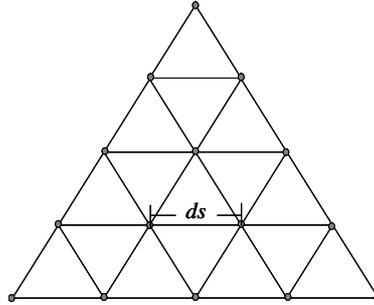


Figure 3: Equidistant reference points on the reference plane.

Computing the Intersection Points and Checking Nondominance. Given a reference point q on R , the revised NBI subproblem searches for the closet point to the reference point on the boundary of Y along the direction of the normal direction e . The revised NBI subproblem is as follows:

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & q + te \in Y \\ & t \geq 0. \end{aligned} \tag{10}$$

There are three scenarios for the solutions of (10), as we can see from Figure 2.

1. There is no intersection between the normal and the boundary of Y .
2. The normal and the boundary of Y intersect, but the intersection point is dominated.
3. The intersection point is nondominated.

If LP (10) is infeasible, then there is no intersection between the normal and the boundary of Y , else there is an intersection point. We know that not every intersection point is a nondominated point. Therefore, we need to check if it is dominated or not.

Theorem 3.5. *Assume that $\lambda \in \mathbb{R}_{>}^p$ and $\bar{y} \in Y$. Then \bar{y} belongs to Y_N if and only if \bar{y} is an optimal solution to the the following problem*

$$\begin{aligned} \min \quad & \langle \lambda, y \rangle \\ \text{s.t.} \quad & y \leq \bar{y}, \quad y \in Y. \end{aligned} \tag{11}$$

The reader is referred to (Ehrgott 2005) for a proof. According to Theorem 3.5, we can check if a point is dominated or not by solving (11).

3.4 Discrepancy Analysis of the Nondominated Points

The nondominated surface for an MOLP is the union of the nondominated faces. These nondominated faces are polygons due to Y being a convex three dimensional polytope.

Given a nondominated face, the angle between the reference plane and the plane of the nondominated face can be calculated as follows:

$$\cos \theta = \frac{\langle n^1, n^2 \rangle}{\|n^1\| \|n^2\|}. \tag{12}$$

$n^1 \in \mathbb{R}^p, n^2 \in \mathbb{R}^p$ are the normal vector of the reference plane and the plane of the nondominated face, respectively.

Because the normal vector of the reference plane n^1 is equal to $e \in \mathbb{R}^p$, therefore, formula (12) can be written as:

$$\cos \theta = \frac{n_1^2 + \dots + n_p^2}{\sqrt{(n_1^2)^2 + \dots + (n_p^2)^2} \sqrt{p}}. \tag{13}$$

According to Theorem 3.3 and Definition 3.4, a set $F \in \mathbb{R}^p$ is a face of Y_N of the MOLP (3) if and only if F equals to the optimal solution set $Y^*(\lambda)$ to the problem

$$\min\{\langle \lambda, y \rangle : y \in Y\} \tag{14}$$

for some $\lambda \in \mathbb{R}_{>}^p$. Therefore, we know $n^2 \in \mathbb{R}_{>}^p$ and we have

$$\frac{n_1^2 + \dots + n_p^2}{\sqrt{(n_1^2)^2 + \dots + (n_p^2)^2} \sqrt{p}} > \frac{n_1^2 + \dots + n_p^2}{\sqrt{(n_1^2 + \dots + n_p^2)^2} \sqrt{p}} = \frac{1}{\sqrt{p}}. \tag{15}$$

When $n^1 = kn^2, k \neq 0$, we have $\cos \theta = 1$. So the range of $\cos \theta$ is

$$\frac{1}{\sqrt{p}} < \cos \theta \leq 1 \tag{16}$$

and θ is in the range of $0 \leq \theta < \arccos \frac{1}{\sqrt{p}}$.

If $p = 2, 0 \leq \theta < \frac{\pi}{4}$. If $p = 3, 0 \leq \theta < \arccos \frac{\sqrt{3}}{3}$. We can see that when p increases, the range of angles between the reference plane and a plane of the nondominated face will increase.

Suppose we use a triangular lattice with edge length ds (see Figure 3) to produce the reference points, then the distance between the nondominated points can be calculated as $(ds/\cos \theta)$.

Figure 4 shows an example with two objectives ($p = 2$). The nondominated faces are line segments. F_1 is a nondominated face, while F_2 is a weakly nondominated face. The biggest angle between the nondominated face and the reference plane is approaching $\frac{\pi}{4}$. $\theta = \frac{\pi}{4}$ is the angle between the reference plane and the weakly nondominated face. The distance between the nondominated points obtained by the revised NBI method is between ds and $\sqrt{2}ds$.

For $p = 3, ds \leq d < \sqrt{3}ds$. As p increases, the range of the distance d between the nondominated points on the nondominated surface increases. For our MOLP model $p = 3$, we think the range is acceptable. This result quantifies the quality of representation in terms of coverage and uniformity. Cardinality can be controlled by the number of reference points.

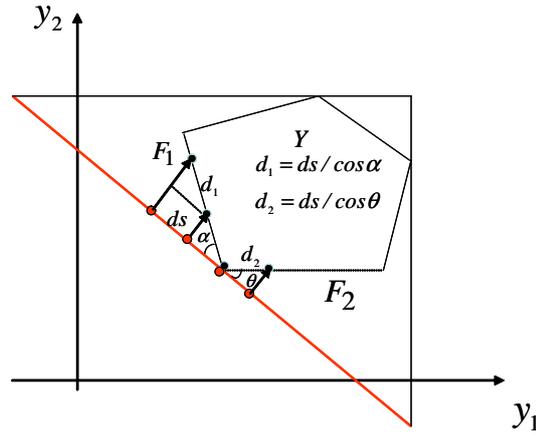


Figure 4: Discrepancy analysis for a 2D example.

Case	tumor	critical organ	normal tissue	bixels
Acoustic	9	47	999	594
Prostate	22	89	1182	821
Pancreatic lesion	67	91	986	1140

Table 1: The number of voxels (total = m) and bixels (n).

4 Results

Three clinical cases are used: an acoustic neuroma (acoustic), a prostate and a pancreatic lesion, see Figure 5. These simplified cases have a voxel size of $5mm$ on a single CT slice. For all examples, a total of 72 evenly distributed beams were used at angles $5^\circ n$, where $n = 0, \dots, 71$. The number of voxels and bixels used for optimization of each case is shown in Table 1 and some prescription information is shown in Table 2.

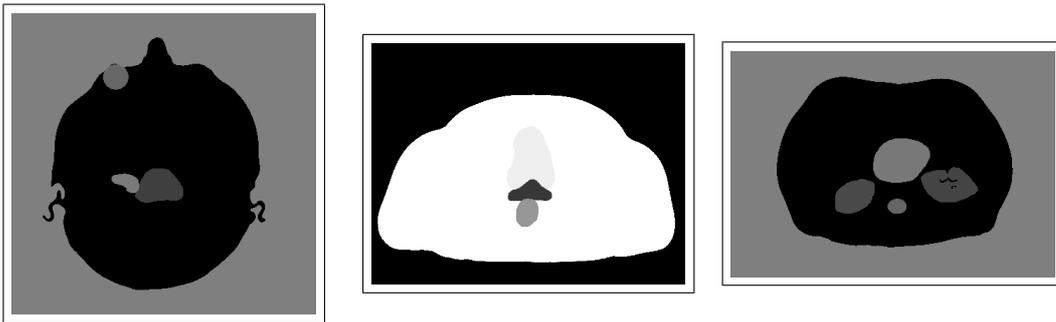


Figure 5: Pictures from left to right are acoustic, prostate and pancreatic lesion.

In each case, as we mentioned in our MOLP model, we consider three-dimensional trade offs, the maximum deviation from tumor lower bounds, the maximum deviation from critical organ upper bounds and the maximum deviation from the normal tissue upper bounds.

In Table 3, we list the number of reference points, the number of intersection points between the normal and the boundary of Y , the number of nondominated points and the computation time for producing the nondominated points for each

Case	TUB	TLB	CUB	RUB	αUB	βUB	γUB
Acoustic	87.55	82.45	60, 45	0	16.49	12	87.55
Prostate	90.64	85.36	60, 45	0	17.07	12	90.64
Pancreatic lesion	90.64	85.36	60, 45	0	17.07	12	90.64

Table 2: Lower and upper bounds for tumor, critical organs and normal tissue (in Gy).

	reference points	points on Y	nondominated points	ds	time (seconds)
Acoustic	378	72	72	1.04	56.3
Prostate	378	144	112	4.79	101.3
Pancreatic lesion	378	145	129	3.31	523.9
Acoustic	153	29	29	1.59	27.5
Prostate	153	62	48	7.30	44.5
Pancreatic lesion	153	59	54	5.06	213.9

Table 3: The number of reference points, points on Y , nondominated points and the distance between the reference points ds .

case. For all three cases, more than half of the reference points will not produce intersection points. We know that no intersection means that LP (10) is infeasible and it does not take long to check the infeasibility as it is very simple, so the reference points with no intersection points being produced will not cause a big problem for the computation time. Moreover, we can see from the prostate and pancreatic lesion cases in Table 3, that not every intersection point corresponds to a nondominated point. Therefore, it is necessary to check nondominance even though it takes time.

The computation time is related to the number of reference points which corresponds to the number of LPs to be solved. Therefore, for the same case, more reference points need more computation time as we can see in Table 3.

We show the nondominated points of the three clinical cases in Figures 6 and 7. We can see from these pictures that the nondominated points are evenly distributed. The revised NBI method overcomes the deficiency of the NBI method, i.e., the calculated nondominated points cover the whole nondominated surface. As long as we have enough equidistant points on the reference plane, the nondominated points produced will be a good representation of the nondominated surface according to coverage, uniformity and cardinality, the three attributes of our discrete representation.

5 Conclusion

In this paper, we formulate the beam intensity optimization problem as an MOLP and apply a revised normal boundary intersection method to produce a representative subset of the nondominated set. This representative set can help the decision maker to choose a most preferred plan, thus avoiding trial and error process.

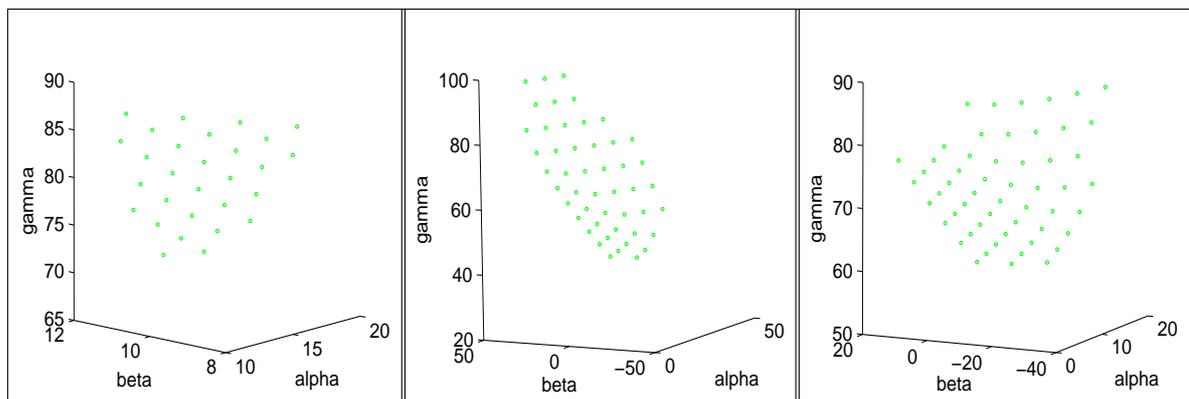


Figure 6: Pictures from left to right are the nondominated points of acoustic, prostate and pancreatic lesion with 153 reference points.

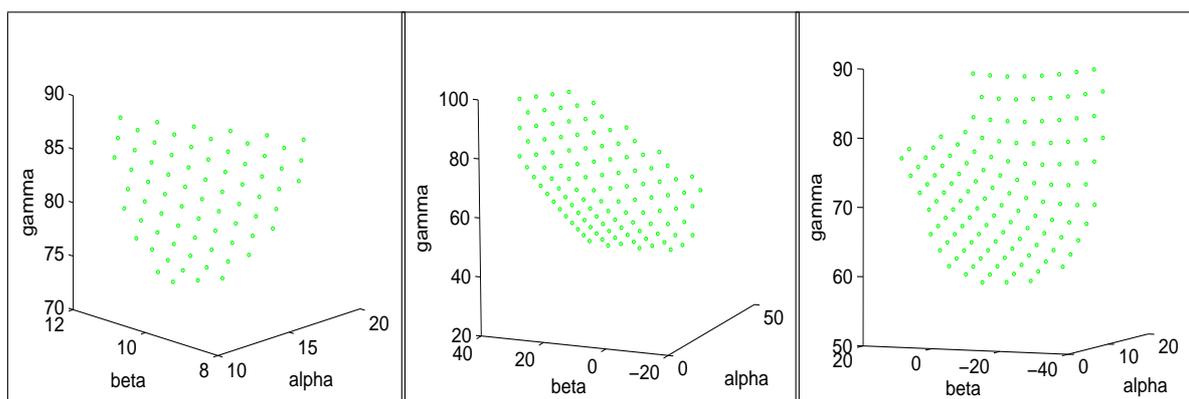


Figure 7: Pictures from left to right are the nondominated points of acoustic, prostate and pancreatic lesion with 378 reference points.

Discrepancy analysis of the nondominated points produced by the revised NBI method proves that the nondominated points are evenly distributed for our MOLP problem. Moreover, the clinical results also show us that the nondominated points are evenly distributed.

In this paper, we only show the results for 2D clinical cases, but this revised NBI is able to solve 3D cases. The only difference is the computation time because the size of the LPs increases.

References

- Benson, H. P., and S. Sayin. 1997. "Towards Finding Global Representations of the Efficient Set in Multiple Objective Mathematical Programming." *Naval Research Logistics* 44:47–67.
- Cotrutz, C., M. Lahanas, C. Kappas, and D. Baltas. 2001. "A multiobjective gradient-based dose optimization algorithm for external beam conformal radiotherapy." *Physics in Medicine and Biology* 46 (8): 2161–2175.
- Das, I., and J. E. Dennis. 1998. "Normal-boundary intersection: A new Method for generating the pareto surface in nonlinear multicriteria optimization problems." *SIAM Journal on Optimization* 8 (3): 631–657.
- Das, I., and J.E. Dennis. 1997. "A closer look at drawbacks of minimizing weighted

- sums of objectives for Pareto set generation in multicriteria optimization problems.” *Structural and Multidisciplinary Optimization* 14:63–69.
- Ehrgott, M. 2005. *Multicriteria Optimization*. Springer.
- Hamacher, H.W., and K.-H. Küfer. 2002. “Inverse radiation therapy planing – A multiple objective optimization approach.” *Discrete Applied Mathematics* 118 (1-2): 145–161.
- Holder, A. 2003. “Designing Radiotherapy Plans with Elastic Constraints and Interior Point Methods.” *Health Care Management Science* 6 (1): 5–16.
- Küfer, K.-H., A. Scherrer, M. Monz, F. Alonso, H. Trinkaus, T. Bortfeld, and C. Thieke. 2003. “Intensity-modulated radiotherapy – A large scale multicriteria programming problem.” *OR Spectrum* 25:223–249.
- Lahanas, M., E. Schreibmann, and D. Baltas. 2003. “Multiobjective inverse planning for intensity modulated radiotherapy with constraint-free gradient-based optimization algorithms.” *Physics in Medicine and Biology* 48 (17): 2843–2871.
- Rockafellar, R. T. 1970. *Convex Analysis*. Princeton University Press, Princeton, New Jersey.
- Sayin, S. 2000. “Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming.” *Mathematical Programming* 87:543–560.
- Shao, L. 2005. “A survey of beam intensity optimization in IMRT.” *Proceedings of the 40th Annual Conference of the Operational Society of New Zealand*. 255–265.

Decomposition and Pricing Methods for solving MILP Model for an Integrated Fishery

M. Babul Hasan

Dept. of Management, University of Canterbury
Christchurch, New Zealand
b.hasan@mang.canterbury.ac.nz

and

John F. Raffensperger

Dept. of Management, University of Canterbury
Christchurch, New Zealand
john.raffensperger@canterbury.ac.nz

October 2006

Abstract

In this paper, we investigate Dantzig-Wolf decomposition (DWD) and decomposition based pricing (DBP) to generate upper bounds on the mixed integer fishery model (Hasan & Raffensperger, 2006). The formulation of the integrated fishery model resulted in a large-scale mixed integer linear program (MILP), consisting of a fishing subproblem, a processing subproblem, and complicating side constraints. To solve such a large MILP, with thousands of decision variables and constraints, we develop a new decomposition based pricing method by decoupling trawler scheduling and processing. We propose a decomposition based O'Neill pricing (DBONP) to improve the solution from DBP. Numerical results for several planning horizon models are presented.

Key words: Lagrangean relaxation; decomposition; pricing; fishing trawler scheduling; processing.

1. Introduction and Literature Review

To optimize the operations of a fishery consisting of several harvesting and manufacturing (processing) facilities, the manager needs a MILP. For large model, the optimization of the entire problem may not be efficient, since it sometimes requires long time for the solution of a total problem.

The natural complexity of MILP has led to much research in approximation methods for these problems. LP relaxation and LR are commonly used to generate bounds. Computing such bounds is an essential element of the commonly used branch-and-bound algorithm. The bounds are generally computed by solving either a relaxed or dual bounding subproblem of the original problem. Although the bound from the LP relaxation is commonly used, it is often too weak to be effective. LR has been widely used for about two decades in many practical applications. Well-chosen LR strategies usually provide tighter bounds than that of the LP relaxation.

The literature on relaxation approaches and its application is enormous. Here, we quote the papers that directly relate to the fishery model, and that introduce novel ideas, new modelling approaches and decomposition algorithms. Held and Karp [10, 11] first used

a Lagrangean problem based on minimum spanning trees to devise a successful algorithm for travelling salesman problem. Geoffrion [6] introduced the name for this idea as “Lagrangean relaxation”, developed relevant theories and explored the usefulness of LR in the context of branch-and-bound methods for mixed integer linear programs. Fisher [4] reviews the LR and documented a number of successful applications of this method.

Quota-based integrated commercial fisheries own fishing trawlers, processing firms, and fish quotas. To maintain and improve these fisheries resources and their utilization, the fisheries have to be viewed as a total system from the fish in water to the fish in plate. This highlights on the co-ordination of fishing, trawler scheduling, and processing of quota based integrated commercial fisheries. Mikalsen and Vassdal [14] developed a multi-period LP model for one month production planning for smoothing the seasonal fluctuations of fish supply. Jensson [12] developed a product mix LP model to maximize profit of an Icelandic fish processing firm over a five period planning horizon. Gunn, Millar and Newbolt [8] developed a model for calculating the total profit of a Canadian company with integrated fishing and processing. Hasan and Raffensperger [9] developed a MILP to co-ordinate trawler scheduling, processing plans, and labour allocation for an integrated commercial fishery of New Zealand.

Unfortunately, none of the above papers discussed the solution procedure of the large scale complex model. In this paper, we develop three pricing methods include a decomposition based pricing (DBP), a decomposition based O’Neill’s pricing (DBONP). We present numerical examples to illustrate the application of the proposed methods.

The remainder of the paper is organized as follows. In section 2, we briefly present the fishery model, its Lagrangean relaxation (LR). Section 3 presents the DBP procedure. In section 4, we present DBONP. Section 5 compares DBP and DBONP.

2. The Fishery Model

In this section, to avoid detraction from the essence of this paper, we briefly discuss the MILP model of Hasan and Raffensperger (2006).

(P) Maximize

$$= - \left(\sum_p \sum_a \sum_t \sum_u \sum_v (t-u) V_{t,v} w_{p,a,u,t,v} + \sum_i \sum_l \sum_t \sum_v \sum_a C_{a,i,t,v} f_{a,i,l,t,v} + \sum_i \sum_l \sum_t I_t z_{i,l,t} \right) + \left(\sum_i \sum_j \sum_l \sum_t P_{i,j,l} s_{i,j,l,t} - \sum_t Lr_t yr - \sum_t Lo y o_t - \sum_i \sum_j \sum_l \sum_t J_t r_{i,j,l,t} \right)$$

subject to

Fishing constraints

$$f_{a,i,l,t,v} = \sum_p \sum_u ET_{a,i,u,t,v} \times FR_{i,l} \times w_{p,a,u,t,v} * FRaw_{a,i,v} \text{ for all } a, i, l, t \text{ and } v. \tag{1a}$$

$$\sum_a \sum_p \sum_{t=2}^{N_v} w_{p,a,u,t,v} + wr_{1,v} = 1 \text{ for all vessel } v. \tag{1b}$$

$$\sum_a \sum_p \sum_{u=1}^{\max\{1, t-N_v\}} w_{p,a,u,t,v} + wr_{t-1,v} - wr_{t,v} - \sum_a \sum_p \sum_{t1=t+1}^{\min\{t+N_v, T\}} w_{p,a,t1,v} = 0 \text{ for all } v. \tag{1c}$$

$$\sum_i \sum_l z_{i,l,t} \leq MI \text{ for all } t. \tag{1d}$$

$$q_{a,i,t-1} - \sum_l \sum_v f_{a,i,l,t,v} = q_{a,i,t} \quad \text{for all } a, \text{ and } i \text{ and } t. \quad (1e)$$

Processing constraints.

$$LR_{i,l,t} \leq \sum_j F_{i,j} x_{i,j,l,t} \leq UR_{i,l,t} \quad \text{for all } i, l \text{ and } t. \quad (2a)$$

$$LM_{i,j,l,t} \leq x_{i,j,l,t} \leq UM_{i,j,l,t} \quad \text{for all } i, j, l \text{ and } t. \quad (2b)$$

$$r_{i,j,l,t-1} + x_{i,j,l,t} - s_{i,j,l,t} = r_{i,j,l,t} \quad \text{for all } i, j, l \text{ and } t. \quad (2c)$$

$$\sum_j \sum_l r_{i,j,l,t} \leq MIP \quad \text{for all } i \text{ and } t. \quad (2d)$$

$$z_{i,l,0} = z_{i,l,T} \quad \text{for all } i, l, \text{ and } t. \quad (2e)$$

$$\sum_i \sum_j \sum_l H_{i,j} x_{i,j,l,t} - yr - yo_t \leq 0, \quad \text{for all } t. \quad (2f)$$

$$LAR_t \leq yr \leq UAR_t \quad \text{for all } t. \quad (2g)$$

$$yo_t \leq Rt \times yr \quad \text{for all } t. \quad (2h)$$

Complicating constraints

$$z_{i,l,t-1} + \sum_a \sum_v f_{a,i,l,t,v} - \sum_j F_{i,j} x_{i,j,l,t} = z_{i,l,t} \quad \text{for all } i, l, \text{ and } t. \quad (3)$$

$$f_{a,i,l,t,v}, w_{p,a,u,t,v}, wr_{t,v}, q_{a,i,t}, x_{i,j,l,t}, s_{i,j,l,t}, r_{i,j,l,t}, yr, yo_t, \geq 0 \quad (4a)$$

$$w_{p,a,u,t,v} \in \{0,1\}, wr_{t,v} \in \{0,1\} \quad (4b)$$

The above MILP problem consists of a fishing subproblem and a processing subproblem, complicated by the small set of constraints (3). Using LR, we can relax these side constraints. It will then be easier to solve, and the objective value will be an upper bound (since it is a maximization problem) on the optimal value of P .

2.3 Lagrangean relaxation for the fishery model

We now present the LR of the fishery mode in algebraic notations in detail. For all i, l , and t , let $\lambda_{i,l,t}$ be a non-negative vector of multipliers for the constraint set 3, then the LR of the fishery model can be defined as:

$$\begin{aligned} (PR_\lambda) \quad & \text{Maximize} \\ & - \left(\sum_p \sum_a \sum_t \sum_u \sum_v (t-u) V_{t,v} w_{p,a,u,t,v} + \sum_i \sum_l \sum_t \sum_v \sum_a C_{a,i,t,v} f_{a,i,l,t,v} + \sum_i \sum_l \sum_t I_t z_{i,l,t} \right) \\ & + \left(\sum_i \sum_j \sum_l \sum_t P_{i,j,l} s_{i,j,l,t} - \sum_t Lr_t yr - \sum_t Lo yo_t - \sum_i \sum_j \sum_l \sum_t J_t r_{i,j,l,t} \right) \\ & - \sum_{i,l,t} \lambda_{i,l,t} \left(z_{i,l,t-1} - z_{i,l,t} + \sum_a \sum_v f_{a,i,l,t,v} - \sum_j F_{i,j} x_{i,j,l,t} \right) \end{aligned}$$

subject to $1a - 1e, 2a - 2h, 4a-4b.$

The potential usefulness of LR is largely determined by how near its optimal solution is to that of the integer program P . This furnishes a criterion to choose an appropriate λ . The ideal choice would be to take λ as an optimal solution of D (Fisher [4]).

3. Decomposition based pricing (DBP)

In this section, we develop a new DBP technique for the efficient solution of the fishery model. Mamer and McBride (2000) first develop a DBP procedure for solving a multi-commodity flow problem. We extend their concept for solving the fishery model. In DBP, instead of using the subproblem to produce an extreme point of the relaxed polytope for inclusion in a master, we include the optimal basic columns of the subproblems into the restricted master. We then solve the restricted master to obtain an improved primal basic feasible solution to the original problem, and to obtain a new set of dual prices. We then use the dual solutions obtained from this master in the subproblems. We continue this process until no new variable comes to the restricted master. Constructing the restricted problem in this fashion assures a basic feasible solution to the original problem, and the size of the restricted master tends to be small.

3.1 DBP for the fishery model

Using LR, we first relax the trawler scheduling and processing subproblems as in section 2. Let $\theta_{i,l,t}$ be the simplex multipliers associated with the inventory balance constraint (3) in the restricted master.

Restricted master (M^k):

Maximize

$$= - \left(\sum_p \sum_a \sum_t \sum_v (t-u) V_{t,v} w_{p,a,u,t,v} + \sum_i \sum_l \sum_t \sum_v \sum_a C_{a,i,t,v} f_{a,i,l,t,v} + \sum_i \sum_l \sum_t I_t z_{i,l,t} \right) + \left(\sum_i \sum_j \sum_l \sum_t P_{i,j,l} s_{i,j,l,t} - \sum_t Lr_t yr - \sum_t Lo y o_t - \sum_i \sum_j \sum_l \sum_t J_t r_{i,j,l,t} \right)$$

Subject to 1(a) – 1(e) and 2(a) – 2(g)

$$f_{a,i,l,t,v}, w_{p,a,u,t,v}, wr_{t,v}, q_{a,i,t} \geq 0 \text{ and } w_{p,a,u,t,v} \in \{0,1\}, wr_{t,v} \in \{0,1\}.$$

$$x_{i,j,l,t}, s_{i,j,l,t}, r_{i,j,l,t}, yr, y o_t \geq 0. \in I^k$$

We present the proposed DBP procedure as:

3.2 Algorithm

The proposed DBP algorithm for the fishery model is:

Step 0: Initialize. Set iteration $k = 1$. Pick a set of prices $\theta_{i,l,t}^k$ (let $\theta_{i,l,t}^0$ is zero).

Step1: Use LP Relaxation and solve subproblem S_1^k and solve subproblem S_2^k . For $x^i > 0$ put i in I^k , where $I^k = \{i : x^i > 0 \text{ in } S_1, \text{ and } S_2 \text{ for any iteration } 1, 2, \dots, K\}$

Step 2: Solve M^k and get dual prices $\theta_{i,l,t}^k$.

Step 3: For stopping criterion, check one of the following ways:

- (i) If $v(S_1^k + S_2^k) = v(M^{k+1})$, then stop. or
- (ii) If no new positive variables come to the restricted master, then stop.

Else go to step 1.

Step 4: Solve the Final restricted master problem as an IP.

3.3 Numerical Example

Depending on (i) the initial feasible solution and (ii) stopping criterion, we solve the model in four different ways as described below.

(i) **Initial dual prices:** We create an initial dual solution $\theta_{i,l,t}$ in three ways.

I1: Start with zero initial dual prices, $\theta_{i,l,t} = 0$ for all i, l, t .

I2: Solve the LP relaxation of the original IP problem, and let $\theta_{i,l,t}$ be the dual price of the relaxed inventory balance constraint.

(ii) **Stopping criterion:** We use two different criteria for stopping the iterations.

SC1: Stop when subproblem profit equals the restricted master profit. Here we solve the trawler scheduling subproblem as an LP.

SC2: Stop when no new variables come into the restricted master problem. Here we solve the trawler scheduling subproblem as an IP.

In Table 1, we compare the number of iterations, computational time, number of variables in the restricted master and optimal profit obtained from 5, 10, 15, 20, 25, and 30-period models by DBP with zero initial dual prices (I1) and stopping criteria 1 (SC1). We also calculate the percentage of solution gap as

$100 \times (\text{IP solution} - \text{DBP solution}) / \text{IP solution}$. We observe that, the DBP with I1 and SC1 has only 0.14% of solution gap.

Planning Horizon	Iterations	Solution time (s)	Number of variables		DBP solution	% solution gap
			Original problem	Restricted master		
5	26	156	2,193	1,308	\$522,763.50	0.00%
10	29	257	4,423	2,815	\$1,065,775.00	0.00%
15	32	341	6,803	4,272	\$1,579,440.00	0.16%
20	29	365	9,333	5,691	\$1,874,097.30	0.32%
25	29	414	11,989	7,026	\$2,119,938.20	0.09%
30	25	944	16,139	8,115	\$2,293,803.30	0.31%

Table 1: DBP results with I1 and SC1.

Table 2 shows the DBP results of 5, 10, 15, 20, 25, and 30-period models with I2 and SC2. It has 0.13% of solution gap.

Length of planning Horizon	Number of Iterations	Solution time (s)	Number of variables in restricted master	DBP solution	% solution gap
5	28	208	1,282	\$522,764	0.00%
10	28	252	2,724	\$1,065,775	0.00%
15	35	373	4,092	\$1,579,466	0.16%
20	29	359	5,420	\$1,875,597	0.24%
25	35	534	6,540	\$2,120,282	0.08%
30	30	952	7,623	\$2,292,894	0.35%

Table 2: DBP results with I2 and SC2.

We observed that the solutions obtained from all the cases of the proposed DBP procedure are very close the original solution. Also the comparison of these tables show that the classical LR results of improved bounds, when the subproblem is not naturally integral, does not follow analogously for DBP. This is because the master does not produce convex combination of subproblem solutions. We also notice that, the best

average percentage solution gap is 0.13% and is obtained with I2 and SC2. The second best percentage solution gap is 0.14% obtained with I1 and SC1.

From Table 3 shows DBP takes fewer iterations and much less time to solve the fishery model than DWD. Table 1 shows that the number of variables in the restricted problem is much less than that of the original problem. That is why DBP takes less time to solve the fishery model than that of DWD.

	DWD	DBP
Number of Iterations	1168	26
Computational time	3:58:49	0:02:58
Subproblem1 profit	\$432,138.0	\$432,132
Subproblem2 profit	\$90,628.5	\$90,631.5
Total profit	\$522,763.5	\$522,763.5
Master profit	\$522,763.5	\$522,763.5

Table 3: Comparison of the number of iterations, and computational time taken by DBP & DWD to solve a 5-period model.

4. DBONP

In this section, we propose an algorithm to improve the solutions obtained by the DBP procedure proposed in Section 3. We name the proposed technique as decomposition based O’Neill pricing (DBONP). The DBONP is based on the theorem of (Gomory & Baumol, 1960), O’Neill price (2005), and decomposition based pricing. We incorporate O’Neill price with decomposition based pricing in the proposed algorithm.

4.1 O’ Neil’s price

Based on the theorem of Gomory and Baumol (1960), O’ Neill *et. al* (2005) developed a technique for constructing a set of linear prices from solving a MILP and an associated LP. They first solved a MILP, set the integer variables to their optimal values, and then removed the integrality constraints to convert the MILP to an LP. They used the dual prices obtained from this LP to form an efficient contract.

4.2 Decomposition based O’Neill’s pricing (DBONP)

In this section, we present our proposed decomposition based O’Neill’s pricing (DBONP) method for the efficient solution of the for fishery model.

Loop 1. We first relax the integer restrictions from the variables, the complicating side constraints (inventory balance constraint (3)) and apply decomposition based pricing algorithm developed in Section 5 and obtain the final restricted master as an LP.

Loop 2. We then solve this final restricted master as an IP, set the integer variables to their optimal values, and convert it to an LP.

Let $\theta_{i,l,t}$ for all i, l and t , be the dual prices associated with the central constraints and $\theta_{p,a,u,t,v}$, and $\theta_{2,t,v}$ be the dual prices associated with the integrality constraints.

Then the fishing subproblem for the fishery can be presented as:

Trawler scheduling subproblem S_1^k

Maximize

$$\begin{aligned}
&= - \left(\sum_p \sum_a \sum_u \sum_t \sum_v (t-u) V_{t,v} w_{p,a,u,t,v} + \sum_i \sum_l \sum_t \sum_v \sum_a C_{a,i,t,v} f_{a,i,l,t,v} + \sum_i \sum_l \sum_t I_t z_{i,l,t} \right) \\
&- \sum_{i,l,t,k} \theta_{i,l,t} \left(z_{i,l,t-1} - z_{i,l,t} + \sum_a \sum_v f_{a,i,l,t,v} \right) \\
&- \sum_p \sum_a \sum_u \sum_t \sum_v \theta 1_{p,a,u,t,v} (w_{p,a,u,t,v} - w_{p,a,u,t,v}^*) - \sum_t \sum_v \theta 2_{t,v} (wr_{t,v} - wr_{t,v}^*)
\end{aligned}$$

subject to (1a) – (1e)

$$f_{a,i,l,t,v}, z_{i,l,t}, w_{p,a,u,t,v}, wr_{t,v}, q_{a,i,t} \geq 0 \text{ and } w_{p,a,u,t,v} \in \{0,1\}, wr_{t,v} \in \{0,1\}.$$

The restricted master in the second loop takes the form.

Restricted Master (M^k):

Maximize

$$\begin{aligned}
&= - \left(\sum_p \sum_a \sum_t \sum_v V_{t,v} w_{p,a,u,t,v} + \sum_i \sum_l \sum_t \sum_v \sum_a C_{a,i,t,v} f_{a,i,l,t,v} + \sum_i \sum_l \sum_t I_t z_{i,l,t} \right) \\
&+ \left(\sum_i \sum_j \sum_l \sum_t P_{i,j,l} s_{i,j,l,t} - \sum_t Lr_t yr - \sum_t Lo y o_t - \sum_i \sum_j \sum_l \sum_t J_t r_{i,j,l,t} \right)
\end{aligned}$$

Subject to 1(a) – 1(e) and 2(a) – 2(g)

For positive integer variables

$$w_{p,a,u,t,v}^k = w_{p,a,u,t,v}^{k*} \quad (5)$$

$$wr_{t,v}^k = wr_{t,v}^{k*} \quad (6)$$

$$f_{a,i,l,t,v}, w_{p,a,u,t,v}, wr_{t,v}, q_{a,i,t} \geq 0$$

$$x_{i,j,l,t}, s_{i,j,l,t}, r_{i,j,l,t}, yr, yo_t \geq 0. \in I^k$$

4.3 DBONP algorithm

We now write the pricing algorithm DBONP in detailed. It has two loops. Loop1 uses decomposition based pricing as in section 3 to get the final restricted master. In loop2, we solve the final restricted master as an IP, we set the integer variables to their optimal values, and we convert the restricted master to an LP. Then we solve this LP master to obtain new dual prices, and use the dual prices to solve subproblems. This procedure terminates when no new variable is found.

LOOP1

Step 0: Initialize. Set iteration $k = 1$. Pick a set of prices $\theta_{i,l,t}^k$ (let $\theta_{i,l,t}^0$ is zero).

Step 1: Solve subproblem S_1^k and solve subproblem S_2^k as IP. For $x^i > 0$ put i in I^k , where $I^k = \{i : x^i > 0 \text{ in } S_1, \text{ and } S_2 \text{ for any iteration } 1, 2, \dots, K\}$

Step 2: Solve M^k as LP and get dual prices $\theta_{i,l,t}^k$ and pass it to subproblems.

Step 3: If $v(S_1^k + S_2^k) = v(M^{k+1})$, then stop. Else go to step 1.

LOOP2

Step 4: Solve the restricted master problem as an IP.

Step 5: Fix x^i . For positive $x^* = 1$, and for zero variables $x^* = 0$.

Step 6: Solve master with fixed x^i as LP. Get dual prices $\theta_{i,l,t}, \theta_{1,p,a,i,l,t,v}, \theta_{2,t,v}$ and pass it to subproblems.

Step 7: Solve the subproblems with the dual prices obtained from step 6. If no new variables enter into the restricted master, then stop. Else go back to step 4.

4.3.1 Numerical example

Solving different planning horizons, we compare the solutions with that of obtained from original MILP, LP relaxation, and DBP. Results are presented in Table 4. We observe that a 5-period, a 10-period or a 25-period model has no solution gap. But a 15-period, a 20-period or a 30-period model has slight percentage of solution gap. For example a 30-period model has only 0.02% solution gap. The average percentage solution gap of six different planning horizon models is only 0.04%.

Length of planning Horizon	Number of variables	Number of Iterations	Solution time (s)	DBP solution	DBONP solution	% solution gap
5	489	29	217	\$522,764	\$522,764	0.00%
10	1,284	27	216	\$1,065,540	\$1,065,775	0.00%
15	2,229	33	345	\$1,579,309	\$1,579,570	0.15%
20	3,324	48	912	\$1,874,097	\$1,878,580	0.08%
25	6,440	45	796	\$2,120,282	\$2,121,887	0.00%
30	6,938	44	3562	\$2,293,803	\$2,300,230	0.02%

Table 4: Comparison of the optimal solutions obtained from DBP procedure, DBP and O’Neill’s pricing method to that of IP solution.

From Table 7, we observe that all the solutions obtained from the proposed pricing method are either equal to the optimal solutions (5-period, 10-period, 25-period models) or very close to the optimal solutions (15-period, 20-period and 30-period models).

To see why this little difference in the total profit is, we compare the optimal solution of the original problem with that of obtained from the DBONP. We observed that the number of trawler trips in the proposed pricing method coincide with that of the original problem. But we notice a little difference in the period of landings. For example, in the original problem of a 30-period model, trawler 1 lands its catch on period 4, 7, 10, 14, 18, 22, 26, and 30. On the other hand, in the DBONP method, in a 30-period model, trawler 1 lands its catch on period 4, 7, 11, 15, 19, 23, 26, and 30. Results are shown in Figure 1 and Figure 2. As a result, we notice a slight change in the processing accordingly.

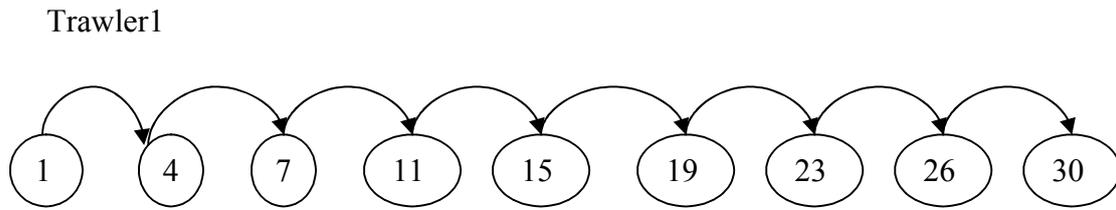


Figure 1: Scheduling of trawler 1 in the proposed improved method

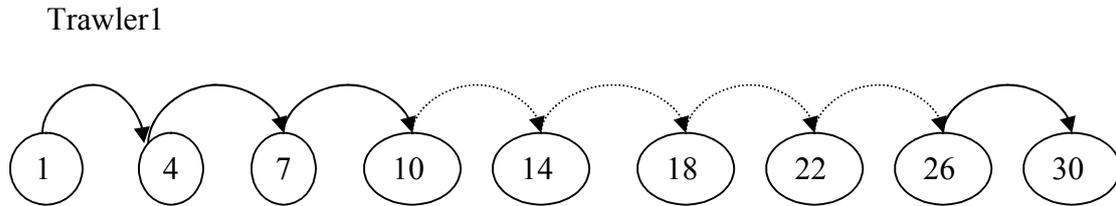


Figure 2: Scheduling of trawler 1 in the original problem

5. Comparison of DBP and DBONP

In this section, we compare the number of iterations, computation time, number of variables and optimal values obtained by solving different planning horizon models by DBP and DBONP. From Figure 3, 4 and 5, we observe that DBONP takes a higher number of iterations and higher computation time but produces better solutions than that of DBP.

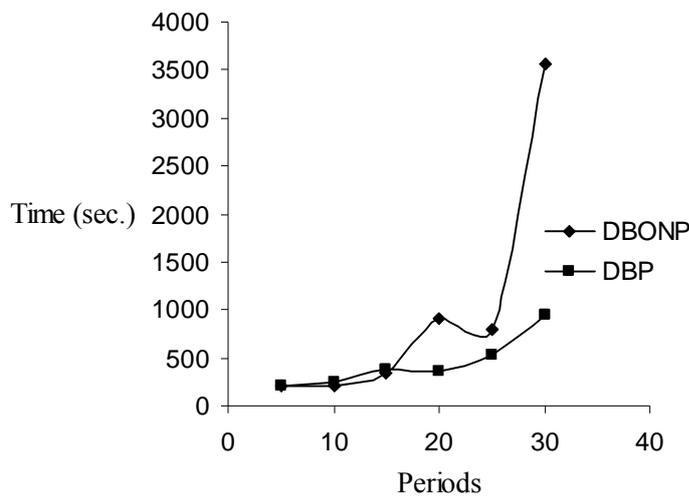


Figure 3: Comparison of solution time required to solve different planning horizons by DBP, DBONP.

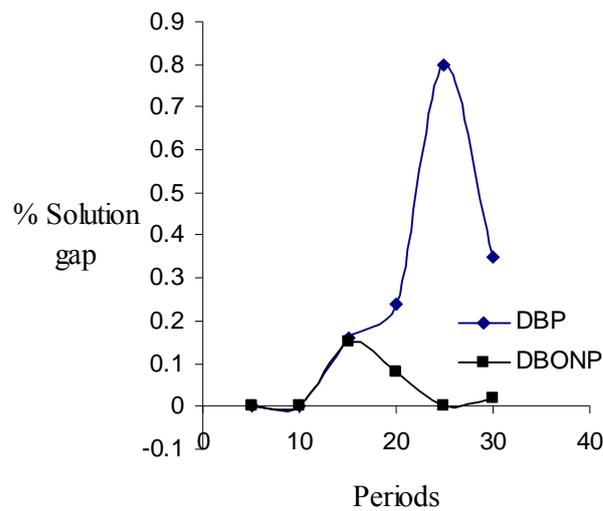


Figure 4: Comparison of percentage solution gap of DBP and DBONP.

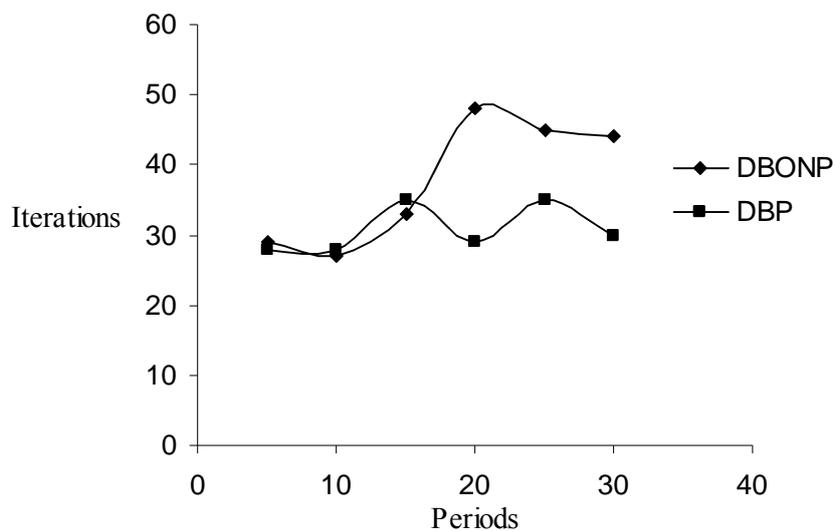


Figure 5: Comparison of number of iterations required to solve different planning horizons by DBP, and DBONP.

6. Conclusion

In this paper, we proposed new pricing methods for solving large-scale MILP fishery model. The model consists of a fishing subproblem, a processing subproblem, and complicating side constraint. For the efficient solution of such a large MILP with thousands of decision variables and constraints, we developed three pricing methods including a decomposition based pricing (DBP), and a decomposition based O’Neill’s pricing (DBONP). Solving different planning horizon models, we discovered that, though DBONP took longer time to solve, it yielded better solutions than all other methods. Numerical results for several planning horizon models are presented.

7. References

- Clement. 2004. *New Zealand commercial fisheries: The atlas of area codes and TACCs*, Clement & associates Ltd., 98 Vickerman st. Nelson, New Zealand.
- Dantzig, G.B.1963. *Linear programming and extension*, Princeton university press, U.S.A.
- Fisher, M.L.1971. Optimal solution of scheduling problems using Lagrangean multipliers, Part I, *Operations Research* **21**:1114-1127.
- Fisher, M.L.1981. The Lagrangean relaxation method for solving integer programming problems, *Management Science*, **27 (1)**: pp. 1-18.
- Fourer, R., D.M. Gay, and B.W. Kernighan. 1993. *AMPL: A modelling language for mathematical programming*, Curt Hinrichs, 511 Forest Lodge Road, Pacific Grove, CA 93950, USA.
- Geoffrion, A.M.1974. Lagrangean relaxation for integer programming, *Mathematical Programming, Study 2*: 82-114.
- Guignard, M. and S. Kim. 1987. Lagrangean decomposition: A model yielding stronger Lagrangean bounds, *Mathematical Programming* **39**: 215-228.
- Gunn, E.A, H. H. Millar, and S. M Newbold. 1991. A model for planning harvesting and marketing activities for integrated fishing firms under an enterprise allocation scheme, *European Journal of Operational Research* **55**: 243-259.
- Hasan, M.B. and J. F. Raffensperger. 2006. A mixed integer linear program for an integrated fishery, *ORiON*, **22(1)**: pp. 19-34.
- Held, M., Wolfe, P. and Crowder, H.P. 1974, Validation of subgradient optimization, *Math. Programming*, **6**, pp. 62-88.
- Held, M. and Karp, R.M., 1970, The travelling salesman problem and minimum spanning trees, *Operations Research*, **18**, no. 6, pp. 1138-1162.
- Jensson, P.1988. Daily production planning in fish processing firms, *European Journal of Operational Research* **36**: 410-415.
- Mamer, J.W. and R.D. McBride. 2000. "A decomposition-based pricing procedure for large-scale linear programs: An application to the linear multi-commodity flow problem," *Management Science* **46(5)**: 693-709.
- Mikalsen, B. and T. Vassdal. 1981. *A short term production planning model in fish processing*, in *Applied operations research in fishing*, K.B. Haley, ed. New York, NY, Plenum Press: 223-233.
- Millar, H. H. and E.A. Gunn. 1992. A two-stage approach to planning harvesting and marketing activities integrated fishing enterprises, *Fisheries Research* **15**:197-215.
- New Zealand Official Yearbook, 2004/2005, New Zealand Government, Wellington.
- O'Neill, R.P., P.M. Sotkiewicz, B.F. Hobbs, M.H. Rothkopf, and W.R. Stewart. 2005. "Effective market-clearing prices in markets with non-convexities," *European Journal of Operational Research* **164**:269-285.
- Randhawa, S. U. 1994. *Integrating simulation and optimization: an application in fish processing industry*, Proceedings of The Winter Simulation Conference, J.D. Tew, S. Manivannan, D.A. Sadowski, and A.F. Seila (eds.):1241-1247.
- Shepardson, F. and Marsten, R.E., 1980, A Lagrangean relaxation algorithm for the two duty period scheduling problem, *Management Science* **26(3)**: 274-281.

Quality vs. Power?

In defence of academic values - Community OR in action

John Paynter, Jim Sheffield, David Sundaram and Dan Trietsch
Department of Information Systems and Operations Management
University of Auckland, New Zealand
j.paynter, j.sheffield, d.sundaram, d.trietsch @auckland.ac.nz

Abstract

The value of tertiary education world-wide is under threat from many forces - external and internal. Curriculum changes are forced through based more upon expediency and power structures than upon quality of programmes or quality of the workplace considerations. Despite being a "research-led world-class institution" the University of Auckland is not immune to these forces. This paper examines the case of changing the curricula offerings and downsizing a department on the basis of past and forecasted EFTS and FTE figures. On the one hand there are the proposal and confirmation documents from the managing authorities and on the other responses based on a re-examination of the figures. Staff members formed a community in gathering information, examining and presenting the data and responding to the proposal and confirmation. The responses also referred to the University's charter and quality statements, the need to re-examine the assumptions behind the proposal and other alternatives that could be explored. In this light, the findings of the confirmation document and the final outcome are examined using a variety of analytic tools.

1 Introduction

The Department of Information System and Operations Management (ISOM), was the first of a number of departments at the University of Auckland (UoA) facing redundancies. As presented June 27 2006, and since confirmed 15 August, the plan decreased staff numbers from 31 by 8. Individual staff were informed on September 22nd that their positions were disestablished and given notice on October 27th. Since then similar proposals have been put to the Faculties of Arts and Education.

After rapid growth in the late 90s and turn of the century, student numbers in the department had declined over the past two years. This was due to several reasons. There was a world-wide downturn in Information Systems (IS) and Information Technology (IT) education. New Zealand was no exception. Although the dotcom crash had resulted in some job losses its impact was more perceived than actual. However the attractiveness of an IT career had been dissipated. People who wanted to work with computers now could do so within their own disciplines as IT became increasingly commoditised. Enrolments of domestic students in IT programmes declined. Ironically too, women were becoming more and more represented in the student body (over 50%), but the numbers of women taking IT had declined from 20% to 10%. Much of the NZ market was taken up by foreign full fee-paying students. The increase in the value of the NZ\$, bad publicity overseas and increasing numbers of students being educated in China all contributed to a falling off in numbers. A similar perception was true for Operations Research (do any departments of that name still exist?) and related disciplines. Attempts to increase the

attractiveness of the programmes offered, by for instance making them interdisciplinary had been seen as another way to overcome these problems. For instance at UoA the Bachelor of Business and Information Management (BBIM) was offered at the satellite campuses and a major in electronic commerce was also offered. Far from increasing student numbers, enrolments cannibalised existing programmes and diluted the teaching effort and increased the amount of administration required.

A large new business building currently under construction is said to have increased the budgetary pressure on the Business School and The University of Auckland as a whole. Thus ISOM and the allied BBIM programme were seen as prime candidates for cuts.

The announcement (the proposal) was timed for the mid-semester break when staff were marking final exams, preparing for the next semester and attending conferences or otherwise trying to do research. Several key personnel were away on these activities and the department heads had not been forewarned. Staff held meetings in ones and twos as well as general department meetings to formulate an action plan.

We report how the skills of individuals and of the team were brought together to respond to the proposal and the subsequent confirmation document. Other action is still continuing today. Thus this is an example of "Community Operations Research". In most such places where it is practiced (Ritchie et al, 1994) one or more outside experts or a single expert who is part of the community work with a community towards an action. What is of interest here is that all of the participants were experts, although as is common with most such organisations, they are discouraged from using their expertise in-house.

The next section outlines how community OR is practiced, then how it was applied in this instance. Results from the analysis are then presented, followed by a discussion of what did and did not occur and what might have happened. Finally we conclude with thoughts as to what else could have been done.

2 Literature review

2.1 Community Operations Research

"Any community group or organisation is faced with the challenge of organising itself and its work as effectively as possible in order to stand the best chance of achieving its aims and objectives. This is made all the more difficult as such groups often have to work within tight constraints imposed by limited resources (time, people and money), and in a frequently hostile environment" (Ritchie et al, 1994, p1).

Just as there is no widely accepted definition of OR, although "the science of doing better" is gaining credence, there is no understandable definition of community OR. Ritchie et al in the cases they present attempt to convey what constitutes community OR. Whereas there is a substantial body of community OR in the UK, no such community appears to exist in New Zealand. One of the writers in the Ritchie book, Gerald Midgley is now a member of ORSNZ and works at ESR, Christchurch where he is given the opportunity to put some of the 'principles' in practice. Others surely use the knowledge and skills gained in OR in working within their communities. Some examples could include Andrew Mason's objection to a storm water pipe at Herne Bay or John Paynter's funding applications for community groups such as Newton Central Primary School and Auckland Masters Athletics Association.

2.2 Fairness, Equity, Justice, and Truth

Under the Employment Relations Act (ERA) employers must act fairly and equitably with employees affected by such restructuring decisions as represented here (Department of Labour, 2000). The ancient Hebrews (Deuteronomy 25:13-15) mandated that “Thou shalt not have in thy bag diverse weights, a greater and a smaller: Neither shall there be in thy house a greater bushel and a smaller. Thou shalt have a just and a true weight, and thy bushel shall be equal and true.” The key principles on which our arguments rest are fairness, equity, justice, and truth.

3 Method

3.1 Why Community OR?

The definition of Community OR epitomises the situation in which ISOM found itself. It was the greatest challenge in the life of the department. The aim was to refute the proposal (or in the least to minimise job losses). The timeframe given to reply was tight and already busy staff working under pressure from changes to the courses within the programme and the structure of the programme itself (i.e. a change from a standard of 7 courses per student per year to 8; and for staff from teaching 3 to 4 such courses per year), and budgetary constraints. E.g. lecturers would take tutorials and do an increased share of the marking as budgetary constraints limited the employment of coordinators, tutors and markers. Whereas UoA staff external to the department were required to release (some) information requested, they were not forthcoming in saying what was available. Thus the information was drip fed into the analysis that was being formed to refute the proposal. Some facts about the process¹ are only becoming apparent now.

3.2 The approach

A twin approach was adopted. We were concerned that aspects of both the process and substance are not being addressed properly in this matter. The union (AUS) would focus on the process argument and members of the department would focus on the substance of the proposal.

3.2.1 Process

One of the ISOM staff invited Bill Hodge, a lecturer at UoA Law School to address the department as to the situation. He stuck to the general situation rather than details of this case, except in the matter of ‘tenure’, where some staff appeared to be shocked to learn that this did not give immunity to being made redundant. Bill reiterated that the employer (the Vice Chancellor in this instance) must follow the procedures set down in handling such a situation. The employer must also be seen to engage in true consultation. However the employment court would not consider arguments that the employer did not have the right to manage their own business, no matter how badly the employer might seem to be doing so. He did indicate, however, that all employees on contracts must be terminated before considering the disestablishment of permanent positions. (Perhaps as a response to this, in a surprising subsequent move, the administration offered permanent positions to all employees that have been on contracts—including both academics who engage in research and others whose roles were essentially limited to support functions. Eventually, most of these positions were retained at the expense of researchers with years of seniority.) The department

¹ Simpson Grierson, Faculty of Business and Economics Restructuring, 8 November 2006.

members present (including those on conference calls from overseas) felt that the process could be challenged on the grounds that the process was rushed. The legal situation with regard to cases etc is not further considered here, as this paper is about the community's response, rather than that of legal counsel.

3.2.2 Substance

Members of the department used their contacts and expertise to gather information and to request clarification and information from the Business School management. The issues faced were complex and it was argued (unsuccessfully for the most part) that more time was required to gather, digest and process the complex information. Some used their specific OR skills. E.g. as an appendix to the ISOM response, a stochastic model was developed showing the change in enrolments subsequent to the change in pathways under the new degree structure. At all times it was stressed that the document produced to the University's tight deadlines was a living document – that more would be added as information became available.

In addition the individual members of the department were encouraged to submit individual responses. Many chose to use the strengths of their own disciplines. For instance Tiru Arthanari (2006) emphasised quality, his research and teaching area, while John Paynter (2006) whose field lay in interdisciplinary studies stressed the need to explore other options than redundancy.

4 Results

4.1 Management Proposal and our Response

Two concerns expressed in the introductory statement of the management proposal were with respect to 'revenue' and 'expenditure'. Hence in our response to the initial proposal we focussed on the financials since the exercise was seen to be fundamentally driven by the perception of a "significant drop in student revenue" and therefore a corresponding drop in "surplus revenue" (contribution). We attempted to allay these concerns by using a number of sources to prove that the ISOM department:

- had **stabilised revenue** (2006 revenue was keeping up with 2005 revenue)
- had significantly **reduced expenditure** (by more than ½ a million dollars)
- had generated **surplus revenue** in excess of the 2006 budget (a surplus revenue of more than 6 million dollars and in excess of ½ million dollars when compared with the surplus revenue of 2005)
- had **increased enrolments** (an increase of more than 600 individual course enrolments when compared with 2005)
- had strengths that could be leveraged to **increase revenue** considering the current supply and demand statistics from MOE, NZIS, Gartner, etc. (particularly in Enterprise Systems, Supply Chain Management, and Business Intelligence)

In this paper we just focus on 'surplus revenue' as it is at the heart of the employer's argument and at the heart of the response. A clear upward trend is seen when we view the surplus revenues generated by the ISOM department from 2000-2006 (Figure. 1). The reduction in 2004 is attributable to the factors mentioned in the introductory section of this paper. Significant efficiency measures introduced in the department had resulted in an increase in surplus revenues in 2006. Surplus revenues were higher in 2006 compared to 2000, 2001, and 2005. This is especially laudable considering the shrinkage of overseas students that the University is facing. The graph was created

while we were in the middle of this exercise and the authors have every reason to believe that the ‘surplus revenue’ for 2006 is as displayed in Figure 1 or even better.

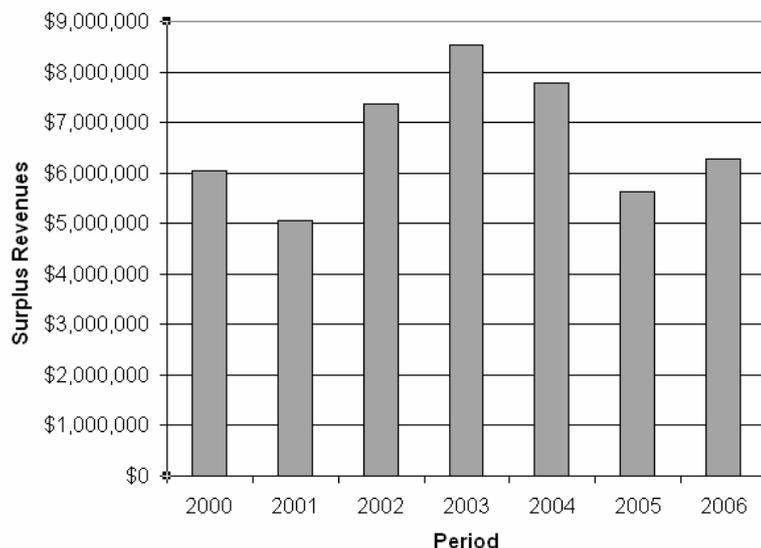


Figure 1: Surplus Revenues generated by ISOM from 2000-2006 (Source: Faculty Financial Data)

4.2 Confirmation Document and ISOM Response

At the heart of the case put forward by the University and Faculty to cut 8 ISOM Staff members (25% of the Department) was the argument that the current Student-Staff Ratio (SSR) of the ISOM department is below the target SSR of **23**. In answer

to the confirmation document from management the ISOM department put forth a response that attempted to:

- propose a **FAIR** and **EQUITABLE** SSR for ISOM (Section 4.3) using the latest data as provided in the “University Staff Remuneration & Resourcing: A Comparison of New Zealand and Selected International (Australia, Canada, England, USA) Data” prepared for The New Zealand Vice Chancellors Committee and The Association of University Staff of New Zealand by Deloitte’s on December 2005.
- prove decisively that a target SSR of 23 for the ISOM Department is **INEQUITABLE** and **UNJUST** whatever be the standard of comparison (Section 4.4).

4.3 A fair and equitable SSR for ISOM

ISOM has Information System (INFOSYS) courses that are funded at Category B, higher than the Category A funding that applies to Operations Management (OPSMGT) courses (and to the rest of the Business School). In the following paragraphs we attempt to calculate a fair and equitable SSR for ISOM using the latest data as provided in the December 2005 Deloitte’s Report. An analysis of enrolments from 2001-2006 shows that ISOM had 22399 enrolments in INFOSYS² (IS) courses and 3176 enrolments in OPSMGT (OM) courses over this period. This works out to 87.6% INFOSYS and 12.4% OPSMGT enrolments.

Since INFOSYS courses are funded at the level of ‘Information Technology’ courses we use the G8 universities SSR for ‘Information Technology’ departments as a guide for INFOSYS. Namely **16.2** (Table 1). Since OPSMGT courses are predominantly funded at the level of ‘Management & Commerce’ courses we use the G8 universities SSR for ‘Management & Commerce’ departments as a guide for OM.

² Note that over the years many of the OPSMGT courses have had significant increases in IT content that they have been recast as INFOSYS courses. This has also led to many of the OPSMGT staff teaching INFOSYS courses.

Namely: **26.3** (Table 1). We integrate these SSR figures with the ratio of IS to OM courses as below:

Target SSR for ISOM based on G8 comparison = $16.2 \times 87.6\% + 26.3 \times 12.4\% = \mathbf{17.45}$

Discipline Grouping	G8 Universities	All Australian Universities
Natural & Physical Sciences	16.2	15.5
* Information Technology	16.2	19.5
Engineering & Relating Technologies	18.5	17.8
Architecture & Building	16.6	21.7
Agriculture, Environmental & Related Studies	13.6	12.9
Health	10.9	14.6
Education	19.4	23.0
* Management & Commerce	26.3	27.7
Society & Culture	21.6	21.4
Creative Arts	14.9	18.2

Source: Data extracted from AVCC analysis – using onshore students only and a combination of teaching only and teaching and research staff.

Table 1: Student:Staff Ratios by Discipline Groupings (Deloitte, 2005)

A simple sensitivity analysis shows that even with a change of the IS:OM ratio from 87.6:12.4 to 80:20 will result only in a marginal increase in the target SSR. Target SSR for ISOM based on 80:20 split between IS:OM and in comparison with G8 = $16.2 \times 80\% + 26.3 \times 20\% = 18.22$.

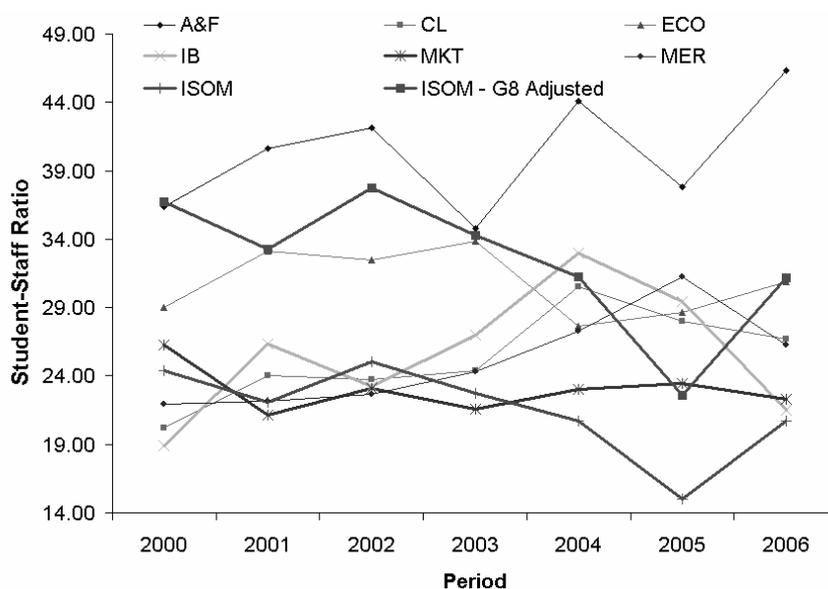


Figure 2: Comparing ISOM’s raw and G8 adjusted SSR’s with sister departments from the Faculty of Business and Economics

Figure 2 shows ISOM’s SSR performance in comparison to other departments in the Business School. We show ISOM both with and without correction for the G8 comparison. The corrected data were adjusted by a factor of $26.3/17.45 = 1.51$. Except for an outlier in 2005—which the Department had previously

addressed by reducing the staff by 8 FTE (20%)—ISOM is consistently among the top three performers (as also indicated by its financial contribution).

Furthermore, targeting ISOM was also the result of not understanding variation. The Consultation Document was based on projections that the student

numbers will continue to fall whereas in fact the number of students started to increase as of 2006 and there was no statistically valid evidence that it was going to decline. Coming on top of a recent 20% reduction, targeting ISOM for another 25% (i.e., 40% in about a year) was not only an example of incorrectly comparing oranges and apples but also not understanding variation and acting in haste.

4.4 Comparisons of mandated SSR with similar departments and institutions

The current SSR for ISOM in 2006 based on data provided by the Faculty is **20.7**. The fair SSR based on the latest Deloitte's report of **December 2005** is **17.45**. The University and Faculty SSR target for ISOM based on 2002 data (Figure 3) is **23**.

Figure 3 illustrates that the University and Faculty in setting the target SSR for ISOM had used outdated data in a selective fashion.

STUDENT STAFF RATIO (SSR)

	UoA SSR 2002	UoA SSR 2003	G7 Ratio	G7 Adjusted Ratio	UoMel SSR 2002	UoQ SSR 2002	UoNSW SSR 2002	UoSyd SSR 2002	UoWA SSR 2002	UoAdel SSR 2002	Monash SSR 2002
ISOM	25.0	26.3	26.9	23.7	25.7	22.7	22.2		21.0	25.5	29.9

POSTGRADUATE PERCENTAGE (PG)

	UoA PG 2002	UoA PG 2003	G7 %	G7 Adjusted %	UoMel PG 2002	UoQ PG 2002	UoNSW PG 2002	UoSyd PG 2002	UoWA PG 2002	UoAdel PG 2002	Monash PG 2002
ISOM	17.3%	19.5%	27.4%	28.4%	6.0%	18.1%	33.0%		16.0%	2.2%	37.6%

POSTGRADUATE RESEARCH PERCENTAGE (PGR)

	UoA PGR 2002	UoA PGR 2003	G7 %	G7 Adjusted %	UoMel PGR 2002	UoQ PGR 2002	UoNSW PGR 2002	UoSyd PGR 2002	UoWA PGR 2002	UoAdel PGR 2002	Monash PGR 2002
ISOM	2.4%	3.1%	2.2%	2.3%	2.7%	1.4%	2.2%		5.5%	5.5%	1.9%

	UoMel	UoQ	UoNSW	UoSyd	UoWA	UoAdel	Monash
SSR	Exact Match	Best Fit	Exact Match	Not Available	Exact Match	Excluded	Excluded
PG	Exact Match	Best Fit	Exact Match	Not Available	Exact Match	Excluded	Exact Match
PGR	Exact Match	Best Fit	Exact Match	Not Available	Exact Match	Excluded	Exact Match

Figure 3: 2002 based ISOM Benchmarking Data used by the University and Faculty to arrive at the target of 23 (Source: Email 17/7/2006 from the Director of Administration)

4.4.1 Comparisons with departments of a similar nature such as Computer Science, University of Auckland and IT Departments in G8

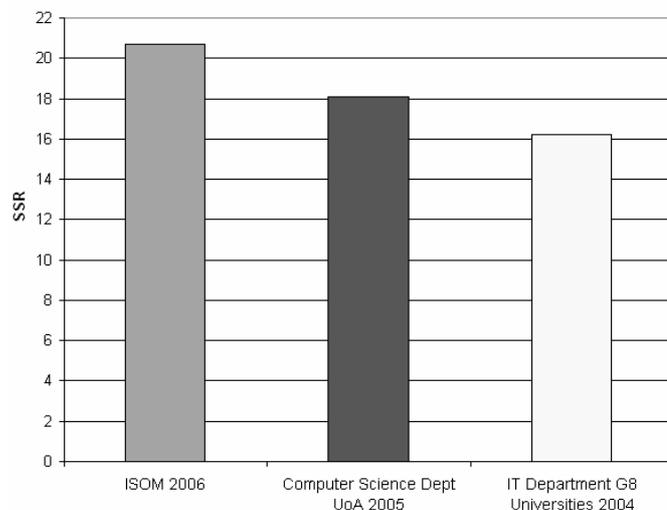


Figure 4: Comparing current ISOM SSR with CS/IT Departments in UoA & G8

Approximately 87.6% of student enrolments are at Category B funding which is similar to the Computer Science department at UoA and also the IT departments at G8 and other Australian universities.

As Figure 4 illustrates the current ISOM SSR is well above the SSR of the Computer Science Department at UoA and significantly higher than the SSR of the IT departments of the G8 universities. The

stature of the UoA demands that it compares itself with the G8 universities and not the average of all the Australian Universities.

4.4.2 Comparison to targets set by the University of Auckland

The UoA submission of its 'Profile' for 2006-2008 on 31st October 2005 to the *Tertiary Education Commission*

suggests that they want to "Create and maintain an outstanding teaching and learning environment". In order to achieve this they target SSRs of 17 for 2006, 16.5 for 2007, and 16 for 2008. At the same time they suggest that ISOM should achieve SSRs of 23.5 in 2007 and 23 in 2008 (Figure 5).

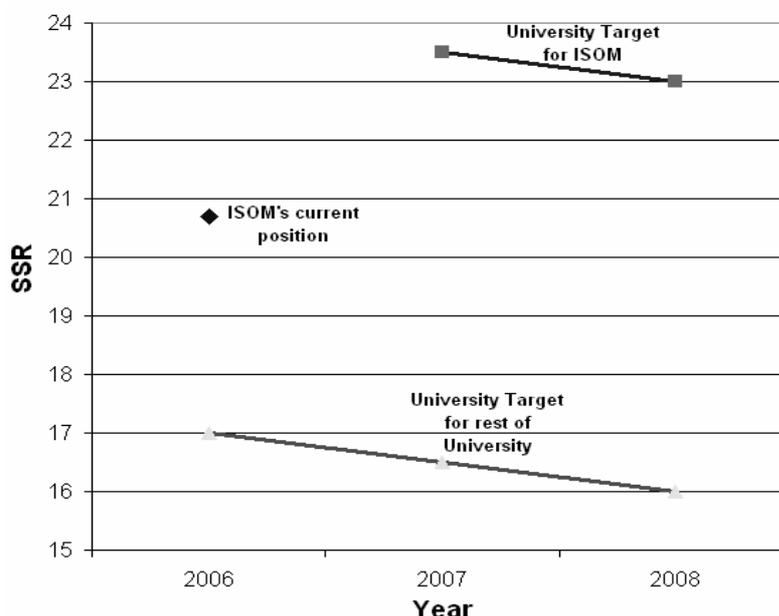


Figure 5: Comparing ISOM's current position with University of Auckland Targets

4.4.3 Comparison to University Averages (University of Auckland, NZ Universities, and other International Universities)

ISOM's current SSR is significantly higher than the SSRs of UoA, NZ Universities, and other international universities as illustrated in Figure 6. It illustrates even more clearly that the SSR target of 23 for ISOM is unjustly high compared to local and international university averages.

5 Discussion

As is pointed out by Ritchie et al, the 'numbers racket' while important is not the entire answer. "Rational analysis is not by any means the dominant paradigm when it comes to political decision-taking, and the OR practitioners must be careful to remind groups that numbers provide only a brittle shield." (p237)

The ISOM situation most closely resembled the "Aldermoor School" case (Molinero, in Ritchie et al). It was similar in the purported falling roll and although the school and in this case ISOM community put up a strong case the educational authorities made the decisions on what appeared to be political grounds rather than truly consulting with their communities. Again as a parallel the community (Aldermoor School/ISOM) "had no right to put their case to the Education Committee/Vice Chancellor, so that this was done by the LEA/Dean – by the very same people who had argued for closure/downsizing" p89.

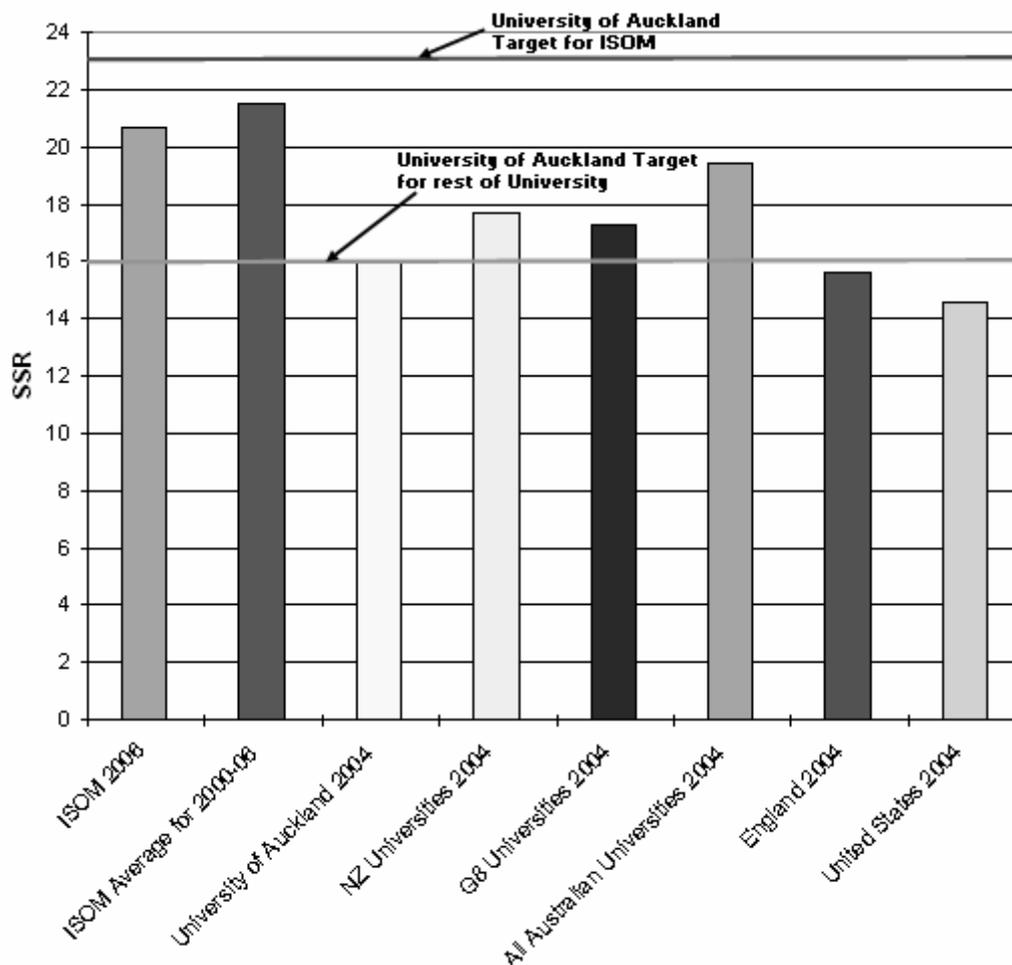


Figure 6: Comparing ISOM's SSR with local and international universities

A similar situation occurred at Hull University (per comm., Gerald Midgley) where similar financial pressures resulted in job losses, and a reduction in the ranking of the University. It took years of pressure to remove the driving forces of the change and then to see an improvement in rankings.

However the University's own strategic plan calls for a reduction of the present SSR ratio to 16 or lower over the next few years. The rationale behind this lowered SSR is so that the University can meet its charter towards being a world-class University. We believe that it is not in the best long-term interest of the University, nor the short-term interests of the students to dilute the quality of teaching and research by decreasing staff numbers (thus increasing the SSR).

6 Conclusions

The matters presented here are complex. There are many issues, not yet surfaced. There are ambiguities in the data and the interpretation of them. In addition only the views of two stakeholders (University of Auckland management and staff) have been expressed. Other stakeholders include student groups, industry bodies and the government that funds the University. Nevertheless, this paper shows how the community of staff in ISOM department pooled together their knowledge and skills to engage in academic debate on the issues.

The situation as outlined here is a first for New Zealand Universities. ISOM's dispute in the rounds of proposal and confirmation, followed by disestablishment and

redundancy of individual staff, appears lost. However, debate of these issues is likely to continue using other forums such as governance (UoA Council) and the legal system. The documentation of the processes, the analysis and discussion of the data, as presented here, may form a template for future actions as the battlefields move to other faculties and indeed different Universities. It is hoped that this paper contributes to the rational debate. It should be recognised that many of the softer interpersonal issues are more important in moving forward. The key to this is communication.

Acknowledgements

The authors wish to acknowledge the contributions of present and former staff, some of whom are sadly no longer with us. They have made contributions to research and teaching and service to academia, ORSNZ, many organisations through the practical application of their skills and to the community.

7 References

- Arthanari, T. (2006) Response to Consultation Document entitled “Proposal to Restructure the Department of Information Systems and Operations Management” dated 26th June 2006.
- Deloitte, (2005) University Staff Remuneration & Resourcing: A Comparison of New Zealand and Selected International (Australia, Canada, England, USA) Data, prepared for The New Zealand Vice Chancellors Committee and The Association of University Staff of New Zealand, December.
- Department of Labour (2000) New Zealand Employment Relations, Retrieved 11 November 2006 from <http://www.ers.dol.govt.nz/>
- ISOM (2006) “The Information Systems and Operations Management Departmental Response to the Consultation Document entitled “Proposal to Restructure the Department of Information Systems and Operations Management” dated 26th June 2006”.
- ISOM (2006) “The ISOM Departmental Response to the Confirmation Document entitled “Restructuring of the Department of Information Systems and Operations Management” dated 15th August 2006”.
- Mason, A. (2006), Submission in Respect of a Resource Consent Application, dated 21 July.
- Paynter, J. (2006) Response to the Consultation Document entitled “Proposal to Restructure the Department of Information Systems and Operations Management” dated 26th June.
- Ritchie, C., A. Tarket, and J. Bryant, (1994) “Community Works, 26 case studies showing Community Operational Research in action” Community Operational Research Unit Publications: No. 1.
- The University of Auckland, (2005), “Profile 2006-2008 Parts B & C”, Submission to the Tertiary Education Commission, October.
- University of Auckland (2006) Consultation Document entitled “Proposal to Restructure the Department of Information Systems and Operations Management” dated 26th June.
- University of Auckland (2006) Confirmation Document entitled “Restructuring of the Department of Information Systems and Operations Management” dated 15th August.

Safe Scheduling

Kenneth R. Baker
Tuck School, Dartmouth College
Hanover, NH 03755 USA

Dan Trietsch
ISOM Department
The University of Auckland
New Zealand
d.trietsch@auckland.ac.nz

Abstract

Traditional approaches to stochastic scheduling have yet to make an impact on scheduling practice. We review a number of results in stochastic scheduling in a historical context in order to highlight its shortcomings. We explain the logic of safe scheduling and survey some of its basic results. This includes different approaches to the formulation of stochastic scheduling problems attempting to bridge the gap between theory and practice. We also discuss some insights and potential research directions.

1 Introduction

Research in scheduling theory began just over 50 years ago, however, scheduling *practice* has not yet reached the full potential that scheduling *research* has promised. One telling example is that the vast majority of scheduling support systems, such as those implemented in major ERP systems, take a deterministic approach. To explore the gap between stochastic problems and deterministic solutions, we look critically at our theory and advocate stochastic scheduling models that address practical problems in a theoretically sound yet practical way.

What accounts for this gap between theory and practice? One key reason, we believe, is that researchers have pursued stochastic scheduling mainly as an extension of deterministic scheduling. Almost all stochastic models in the literature are based on corresponding deterministic models, and as a result, the typical stochastic model misses an important part of the problem. To use an analogy, imagine that we attempted to build stochastic inventory models by relying on deterministic lot sizing and optimizing by expectation. We might then propose the use of the formula

$$Q^* = \sqrt{\frac{2E(D)S}{H}}$$

(where $E(D)$ replaces D) as our response to the stochastic nature of demand. Q^* is indeed optimal for this simplistic model, but there are no safety stocks here! So the model fails to support the most important practical response to stochastic demand! Just as safety stocks are vital to practical inventory policies, *safety time* is vital to practical scheduling policies. Nevertheless, the optimal determination of safety times has no counterpart in deterministic scheduling. *Safe scheduling* departs from the dominant paradigm in stochastic scheduling by considering safety time explicitly. In some cases, this can be done analytically; in others, it exacerbates an already tough problem. But we believe that appropriate stochastic models—those that take safety into account—afford the best practical representation of reality. Stochastic models should provide a better balance between usefulness and tractability than deterministic models. Our guiding

principle is that a good heuristic solution of a relevant stochastic model should be preferred over an optimal solution to an oversimplified deterministic model.

In what follows we discuss the historical roots of scheduling theory, which led to the status quo. Next we summarize various approaches to safe scheduling and we briefly survey early results that represent these approaches. Finally, we speculate about future directions for research in this field. We do not present new results here, but we do provide links to two unpublished working papers—Trietsch and Baker (2006a, 2006b)—that include such results. In fact, this paper cannibalizes liberally from the latter.

2 The Historical Roots of Scheduling Research

Basic models in scheduling theory typically involve the following assumptions:

1. At time zero, n single-operation jobs are simultaneously available.
2. Each machine can process at most one job at a time.
3. Setup times are included in processing times.
4. Job descriptors are deterministic and known in advance.
5. Machines are continuously available (and do not break down).
6. Machines are never kept idle while work is waiting.
7. Once an operation begins, it proceeds without interruption.

The key parameters in the single-machine sequencing model include the processing time for job j (p_j), the due date (d_j), and the weighting factor (w_j). As a result of sequencing decisions, job j completes at time C_j , and the set of completion times is summarized in a measure of scheduling performance that serves as the model's objective function. We sometimes use bracket notation to indicate position in sequence; thus, $p[k]$ represents the processing time of the k^{th} job in sequence.

The theory addresses several sequencing objectives: minimizing total flowtime (denoted F), total lateness (L), maximum lateness (L_{\max}), total tardiness (T), maximum tardiness (T_{\max}), number of tardy jobs (U), etc. Several of these objectives also have weighted versions (denoted F_w , etc.) that have been addressed as well. In addition, these objectives are all *regular* measures of performance, which means that if we alter a schedule so that one job finishes later and no jobs finish earlier, then the objective function cannot get better. With respect to regular measures, assumptions 6 and 7 are redundant. Stochastic models are associated with relaxing assumptions 4 and 5.

It was clear to the pioneers of scheduling theory that practical scheduling involves both combinatorial complexity and stochastic complexity, but, with few exceptions, stochastic analysis was postponed to a later stage of development. Evidence that this separation was related to the magnitude of the combinatorial analysis lies in the fact that when PERT was introduced (Malcolm et al., 1959), it involved sophisticated stochastic analysis right from the start. Clark (1961) showed how to approximate the moments of a project makespan distribution in spite of the complex dependencies involved. Stochastic analysis was possible because sequencing decisions were not involved. When sequencing issues were subsequently studied in the context of PERT (Wiest, 1964), the analysis reverted to deterministic models. As a result, to date, there are project-related papers that address stochastic issues, some of which also address safety (e.g., Britney, 1976), and there are project-related papers that address sequencing (Herroelen, 2005, provides an extensive survey), but we know of no deep analysis of the two factors in combination.

Aside from PERT, stochastic issues were addressed by queueing and simulation. Queueing theory placed little emphasis on sequencing effects and the main results were approximations that showed how variation is transformed into waiting time. The simplest relevant queueing model—a generalization of the M/M/1 model to job shops—

could also be used to calculate distributions of waiting times (Jackson, 1963), but more general models rarely produced such extensive results. For the study of sequencing issues, simulation was more successful (Conway, et al., 1967). At the start, a job set was generated randomly and stored, complete with release times (arrival times), processing times, due dates, and routing information. For this stored set, various dispatching rules were used, and performance measures were compared. Simulations are still used to compare heuristic dispatching rules, and their usefulness lies in the fact that practical sequencing often involves precisely such rules. With few exceptions, however, these simulations treat job descriptors as deterministic once the job set is determined. In practice, however, we cannot sequence by processing times before we know them; instead, we must rely only on some feature of their distributions, such as their means. Thus, in a technical sense, even simulation studies do not entirely represent stochastic reality.

In the 1970's, the NP-complete equivalence class was discovered, and suddenly the computational limits of most deterministic scheduling models became clear. When the ramifications for scheduling theory percolated through the research community, several responses emerged. Some researchers continued to work with complexity theory, identifying NP-complete models and defining the boundary between easy and hard problems (Garey and Johnson, 1979), while others focused on devising efficient heuristics. Still others took these developments as an opportunity to redirect their efforts, and a renewed interest developed in stochastic scheduling. Not much later, the study of earliness/tardiness (E/T) models became popular (Baker and Scudder, 1990). These two areas developed independently of each other: stochastic theory addressed regular measures, whereas E/T models dealt with non-regular measures. (Scheduling theory might have developed differently if E/T models had matured before stochastic scheduling emerged as a major field, because one way to model safety time is by stochastically balancing earliness and tardiness penalties.)

As mentioned, the stochastic problem arises when we relax assumptions 4 or 5. Literally, that means permitting job processing times, weights, and due dates to all be uncertain, but in practice it is unusual to encounter uncertainty with respect to weights or even due dates. Thus, for practical purposes, the stochastic problem arises when we treat processing times as uncertain; other parameters are known with certainty or represent decisions. Randomness in processing times could occur due to differences in operator skills, quality problems that require rework, setup adjustments that take random time, or variations in input materials. Randomness in processing times could also arise because equipment breaks down and unscheduled maintenance must be carried out. Therefore, relaxing assumption 5 could lead to similar problems as relaxing assumption 4. For the purposes of exposition, we ignore the technical distinctions between these two cases.

Following the vast majority of published results in stochastic scheduling, our default assumption is that processing time distributions are independent. However, this assumption is often unfounded in practice, so we also look for ways to accommodate dependent processing times in our models. Probabilistic dependence may make the analysis intractable, leading us to rely on optimization by simulation methods, but simulation may be justified if we do not want to lose sight of practicality.

Another common assumption in the basic stochastic problem is that all necessary decisions are made in advance. This approach is consistent with static scheduling, where the complete schedule is determined at time zero, before the realizations of any processing times are known. Scheduling can be static even if the model is not. Static scheduling, however, precludes certain kinds of dispatching or contingent scheduling. Suppose, for example, that we know a job is very likely to take one hour but that there is a slight possibility that it will take 40 hours. We can then start

the job and run it for an hour. If the one-hour realization applies, we will have completed the job, but if the 40-hour realization applies, then we can postpone the rest of the job while we give priority to other work. Although potentially useful, contingent scheduling or dispatching of this sort is normally prohibited. Furthermore, even when dispatching is allowed, static scheduling rules can be used repeatedly to actually make decisions. Finally, static scheduling is often preferred in practice because it increases predictability in an uncertain environment.

Due to its historical roots, research on stochastic scheduling has focused on the same performance measures considered in deterministic scheduling (F , T , L_{\max} , T_{\max} , U , etc.) and has sought to minimize their expected values. Thus, typical stochastic models aim to minimize $E(F)$, $E(T)$, $E(L_{\max})$, $E(T_{\max})$, $E(U)$, etc. We refer to such expected-value models generally as *stochastic counterparts* of the corresponding deterministic models, although in a specific case we might instead refer, for example, to “the stochastic F -problem.” The most prevalent practical approach to such models is to substitute the expected values of the processing times for the random variables and then proceed with deterministic analysis. We refer to the resulting model as the *deterministic counterpart*.

In some cases—specifically when the objective is linear—the deterministic counterpart provides a valid solution for the stochastic problem. For instance, Rothkopf (1966) showed that $E(F_w)$ is minimized by Shortest Weighted Expected Processing Time (SWEPT) sequencing ($E(p_{[1]}) / w_{[1]} \leq E(p_{[2]}) / w_{[2]} \leq \dots \leq E(p_{[n]}) / w_{[n]}$). Adapting the optimality proof, it is also possible to show that, under the optimal sequencing rule, the value of $E(F_w)$ is equal to the optimal value of F_w in the deterministic counterpart. Furthermore, these results carry over to the minimization of expected weighted lateness because it differs from F_w by only a constant. In this case, the deterministic counterpart performs as practitioners might expect.

However, this justification of the deterministic counterpart does not extend to many models. Consider the stochastic L_{\max} -problem. Crabill and Maxwell (1969) noted that $E(L_{\max})$ is minimized by Earliest Due Date (EDD) sequencing ($d_{[1]} \leq d_{[2]} \leq \dots \leq d_{[n]}$). But the optimal value of $E(L_{\max})$ is not necessarily equal to the optimal value of L_{\max} in the deterministic counterpart. Similarly, $E(T_{\max})$ is also minimized by EDD sequencing, but again the value of the optimal objective function may not match that of the deterministic counterpart. In these two models, following Jensen’s inequality, the expected value of the objective function may exceed that of the deterministic counterpart. In such a case, we say that there is a nonnegative *Jensen gap* between the results of the deterministic counterpart and the stochastic problem. Jensen gaps are often positive in practice, but they are zero for linear functions, such as $E(F_w)$. To distinguish between piecewise linear convex functions (such as the max function) and linear ones, we define any convex function that is not linear as *meaningfully convex*. In models with meaningfully convex objective functions, we can find instances with strictly positive Jensen gaps. If we are interested in the optimal value of the objective function as well as the optimal sequence, the deterministic counterpart has shortcomings, due to the Jensen gap, even in otherwise well-solved problems such as L_{\max} and T_{\max} . It also has shortcomings for any other meaningfully convex objectives. For example, consider a generalization of the T_{\max} -problem that seeks to minimize the maximum deferral cost, $g_f(C_j)$, where the function $g_f(C_j)$ is monotone nondecreasing. Hodgson (1977), building on a classical result by Lawler, minimized the maximum expected deferral cost, $E[g_f(C_j)]$. But the problem of minimizing the expected maximum remains unsolved! To be specific, when generalizing from the minimization of $E(T_{\max})$, the criterion of interest should really be $E[\max\{g_f(C_j)\}]$, not $\max\{E[g_f(C_j)]\}$. As Crabill and Maxwell first observed, these are not the same measures. (Indeed, there is a nonnegative Jensen gap between them.) We have constructed examples in which both the optimal sequence and

the optimal value of the objective function differ according to which of these objectives is adopted. The optimal sequence is the same, however, in all cases where the cost functions, $g_j(C_j)$, are ordered (i.e., they do not intersect each other), which includes the cases optimized by EDD. Otherwise, the minimization of the expected maximum cost appears to be an unsolved problem.

One special case of Hodgson's result involves the following deferral cost:

$$g_j(C_j) = 1 \text{ if } C_j > d_j \\ = 0 \text{ if } C_j \leq d_j$$

With this definition, $E[g_j(C_j)]$ is the probability that job j is tardy. This case corresponds to minimizing the maximum probability that a job will be tardy. As originally shown by Banerjee (1965), the optimal sequence is EDD. (Hodgson's note extends that result to jobs with arbitrary precedence constraints.)

Consider the stochastic T -problem. The deterministic counterpart has attracted a lot of attention over the years, and current methods can solve large versions of the problem (Szwarc et al., 2001). However, there has been little progress on the stochastic problem. One reason is the difficulty of finding dominance conditions. With few exceptions—one of which is presented in Trietsch and Baker (2006b)—even stochastic ordering of processing times is seldom a strong enough condition to obtain useful dominance conditions. As a consequence, the branch-and-bound approaches that have been effective in the deterministic case have not been successful in the stochastic case.

Consider the stochastic U -problem, where it is also the case that the optimal sequence and the optimal value of the objective function for the stochastic problem may differ from those of the deterministic counterpart. It turns out that it is difficult to adapt the Moore-Hodgson algorithm (Moore, 1969), which solves the deterministic counterpart efficiently by partitioning the jobs into an on-time set and a tardy set. No sequencing rule is known to be dominant in the stochastic case. Furthermore, instances exist where one tardy job can cause many subsequent jobs to be tardy, an effect not captured by the Moore-Hodgson algorithm. Therefore, a general-purpose technique appears necessary for the stochastic U -problem.

3 Safe Scheduling: A Brief Survey

One approach to safe scheduling is the use of service level constraints. To illustrate, we focus on problems involving due dates. We replace the deterministic definition of "on time" by a stochastic one, thus permitting a different type of analysis. Define SL_j to be the *service level* for job j : the probability that job j completes by its due date. Let b_j denote a given target for the service level. Then the form of a service-level constraint for job j is

$$SL_j = \Pr\{C_j \leq d_j\} \geq b_j.$$

We say that job j is *stochastically on time* if its service-level constraint is met; otherwise, the job is *stochastically tardy*. A complete sequence is called *feasible* if all jobs are stochastically on time. Another approach to safe scheduling is by minimizing the economic costs associated with a schedule by expectation. This suggests that the stochastic counterpart of the E/T problem is a safe scheduling model. Indeed, it can be shown that it leads to economical service levels and is thus a more general approach than the arbitrary specification of service levels. These two approaches essentially define how to construct an objective function that does not ignore safety. To actually achieve such safety, there are two major ways: we can either control the load that we undertake so it can be executed safely or we can treat due dates and ready times as decisions to incorporate enough safety time in the schedule. This defines four combinations (Table 1).

\ Decisions Objective	Accept-Reject jobs	Allow enough safety time
Meeting service Level constraints	Balut (1973), Kise and Ibaraki (1983), Akker and Hoogeveen (2004), Trietsch and Baker (2006a)	Golenko-Ginzburg et al. (various), Trietsch and Baker (2006b)
Minimizing expected economic costs	Open (see example in Trietsch and Baker (2006b))	See text below.

Table 1. Safe Sequencing and Scheduling Classification and some Existing Results

Starting with models that involve load control, perhaps the very first paper on safe scheduling was Balut (1973) who proposed extending the classical Moore-Hodgson algorithm for a stochastic U -problem with service level constraints and normal processing times. However, Kise and Ibaraki (1983) showed that the problem is NP-hard and therefore Balut's solution was only a heuristic. Akker and Hoogeveen (2004) solved several special cases all of which exhibited stochastic ordering by efficient polynomial algorithms. Trietsch and Baker (2006a) gave a polynomial solution for any case with stochastic ordering. They also allowed processing times to be linearly associated and thus they include a case that does not require statistical independence. The economic approach with load control is an open research problem, but Trietsch and Baker (2006b) gave an example showing that beyond minimizing expected piecewise linear economic costs it is possible to reduce variance by further rejections. This leads to the idea of including quadratic tardiness penalties so that the benefit of such variance reduction can be taken into account explicitly. They also mention that the problem can be solved by dynamic programming.

Before moving on to models that involve allowing sufficient time, we need to distinguish between sequencing and scheduling. In the context of deterministic scheduling, sequencing is implied by the schedule. When dealing with regular measures we can schedule all activities as early as possible (ES in PERT terminology), and if so the schedule is also implied by the sequence. Hence, sequencing and scheduling are almost synonymous. For safe scheduling there is a distinction because for the same sequence we can devise many schedules with different safety time allowances. Thus a more complete term would be "sequencing and safe scheduling." On the one hand, because one cannot schedule without a sequence it is also permissible to use the shorter term. But on the other hand, when considering models that schedule due dates or release dates optimally, the distinction is important because most of the published results do not include explicit consideration of sequencing at all. Prominent examples are Britney (1976), Yano (1987a,b,c,d), Sarin & Das (1987), Das & Sarin (1988), Ronen & Trietsch (1988, 1993), Kumar (1989), Chu et al (1993), and also see Trietsch (2005a, 2006). Of the last two, the former shows how to estimate parameters of processing times that are linearly associated and the latter generalizes the results of virtually all the previously cited results to a general project network case where activities may be correlated (e.g., by linear association). When we include sequencing in the model, more work has been done concerning sequencing heuristics to meet service level constraints. Notably, Golenko-Ginzburg with various co-authors produced several such results starting in 1988: see Golenko-Ginzburg et al. (2003) for earlier references. See also discussion in Trietsch and Baker (2006b). Work that considers both sequencing and safe scheduling with an economic objective includes Trietsch (1993), Soroush (1999), Portugal and Trietsch (1998, 2001, 2006a, 2006b). Finally, Soroush and Fredendall (1994) presented heuristics for sequencing to minimize expected economic costs without controlling safety time.

4 Safe Scheduling Insights and Research Directions

In this section we focus on insights from models that involve the economic approach with due dates and release times as decisions. For simplicity, however, we will assume that release times are given and we only control due dates. Our first insight concerns safe scheduling for a given sequence and constitutes an economic balance principle. In any schedule, we can define the completion time as $\max\{\text{due date, actual completion time}\}$. Using a project network depiction for the progress of jobs, define the critical path as the longest path in the network. First, consider a single job. If the job is tardy we can say that the processing time is critical, but if the job is early it is the due date that is critical. Define the *criticality* as the probability of being critical and we can see that the service level is equivalent to the criticality of the due date. Trietsch (2006) considered a project with n inputs and a due date such that the cost of starting input j earlier by one time unit is c_j , the tardiness penalty of the project is c_0 , and where s is defined as the sum of all these costs. Then the optimal criticality of an input is c_j/s , unless a constraint forces a higher criticality (which will happen if the due date is dictated exogenously and is too tight). Such higher criticality is always at the expense of the service level. This result can be interpreted as a direct generalization of the newsvendor model for n inputs. Similar insights also exist with respect to other types of safety buffers; e.g., when we determine optimal capacity buffers (Trietsch, 2005b).

The single project case may be considered inappropriate for environments with multiple jobs. Nonetheless, it provides a basis for more advanced analysis. Trietsch (1993) addressed the scheduling of multiple flights into and out of a hub airport. Here each arrival feeds several departures and for this reason the optimal un-capacitated solution tends to schedule all the departures at about the same time. Arrivals, however, are scheduled with different safety time buffers so they are not clustered to the same extent. However, once we consider that the airport capacity is finite, we tend to obtain blocks of flights. When that's the case it is important to sequence flights within blocks. The following insights were obtained:

- After sequencing and re-balancing, blocks approximately maintain their un-capacitated optimal service levels.
- When more flights are involved, higher service levels are required from each flight but the total system service level—defined as the probability *all* flights will be on time—remains approximately the same.
- Nonetheless, the optimal system service level depends on the network structure (thus suggesting that the economical approach supersedes the service level chance constraint approach).
- It is possible to use a surrogate loss function to support sequencing decisions but such a surrogate must be flat near optimum.

Portougal and Trietsch (2006a, 2006b), in different contexts, yielded asymptotically optimal (AO) results. The former—concerning Johnson's problem in a stochastic environment—demonstrated that the deterministic counterpart is AO. Furthermore, when the variance and the mean are agreeable, the Johnson algorithm tended to yield low variance as well as low mean, which is highly useful for safe scheduling. This gives rise to the open research question how to characterize such cases in general; i.e., when can we expect the deterministic counterpart to be AO and when is this sufficient not only for the stochastic counterpart but also for safe scheduling?

Another important issue is the Jensen gap. We have already encountered instances of the single-machine problem where a strictly positive Jensen gap occurs. In the context of safe scheduling, the Jensen gap implies that we must anticipate a larger mean than the deterministic counterpart generates when we determine safety time.

Single-machine models illustrate the thinking that must be applied, but additional complications occur in more general scheduling models.

Consider the flowshop model with a criterion of minimizing the makespan (or some cost function based on the makespan.) In a flow shop, the makespan corresponds to the critical path in the network of operation times—the maximum of several path lengths. The expected value of the makespan is therefore subject to Jensen's inequality. In particular, the mean of the makespan is at least as large as (and typically larger than) the makespan of the deterministic counterpart. By contrast, in the single-machine model, the expected value of the makespan is identical to the makespan of the deterministic counterpart.

A second factor relates to the variance of the makespan. In the flow shop, sequencing decisions affect the variance of the makespan. By contrast, in the single-machine model, the variance of the makespan is not affected by the choice of sequence. Thus, on two counts, stochastic flowshop scheduling problems involving the makespan give rise to complications not seen in the single-machine model.

Portougal and Trietsch (2001, 2006a) explored these effects in the flow shop, both theoretically and by simulation. In a limited experiment, they varied the schedule for a given set of stochastic jobs. They found that as the makespan of the deterministic counterpart increases (due to changes in job sequence), the Jensen gap decreases but the makespan variance increases. The latter result may be surprising: a “dense” schedule—one with little idle time in the deterministic counterpart—appears to have a smaller variance than a “loose” schedule. They also ran an experiment to compare sequences where the only meaningful difference was denseness, and the results generally corroborated the same observation. This property makes it all the more valuable to find a dense schedule in stochastic makespan problems.

Perhaps the most important conclusion from this experiment was that good deterministic counterpart solutions are very valuable but only if the Jensen gap is taken into account in the schedule. Otherwise, dense schedules tend to lead to high Jensen gaps and thus to a larger deviations between plans and realizations. Such deviations may lead practitioners to question the value of careful sequencing.

Consider the Pareto Optimal (PO) set of sequences for mean and variance. In some cases it can be shown that the optimal sequence belongs to this set. In other instances the best sequence from this set provides a heuristic solution. An open research question is which models can be optimized this way. For models that cannot be optimized this way, another research question is to test the performance of the PO approach as a heuristic. A particular case in point is when the PO approach is optimal but only subject to stochastic independence. Because the stochastic independence assumption is rarely valid in practice, the question arises how good it may be as a heuristic only. For example, consider a case studied by Soroush (1999) and by Portougal and Trietsch (2006b): it is required to sequence n jobs on a single machine and set due dates for them such that the sum of E/T penalties for all jobs will be minimized. Soroush recommended a heuristic that sequences by σ_j^2 / w_j , where w_j is a weight determined by both earliness and tardiness penalty rates but not related to the mean, μ . Portougal and Trietsch showed that this heuristic is AO and no essentially different heuristic can be AO. But the model assumes stochastic independence and ignores flowtime. Once the stochastic independence assumption is removed, flowtime becomes important not only by itself but also because it affects the variances of job completion times!

In several cases we observed that stochastic ordering leads to the applicability of deterministic counterpart solutions for stochastic cases, but this is definitely not the case in general. Therefore, one research direction is to ascertain in which cases stochastic ordering is sufficient for this purpose. Note that it is often reasonable to assume

stochastic ordering and that stochastic ordering does not require the strong assumption of stochastic independence.

5 Conclusion

When stochastic models are based on deterministic counterparts without considering safety time, they fail the test of practicality. This is especially true when deterministic counterparts lead to schedules that are too optimistic. Then, the only way to use a deterministic model for a stochastic reality is to build hidden buffers into each job's estimated processing time. However, due to Parkinson's Law, such hidden safety time is often wasted. For this reason, stochastic scheduling models are not used in practice, even though we all know that reality is stochastic.

The conventional approach ignores the Jensen gap, which may cause disappointing results. Second, it ignores the need for safety time. Although deterministic models may be very useful as sequencing heuristics, they may lead to problems unless safety is addressed correctly. If we wish to fully utilize our historic investment in scheduling theory, we must pay more attention to safe scheduling. And we must account for both the Jensen gap and safety time requirements.

6 Acknowledgements

Dan Trietsch wishes to acknowledge the significant influence that his ten years of friendship and collaboration with Victor Portougal (1941-2005) had on his contribution to this paper. (For more details see:

<http://staff.business.auckland.ac.nz/staffpages/dtriets/Portougal&Trietsch.htm>.)

7 References

- Akker, J.M. van den and J.A. Hoogeveen 2004. Minimizing the number of late jobs in case of stochastic processing times with minimum success probabilities. *Journal of Scheduling* (to appear).
- Baker, K.R. and G.D. Scudder 1990. Sequencing with earliness and tardiness penalties: a survey. *Operations Research* 38, 22-36.
- Balut, S.J. 1973. Scheduling to minimize the number of late jobs when set-up and processing times are uncertain. *Management Science* 19(11), 1283-88.
- Banerjee, B.P. 1965. Single facility sequencing with random execution times. *Operations Research* 13, 358-364.
- Britney, R.R. 1976. Bayesian point estimation and the PERT scheduling of stochastic activities. *Management Science* 22(9), 938-948.
- Clark, C.E. 1961. The greatest of a finite set of random variables. *Operations Research* 9, 145-162.
- Conway, R.W., W.L. Maxwell and L.W. Miller 1967. *Theory of Scheduling*, Addison-Wesley.
- Crabill, T.B. and W.L. Maxwell 1969. Single machine sequencing with random processing times and random due-dates. *Naval Research Logistics Quarterly* 16, 549-555.
- Golenko-Ginzburg, D., A. Gonik and Z. Laslo 2003. Resource constrained scheduling simulation model for alternative stochastic network projects. *Mathematics and Computers in Simulation* 63, 105-117.
- Herroelen, W. 2005. Project scheduling—Theory and practice. *Production and Operations Management* 14, 413-432.

- Hodgson, T.J. 1977. A note on single machine sequencing with random processing times. *Management Science* 23, 1144-1146.
- Jackson, J.R. 1963. Jobshop-like queueing systems. *Management Science* 10, 131-142.
- Kise, H. and T. Ibaraki 1983. On Balut's Algorithm and NP-Completeness for a Chance Constrained Scheduling Problem. *Management Science* 29, 384-388.
- Malcolm D.G., J.H. Rosebloom, C.E. Clark and W. Fazar. Application of a technique for a research and development program evaluation. *Operations Research* 1959; 7(5):646-669.
- Moore, J.M. 1968. An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management Science* 15, 102-109.
- Portugal, V. and D. Trietsch 1998. Makespan-related criteria for comparing schedules in stochastic environments. *Journal of the Operational Research Society* 49, 1188-1195.
- Portugal, V. and D. Trietsch 2001. Stochastic scheduling with optimal customer service. *Journal of Operational Research* 52, 226-233.
- Portugal, V. and D. Trietsch 2006a. Johnson's problem with stochastic processing times and optimal service levels. *European Journal of Operational Research* 169, 751-760.
- Portugal V. and D. Trietsch 2006b. Setting due dates in a stochastic single machine environment. *Computers & Operations Research* 33, 1681-1694.
- Rothkopf, M.H. 1966. Scheduling with random service times. *Management Science* 12, 707-713.
- Soroush, H.M. 1999. Sequencing and due-date determination in the stochastic single machine problem with earliness and tardiness costs. *European Journal of Operational Research* 13, 450-468.
- Szwarc, W., A. Grosso, and F. Della Croce 2001. Algorithmic paradoxes of the single-machine total tardiness problem. *Journal of Scheduling* 4, 93-104.
- Trietsch, D. 1993. Scheduling flights at hub airports. *Transportation Research, Part B (Methodology)* 27B, 133-150.
- Trietsch, D. 2005a. The effect of systemic errors on optimal project buffers, *International Journal of Project Management*, 23, 267-274.
- Trietsch, D. 2005b. From Management by Constraints (MBC) to Management by Criticalities (MBC II), *Human Systems Management* 24, 105-115 (Special Issue on Theory of Constraints).
- Trietsch, D. 2006. Optimal feeding buffers for projects or batch supply chains by an exact generalization of the newsvendor model, *International Journal of Production Research* 44(4), 627-637.
- Trietsch, D. and K.R. Baker 2006a. "Minimizing the number of tardy jobs with chance constraints and linearly associated stochastically ordered processing times." Department of Information Systems and Operations Management (ISOM), University of Auckland, Working Paper No. 298, February. http://staff.business.auckland.ac.nz/staffpages/dtriets/AHpaper_Trietsch_Baker.pdf
- Trietsch, D. and K.R. Baker 2006a. "Safe Scheduling." ISOM, University of Auckland, Working Paper No. 313, August. <http://staff.business.auckland.ac.nz/staffpages/dtriets/SafeSchedulingTrietschBaker.pdf>
- Trietsch, D. and F. Quiroga 2005. "Balancing stochastic resource criticalities hierarchically for optimal economic performance and growth." ISOM Working Paper No. 256, May (under revision for *Quality Technology and Quantitative Management*).
- Wiest, J.D. 1964. Some properties of schedules for large projects with limited resources. *Operations Research* 12, 395-418.

Minimizing Average Tardiness and Machine Outsourcing Cost

Alex J. Ruiz-Torres ^{1a} Francisco Lopez ^{1b} and Johnny Ho ²

¹ Department of Information and Decision Sciences
University of Texas at El Paso
El Paso, Texas 79968
United States

² D. Abbott Turner College of Business
Columbus State University
Columbus, Georgia 31907
United States

^a aruiztor@utep.edu

^b fjlopez@utep.edu

² ho_johnny@colstate.edu

Abstract

Decisions involving production outsourcing must consider both the production costs to be charged by the contract manufacturer and the potential loss of customer goodwill due to late orders. This article deals with the problem of finding solutions that consider tradeoffs between outsourcing cost and average tardiness, an important measure of lost customer goodwill. Assumptions include that outsource costs are a function of machine use and that the planning organization owns no production equipment; instead, the production function is completely outsourced. Furthermore, the planning organization has flexibility in terms of the quantity of parallel production resources to outsource and it controls the assignment of jobs to these parallel production resources. The article presents lower bounds for the problem, and several approaches to generate tradeoff solutions.

Keywords: Scheduling, Supply Chain Scheduling, Outsourcing, Pareto Solutions, Multi-Criteria, Average Tardiness, Parallel Machines, Machine Utilization.

1 Introduction

Outsourcing of business functions continues to gain popularity given it provides multiple benefits, including allowing organizations to focus on their core competencies while taking advantage of the expertise and efficiency of the companies from which they outsource. Outsourcing also provides extra capacity without capital investments or hiring of personnel (Puich and Walker (2004)), but outsourcing is not without its "costs," including loss of control, possible higher product cost, possible reduction of flexibility, and the creation of dependency on suppliers.

Capacity flexibility is a significant benefit of manufacturing outsourcing that is not always fully understood. By knowing the available capacity of the manufacturing outsource provider and matching it with production requirements, planners can determine the best set of outsource resources to use. In addition, outsource decisions are

tightly linked to costs and other customer-service criteria, so it is highly beneficial for production planners to be able to select from a set of possible solutions, each involving a specific cost and a specific level of customer service. One reason for this is that, usually, there are tradeoffs between cost and customer service criteria. Assume for example that a set of orders can be completed with an average tardiness of 5 hours if four workcells are outsourced. It could be that the orders can be completed with an average tardiness of 2 hours if five workcells are outsourced instead. Of course, it is assumed that using four workcells is cheaper than outsourcing five.

The assumptions in this article break with some traditional ones since the scenario considered here is one where the production resources do not belong to the planning organization. Instead, the planning organization outsources them and pays accordingly. Research work that considers outsource resources is relatively scarce, but includes Lee et al. (2002), and Kim (2003). The article by Lee et al. (2002) integrates scheduling and outsourcing decision making with the objective of minimizing late orders. Their model considered a set of jobs that require multiple operations, which have specific due dates, and which can be scheduled either on internal or outsource machines. The authors propose a genetic algorithm that determines the machine allocation, the operational sequence of each job, and the assignment of jobs to outsource machines. Kim (2003) describes a control model that determines a dynamic control policy to support decision making when contract manufacturing is available. The model helps determining the amount to be outsourced and the level of processing to be performed by the outsourcing resource.

In addition to the outsource characteristic, another element in this article is a special focus on the tardiness criterion. Research on tardiness related criteria has been conducted by multiple authors both for the single and parallel machine settings, always assuming that a set of n jobs has to be scheduled. A recent paper by Cao et al. (2005) addressed the problem of minimizing tardiness scheduling cost. These authors investigated the problem of simultaneously selecting and scheduling parallel machines to minimize the sum of the total tardiness and the machine "holding" cost. They allowed the machines to have different capacities (unequal machines) and base cost; thus, each job has a processing time value depending on the machine that process it. The goal is minimizing an objective function that combines the weighted tardiness of all jobs and the cost of using the machines. This article builds on the work by Cao et al. (2005) as it considers the cost of machine use in the problem. However, in the proposed problem all machines are identical and the decision is about how many machines to outsource and for how long, given that the time that a machine is used is a possible component of the machine cost.

2 Problem Definition

We assume that there are n one-operation jobs, $N = \{1, \dots, n\}$, which cannot be preempted or divided. There are available m identical outsource machines, $M = \{1, \dots, m\}$. A job j requires p_j units of time to be completed and is due by time d_j . We assume that processing times and due dates are integer values and that $d_j \geq p_j$ so there are no inherent late jobs. Let c_j be the time when job j is completed according to a schedule. The tardiness of job j , t_j , is $t_j = \max [c_j - d_j, 0]$. The total tardiness of any given schedule is $T = \sum_{j \in N} t_j = \sum_{h \in M} T_h$, where T_h is the tardiness on machine h for that schedule. Let k represent the number of machines used in a schedule ($k \leq m$) and let $K = \{1, \dots, k\}$ be the set of machines used, so $K \subseteq M$. Let MC_h be the completion time of the last job on machine h . Assume that the Total Outsource Cost (TOC) of a schedule is $TOC = \sum_{h \in K} \phi (MC_h)$, where $\phi(x)$ is a cost function of x , the time that an outsource machine is used. Let $\phi(x)$ be generically defined as $\phi(x) = a + bx^v$. The first parameter, a , can be thought

of as a set-up cost. Note that values of y greater than one do not seem to be realistic (a function with $y > 1$ implies that the more a machine is used, the higher is the cost per extra unit of time used). On the other hand, positive values of y , smaller than one indicate a less-than-linear relationship, like for example in the case of volume discounts. The following is additional notation used in the remainder of this article.

This work addresses the problem of simultaneously minimizing the total tardiness (T) and the Total Outsource Cost (TOC). The objective is to generate sets of efficient schedules for these two criteria. Consistent with scheduling literature, we propose the following notation for this problem: $P \parallel \#(T, TOC)$. Note that the form of the cost function $\phi(x)$ determines if there are tradeoffs between T and TOC . In addition to trying to consider realistic cases, this is one more reason for us to only consider functions with $0 < y \leq 1$, which may result in trade-offs as will be illustrated later.

2.1 Lower Bounds: TOC

The following analysis will allow us to determine a lower bound for TOC and to make some interesting observations. Consider the cost function $\phi(x) = a + bx^y$ and assume that $0 < y \leq 1$ and $P > k > 1$, where $P = \sum_{j \in N} p_j$. We use the function $a + bx^y$ as it provides a simple mechanism to represent a non-linear cost function based on the load assigned to the machine (x). Under most practical cases $y > 0$, otherwise the cost is fixed regardless of use, which is not realistic. Furthermore, for a majority of situations, the average usage cost will be constant ($y = 1$) or decrease ($y < 1$) based on the concept of economies of scale and volume price discounts. Clearly when $y = 1$, the machine costs are solely based on the number of machines used, and not related to the specific load of a machine. Finally, in a small set of cases it may be possible to have increasing costs as use increases ($y > 1$), particularly after a set use has been reached (after x hours, cost increase per unit of use). These increasing costs could be used as to maintain flexibility as to serve other customers. We note that this paper does not address such cases, but will be considered in future research.

Suppose that the number of machines to use for a particular schedule, k , is fixed, with $k \leq m$. Consider the following three possibilities for distributing the load P among the k machines: 1) *Unbalanced load*: load $P - k + 1$ units of process time on one of the machines and one unit of process time on each of the remaining $k-1$ machines; 2) *Semi-Balanced load*: load $P - k$ units of process time on one machine, 2 units of process time on the next machine, and one unit of process time on the remaining $k-2$ machines; 3) *Balanced load*: load P equally among all machines (let us suppose just for this example that units of processing time can be divided evenly among the machines). TOC can be re-expressed as follows. In the first case, $TOC = ka + b[(P - k + 1)^y + (k-1)]$; in the second case, $TOC = ka + b[(P - k)^y + (k-2) + 2^y]$; and in the third case, $TOC = ka + bk(P/k)^y$. Note that in all three expressions the first term is ka , so if we are to compare these three $TOCs$, this term can be eliminated. Likewise, b is a common factor in all three second terms, so it can also be removed (as long as $b > 0$). Consequently, the magnitude of TOC is determined by $(P - k + 1)^y + (k-1)$ for the first case, $(P - k)^y + (k-2) + 2^y$ for the second case, and $k(P/k)^y$ for the third case. Note that if $y = 1$, all three expressions reduce to P , implying that the three proposed ways of loading the machines result in identical $TOCs$ and therefore there are no TOC tradeoffs with a fixed k . Theorem 1 demonstrates that the cost of an Unbalanced load is less than or equal to the cost of a Semi-Balanced load, while Theorem 2 demonstrates that the cost of a Semi-Balanced load is less than or equal to the cost of a Balanced load.

Theorem 1. If $0 < y \leq 1$ and $P > k > 1$ then $(P - k + 1)^y + (k-1) \leq (P - k)^y + (k-2) + 2^y$.

PROOF. Since for $y = 1$ the inequality obviously holds, we proceed with $0 < y < 1$. Let $x = P - k$. The problem is equivalent to showing $(x + 1)^y - x^y \leq 2^y - 1$

for $x > 1$. We will use a standard differential calculus technique. Let $f(x) = (x + 1)^y - x^y$. Since the derivative $f'(x) = y[(x + 1)^{y-1} - x^{y-1}]$ is always negative for $x > 1$ (because x^{y-1} is decreasing), and $f(1) = 2^y - 1$, we have $f(x) < 2^y - 1$ for all $x > 1$. q.e.d.

Theorem 2. If $0 < y \leq 1$ and $P > k > 1$ then $(P - k)^y + (k - 2) + 2^y \leq k(P/k)^y$.

PROOF. Since the function $f(x) = x^y$ is concave down, it satisfies the condition $f(a_1x_1 + \dots + a_kx_k) \geq a_1f(x_1) + \dots + a_kf(x_k)$ for any x_1, \dots, x_k and any nonnegative a_1, \dots, a_k such that $a_1 + \dots + a_k = 1$. In particular, if $a_1 = a_2 = \dots = a_k = 1/k$ and $x_1 = P - k$, $x_2 = 2$ and $x_3 = x_4 = \dots = 1$, we get $(1/k)(P - k)^y + (1/k)2^y + (1/k)(k - 2) \leq [(P - k + 2 + k - 2)/k]^y$, so $(P - k)^y + 2^y + k - 2 \leq k(P/k)^y$. q.e.d.

Next we demonstrate that TOC is an increasing function of k when unbalanced loading is assumed (for any value of k we want to minimize its TOC). In other words, as the number of machines used increases, TOC increases: $TOC(k_1) < TOC(k_2)$ if $k_1 < k_2$. Thus the TOC function for k_1 machines assuming unbalanced loading is $ak_1 + b(P - k_1 + 1)^y + b(k_1 - 1)$ and we must demonstrate that if $k_1 < k_2$, then $ak_1 + b(P - k_1 + 1)^y + b(k_1 - 1) < ak_2 + b(P - k_2 + 1)^y + b(k_2 - 1)$. The proof is similar to that in Theorem 1.

Theorem 3. If a and b are positive constants and $k_1 < k_2$, then $ak_1 + b(P - k_1 + 1)^y + b(k_1 - 1) < ak_2 + b(P - k_2 + 1)^y + b(k_2 - 1)$.

PROOF. With the notation $x = P - k_2 + 1$ and $h = k_2 - k_1$, the inequality becomes $(x + h)^y - x^y < (a + b)h/b$. If $f(x) = (x + h)^y - x^y$, then $f'(x) = y[(x + h)^{y-1} - x^{y-1}] < 0$ for all $x \geq 1$, and $f(1) = (1 + h)^y - 1$. Since $(1 + h)^y < 1 + h$, for all $x \geq 1$ $f(x) < f(1) < h < (a + b)h/b$ and the inequality is proven. q.e.d.

While Theorem 3 shows that TOC is an increasing function of k , it turns out that T is a decreasing function (or at least non-increasing) of k , as shown in the following theorem. Let $K_1 \subseteq M$ and $K_2 \subseteq M$ be two subsets of M . Let S_1 be any schedule of the jobs in N using the K_1 machines and T_{S_1} be its corresponding total tardiness.

Theorem 4. If $k_1 < k_2$, then there exists a schedule S_2 on K_2 , with total tardiness T_{S_2} , such that $T_{S_1} \geq T_{S_2}$.

PROOF. Since $k_1 < k_2$, assume without loss of generality that $K_1 \subset K_2$. Then there is a machine in K_2 , h^* , that according to schedule S_1 does not have any job assigned. There are two possible cases: 1) there is a machine that according to S_1 processes a late job, j^* , in position two or later; or 2) no late job is processed in second or later positions on any of the K_1 machines. In the first case, since j^* is not the first job on its corresponding machine, $c_{j^*} > p_{j^*}$. Transferring j^* to h^* implies that it will be the first job on h^* . Thus, its completion time becomes $c_{j^*}^* = p_{j^*}$. Hence, $c_{j^*} > p_{j^*} = c_{j^*}^*$, so $c_{j^*} - d_{j^*} > c_{j^*}^* - d_{j^*}$: the tardiness of j^* is greater in S_1 than after its transfer to h^* , schedule that we can consider S_2 (even if the vacuum left by the removal of j^* is kept. It is obvious that if this vacuum is filled by moving the third and later jobs forward, the lateness of those jobs may reduce but cannot increase). Thus, it is clear that $T_{S_1} > T_{S_2}$. In the second case, when late jobs are process in position one according to S_1 , moving a late job (the first job on some machine) to h^* results in the same completion time of the job (same tardiness for this specific job). The total tardiness of the jobs that remain on the machine from which the late job was removed remains the same (even after moving them forward) because none of them was late in the

first place. Hence $T_{S1} = T_{S2}$. The combination of both cases results in $T_{S1} \geq T_{S2}$.
q.e.d.

Note that the case when T_{S1} is the optimum among all schedules that employ the K_1 machines is just a particular case of Theorem 4, so this theorem also applies to optimal schedules. For some specific forms of the cost function $\phi(x)$, Theorems 1 to 4 above imply that, as k increases, TOC increases and T decreases. The latter is much more likely to occur in highly congested environments. If $T = 0$ for some number of machines, it will certainly do not decrease by adding one extra machine. In heavily congested problems, it is quite likely that T decreases each time an extra machine is added, hence, it is possible that a non-dominated schedule exists for each value of the number of machines used: $k = 1 \dots m$.

2.2 Problem Lower Bounds

Minimizing tardiness for a single machine is NP-Hard (Lawler (1977)), thus finding the optimal value of T , even for the case of a single outsource machine, and even without taking into account TOC , is not a trivial problem. Clearly, the problem discussed in this article is NP-Hard. As described in Section 2.1, the objective is to generate an efficient frontier set where there are tradeoffs between the total tardiness and the total outsource cost. For such a problem it may be possible to find large numbers of efficient solutions, depending on the number of jobs to be scheduled and the number of machines available. However, it is possible to relax or ignore some constraints to develop a limited set of theoretical lower bound solutions. The described theorems support the generation of this lower bound set, based on the following algorithm.

TOC LB algorithm for a given k (TOC-LB)

Step 1. Let $P' = P$, $N' = N$, $v = 1$ and $TOC = 0$.

Step 2. If $k > v$ then let g be the job with minimum processing time: $g = [j: \min_{j \in N'} p_j]$, solving ties arbitrarily, else go to Step 4.

Step 3. Remove g from N' and let $TOC = TOC + \phi(p_g)$, $P' = P' - p_g$, $v = v + 1$. Go to Step 2.

Step 4. Let $TOC = TOC + \phi(P')$.

Step 5. End.

Next we focus on the development of a tardiness lower bound for the problem, and then on the overall set of lower bound solutions. The dominance condition in Koulamas (1997) for the single machine tardiness problem is: 'a sequence where job i is always ahead of job f if $p_i \leq p_f$ and $d_i \leq d_f$ is optimal.' Thus, a lower bound for the single machine can be found by creating an "optimal" job lists that matches the SPT order of processing times with the EDD list of due dates, which is an optimal sequence for $1||T$. When parallel machines are considered (i.e., $m > 1$) as in Azizoglu and Kirca (1998), the assumption on non-divisible jobs, for the purpose of finding a bound, can be relaxed. If each job is then equally distributed among all machines, the time load on each machine is the same, so the single machine process just described can be applied to calculate a lower bound T . Let Sol_{lb} represent the set of non-dominated lower bound solutions.

LB algorithm

Step 1. Let $k = 1$ and $Sol_{lb} = \{\}$.

Step 2. Considering k machines determine the T lower bound as in Azizoglu and Kirca (1998) and the TOC lower bound by procedure *TOC-LB*.

- Step 3. Add lower bound solution to Sol_{lb} .
 Step 4. If $k < m$ and $T > 0$ then $k = k + 1$ and go to Step 2.
 Step 5. End.

3 Solution Approaches

The bi-criteria problem considered in this article has not been previously addressed. This section presents the first attempts at solving the problem. We combine currently available methods that solve the total tardiness problem with several “search strategies” in order to generate the tradeoff solutions.

3.1 Heuristics for the Total Tardiness Problem

Alidaee and Rosa (1997) conducted research that evaluates and compares the performance of several heuristics for traditional versions of the $1||T$ and $P||T$ problems. These researchers report that heuristic MDD outperformed all others under most experimental conditions. In their analysis, second to MDD came the TPI heuristic by Ho and Chang (1991). Details of these two heuristics appear in the corresponding publications. Based on the results of Alidaee and Rosa (1997), we use the TPI and MDD heuristics as sub-procedures in our heuristics.

- **TPI** sub-procedure: Load the jobs into the first available machine in ascending order of the traffic priority index z_j . The value of $z_j = d_j w_d / \max_{j \in N} d_j + p_j w_p / \max_{j \in N} p_j$, and let w_d be the weight of the EDD rule, $w_d = \max [\min [0.5 + (Z - TCR)/TCR, 1], 0]$, and w_p be the weight of the SPT rule, $w_p = 1 - w_d$. The variable TCR is the Traffic Congestion Ratio, $= n \sum_{j \in N} p_j / (m \sum_{j \in N} d_j)$, and Z is a constant, set to 3 as in Ho and Chang (1991).
- **MDD** sub-procedure: This approach separates the unscheduled jobs into two sets, those that will be early on a machine and those that will be late on a machine. The sub-procedure then selects either the job from the on time set that has the smallest flowtime, or the job from the late set with the smallest due date.

3.2 Problem heuristics

We propose three approaches to search for bi-criteria solutions. We assume that the TOC function $\phi(x)$ is such that, either $a > 0$ and $y = 1$, or $y < 1$. For the purpose of the heuristics, take into account the following comments. Let “apply TP ” represent either the employment of a TPI or MDD approach. This implies that each heuristic has two versions; one with TPI and the other with MDD . Note that the term $\Xi+$ used in the heuristics below indicates that the solution just generated is added to the solution set and then all dominated solutions are removed from this set. Let U_h represent the number of late jobs on machine h , and L_h the processing time load on machine h . Also, in the process of selecting jobs, all ties are decided by minimum due date. We present the three heuristics next.

Machine Increase Search Strategy (MI)

- Step 1. Let $k = 1$.
 Step 2. Apply TP . $\Xi+$. If $T = 0$, End; else, continue.
 Step 3. If $k = m$, End; else let $k = k + 1$, $K' = \{1, \dots, k - 1\}$.
 Step 4. Let f be the machine with the maximum number of late jobs ($\max [U_h, h \in K']$). Break ties selecting the machine with largest tardiness ($\max [T_h, h \in K']$). Let i be the shortest job ahead of all late jobs on machine f . Remove job i from machine f and add it to machine k .

- Step 5. Sequence machines f and k by TP . $\Xi+$. If $T = 0$, End; else, continue.
 Step 6. If $L_k < L^*$ and $\sum_{h \in K} U_h > 1$, go to Step 4; else continue.
 Step 7. Apply TP on k machines. $\Xi+$. Go to Step 3.

Machine by Machine Search Strategy (MM)

- Step 1. Let $k = 1$.
 Step 2. Apply TP . $\Xi+$. If $T = 0$, End; else, continue.
 Step 3. If $k = m$, End; else let $k = k + 1$, $K' = \{1, \dots, k - 1\}$, $L^* = P/k$.
 Step 4. Apply TP on the k machines. $\Xi+$. If $T = 0$, End; else, continue.
 Step 5. Let f be the machine with the maximum number of late jobs ($\max [U_h, h \in K]$) with ties solved by selecting the machine with largest tardiness ($\max [T_h, h \in K]$). Let w be the machine with the minimum number of late jobs ($\min [U_h, h \in K]$) with ties solved by selecting the machine with minimum tardiness ($\min [T_h, h \in K]$). Let i be the shortest job ahead of all late jobs on machine f . Remove job i from machine f and add to machine w .
 Step 6. Sequence machines f and w by TP . $\Xi+$. If the new solution is in the efficient set, go to Step 5. Else go to Step 3.

Machine Reduction Search Strategy (MR)

- Step 1. Let $k = m$. $K' = K$.
 Step 2. Apply TP on k machines. $\Xi+$.
 Step 3. If $k = 1$ then End, else continue.
 Step 4. Let f be the machine with the minimum load ($\min [L_h, h \in K']$), break ties arbitrarily. Let $K' = K' - f$.
 Step 5. Let i be the job from machine f with smallest due date. Remove job i from machine f .
 Step 6. Generate $k - 1$ temporary alternative schedules by adding job i to each machine in set K' and then sequencing by TP . Select the schedule with minimum T , break ties by minimum TOC . $\Xi+$.
 Step 7. If $n_f = 0$ then $k = k - 1$ and go to Step 2; else go to Step 5.

The combination of these three approaches with the two tardiness procedures results in six heuristic combinations: *MI-TPI*, *MI-MDD*, *MM-TPI*, *MM-MDD*, *MR-TPI*, and *MR-MDD*.

4 Preliminary Experiments and Results

4.1 Evaluation Method

A comprehensive method for the evaluation of Pareto efficient sets in scheduling problems was presented by Ruiz-Torres *et al.* (2006). This evaluation method combines several previously proposed evaluation methods for Pareto sets. The motivation behind combining several methods is that existing methods do not result in identical rankings of the generated Pareto sets and no method has been accepted as the most appropriate (Carlyle *et al.*, 2003). The evaluation method proposed in Ruiz-Torres *et al.* (2006) analyzes Pareto sets by utilizing three of the existing evaluation methods: DEV proposed by Ruiz-Torres and Barton (2001), the IPF (Integrated Preference Functional) method proposed by Carlyle *et al.* (2003), and the Free Disposal Hull (FDH) formulation, a special case of DEA. The DEV method measures the “deviation” from the solutions (schedules) in set Sol_{Heu} to the solutions in Sol_{LB} (from here the name DEV) using a normalized function. The IPF method employs a function to assess the quality of sets of near-Pareto-optimal solutions (schedules) by estimating the weighted

“expected” utility of each heuristic. The FDH-based method evaluates solutions in terms of the FDH (DEA) “degree of efficiency,” where the degree of efficiency of an inefficient solution (not necessarily the same as its FDH efficiency score) can be computed in terms of the “slacks or surpluses” of the inefficient solution in relation to a solution that dominates it. All evaluation scores are normalized, which results in measures with values between 0 and 1, where in DEV and IPF a lower score is preferred and in FDH a higher score is preferred. For the sake of brevity, the formulations and details about these three methods are not included here. Details are available in Ruiz-Torres *et al.* (2006) and in the original papers where these methods were proposed. For any particular problem instance and a set of heuristics, the three evaluation measures are calculated and then used to build the *normalized aggregate score*; = **Average** (DEV score, IPF score, 1 – FDH score), where clearly a small *aggregate score* indicates a high level of heuristic efficiency.

4.2 Experimental Setup

The experimental parameters are consistent with previous research in parallel machine tardiness problems (Ho and Chang, 1991, Alidee and Rosa, 1998). The processing times are randomly drawn from the discrete uniform interval $[1, 99]$. The maximum due date slack is determined by $s_{max} = (99/2) n / (m CR)$ where m , n , and CR (the congestion ratio) are experimental variables. Higher values of CR indicate smaller due date values, which tend to increase total tardiness. Jobs due dates are the sum of the job’s processing time and a random slack drawn from the discrete uniform interval $[1, s_{max}]$. The levels considered for these experimental variables are m : 5 and 10; n = 75 and 150; and CR = 2 and 4. The fourth and final experimental parameter is the *TOC function* called factor F . We consider two “levels” for factor F . In level 1 (*non-linear*) we assume $a = 0$, $b = 1$ and $y = 1/2$, while in level 2 (*linear*) we assume $a = 100$, $b = 1$, and $y = 1$. Thus outsource costs are based on the load on the machines in level 1, while outsource costs are based on the number of machines used in level 2. A problem instance is a problem in which all experimental variables are fixed at some specific level. The number of replications per problem instance in our simulations was 10.

4.3 Preliminary Results

Table 1 presents the results corresponding to our simulations (scores in bold indicate the best performer). ANOVA results (not presented for the sake of brevity) demonstrated that the most important factor is the heuristic selected. The number of jobs (n) is the only other main effect of statistical significance. All two-factor interactions including the heuristic factor were significant, but this only indicates that the heuristic was the key factor that affects the *aggregate scores*. In order to find if there is a significant difference between heuristics, we used Tukey’s test. This test determined that there is no significant difference between the two dominating heuristics *MM-MDD* and *MR-MDD*, but that there is a significant difference between these two and the remaining four heuristics. Tukey’s test also showed that there is a significant difference between using the *TPI* or the *MDD* subprocedures (*MI-TPI* was outperformed by *MI-MDD*; *MM-TPI* was outperformed by *MM-MDD*; and *MR-TPI* was outperformed by *MR-MDD*). Heuristics *MM-MDD* and *MR-MDD* are the only heuristics that dominated the problem instances, as can be seen in Table 1. While Tukey’s test found no difference over the complete experiment set, *t-test* demonstrated that in 6 out of 16 problem instances there is a significant difference between heuristics *MM-MDD* and *MR-MDD*.

Table 1. *Aggregate Score Results.*

f	n	m	CR	$MI-TPI$	$MI-MDD$	$MM-TPI$	$MM-MDD$	$MR-TPI$	$MR-MDD$
nl	75	5	2	0.659	0.455	0.775	0.289	0.761	0.246
			4	0.842	0.657	0.572	0.222	0.890	0.349
		10	2	0.658	0.603	0.747	0.310	0.738	0.209
			4	0.890	0.758	0.557	0.251	0.697	0.193
	150	5	2	0.661	0.534	0.689	0.299	0.845	0.310
			4	0.867	0.720	0.572	0.123	0.942	0.419
		10	2	0.669	0.641	0.765	0.309	0.787	0.217
			4	0.871	0.678	0.589	0.268	0.770	0.243
l	75	5	2	0.854	0.495	0.788	0.172	0.843	0.171
			4	0.938	0.432	0.929	0.067	0.917	0.067
		10	2	0.930	0.470	0.908	0.134	0.908	0.127
			4	0.850	0.497	0.741	0.265	0.689	0.258
	150	5	2	0.967	0.598	0.952	0.035	0.950	0.033
			4	1.000	0.534	0.997	0.002	0.991	0.000
		10	2	0.978	0.464	0.972	0.034	0.971	0.033
			4	1.000	0.524	0.980	0.034	0.970	0.026

5 Conclusions and Future Work

In recent years, manufacturing outsourcing has become more widespread. However, scheduling research has addressed few scheduling problems related to outsourcing environments. This article proposes an outsourcing scheduling problem which considers tradeoffs between outsourcing cost and average tardiness. Both outsourcing cost and average tardiness criteria are important, since the former has a direct impact on the bottom line; while the latter represents a critical customer service measure. We present lower bounds and propose several heuristics to generate Pareto solutions for the proposed problem. Furthermore, we evaluate the heuristics via the lower bounds under a wide variety of experimental conditions. This article contributes to the literature in that it captures the cost of machine usage, i.e., determining the number of machines to employ and for how long, given that the time that a machine is used is a function of the machine cost.

Research work which deals with outsourcing and scheduling is relatively scarce but quickly emerging. Future research opportunities in this area are plentiful. Taking machines with different capacities (unequal machines) into consideration would certainly enrich the problem and widen its potential applications. Another interesting extension is to address the issue of allowing different importance or weight assigned to various jobs. Moreover, studying optimally solvable special cases should be a worthwhile research avenue. Finally, future research could extend the current two criteria problem to three or more criteria problem.

Acknowledgments

The Authors want to thank professor Piotr Wojciechowski from the Department of Mathematical Sciences at UTEP for his contributions to this research.

References

- Alidaee, B. and D. Rosa. 1997. "Scheduling parallel machines to minimize total weighted and unweighted tardiness." *Computers and Operations Research*, **24**: 775-788.
- Azizoglu, M. and O. Kirca. 1998. "Tardiness minimization on parallel machines." *International Journal of Production Economics*, **55**: 163-168.
- Cao, D., M. Chen, and G. Wan. 2005. "Parallel machine selection and job scheduling to minimize machine cost and job tardiness." *Computers and Operations Research*, **32**: 1995-2012.
- Carlyle, W.M., J.W. Fowler, E.S. Gel, and B. Kim. 2003. "Quantitative comparison of approximate solution sets for bi-criteria optimization problems." *Decision Sciences*, **34**: 63-82.
- Ho, J.C. and Y-L. Chang. 1991. "Heuristics for minimizing mean tardiness for m parallel machines." *Naval Research Logistics*, **38**: 367-381.
- Kim, B., 2003. "Dynamic outsourcing to contract manufacturers with different capabilities of reducing the supply cost." *International Journal of Production Economics*, **86**: 63-80.
- Koulamas, C., 1997. "Decomposition and hybrid simulated annealing heuristics for the parallel machine total tardiness problem." *Naval Research Logistics*, **44**, 109-125.
- Lawler, E.L. 1977. "Pseudopolynomial algorithm for sequencing jobs to minimize total tardiness." *Annals of Discrete Mathematics*, **1**, 331-342.
- Lee, Y.H., C.S. Jeong, and C. Moon. 2002. "Advanced planning and scheduling with outsourcing in manufacturing supply chain." *Computers and Industrial Engineering*, **43**: 351-374.
- Puich, M., and K. Walker. 2004. "Managing quality while outsourcing, *BioPharm International*, **17-9**: 60-61.
- Ruiz-Torres, A.J., J.C. Ho, and F.J. López. 2006. "Generating Pareto schedules with outsource and internal parallel resources." *International Journal of Production Economics*, **103**: 810-825.
- Ruiz-Torres, A.J. and R. Barton. 2001. "Assessment procedure for multiple Pareto solution sets." *6th International Conference of the Decision Sciences Institute*, Chihuahua, Mexico, 2001 Proceedings, CD, Paper 61.

Power and Dependency Barriers to Supplier Integration: A New Zealand Case Investigation

Tillmann Boehme*, Paul Childerhouse, James Corner, Ron Garland and Richard Varey
Waikato Management School, The University of Waikato, New Zealand

* Corresponding author: WMS, Private Bag 3105, Hamilton, NZ.
tb28@waikato.ac.nz

1 Introduction

Integration along the supply chain in order to improve performance and competitiveness is one of the main themes in supply chain management (Childerhouse & Towill, 2003; Frohlich & Westbrook, 2001; van der Vaart & van Donk, 2004). Frohlich and Westbrook (2001) showed in their study that organisations with the greatest level of external integration had the largest rates of performance improvements. However, supply chain integration is a major challenge to the operations management discipline, especially with external entities when taking the existence of power and dependency with suppliers and customers into account. The article focuses on how well New Zealand organisations are externally integrated with their key suppliers. First, prior studies in supplier relationships are presented. Next, the development of the research model is discussed, then a synopsis of the research method used in this study follows, finally research findings, and conclusions are presented.

2 Literature Review

2.1 Introduction

The structure of power in a buyer supplier relationship is likely to have a major impact on the ways in which an organisation is able to manage its suppliers in order to improve their performance (Sanderson, 2004). Therefore, many authors for example, (Bensaou, 1999; Caniels & Gelderman, 2005; Cox, 2001; Kraljic, 1983) tend to capture trust, power, and dependency when studying supplier buyer relationships. One possible but also popular way to capture those phenomena is in form of a 2x2 purchasing portfolio model (see Figure 1) that allows the researcher to analyse and cluster supplier relationships. This purchasing portfolio approach is examined in more detail.

2.2 Purchasing portfolio models

Purchasing portfolio models (see Figure 1) have received much attention in recent literature on strategic planning (Bensaou, 1999; Caniels & Gelderman, 2005; Cox, 2004; Kraljic, 1983). They can be used as analytical tools to organise information and create a classification framework of the variables included in the portfolio (Ellram, 1992). Kraljic (1983) developed one of the very first purchasing portfolio models to analyse the focal organisation's supplier relationships and other authors followed his approach (Bensaou, 1999; Caniels & Gelderman, 2005; Cox, 2004;

Gadde & Hakansson, 2001). Table 1 reviews different purchasing portfolio models including classification dimensions and the four resultant taxa.

Table 1: Review of Different Purchasing Portfolio Models Listed by Publication Date

Author	Classification Dimensions	Four taxa
Kraljic (1983)	Complexity of supply market Importance of purchase	Purchasing -, Material-, Sourcing-, and Supply Management
Bensaou (1999)	Supplier's specific investment Buyer's specific investment	Market Exchange, Captive- Buyer or Supplier, Strategic Partnership
Cox (2004)	Supplier- relative to buyer- power Buyer- relative to supplier- power	Independence; Buyer- or Supplier- Dominance, Interdependence
Caniels & Gelderman (2005)	Supply risk Profit impact	Non-Critical-, Leverage-, Bottleneck-, and Strategic- Items

Source: (Authors)

Recent adaptations and refinements of Kraljic's (1983) classification approach have led to alternative portfolio models using different classification dimensions. However, the fundamental assumption of all portfolio models seems to be the occurrence of differences in power and dependence between buyers and suppliers. The authors in Table 1 identified many dimensions but none of the approaches seem to cover all variables responsible for the existence of power.

2.3 Dependency Variables

This section discusses the existence of power and dependency in buyer supplier relationships. A distinction is drawn in the literature between buyer dependency and supplier dependency. Supplier dependency exists when the buying company is significant for the supplier and vice versa (Motwani, Larson, & Ahuja, 1998). Table 2 is a summary of different supplier dependency variables identified in the literature.

Table 2: Supplier Dependency Variables

Variable	Description of supplier dependency
Purchasing volume / Profit impact	Purchasing volume is the total value of products an organisation purchases from one source. Purchasing volume is the basis of buyer dominance (Cox, 2004; Olsen & Ellram, 1997). Kraljic (1983) instead focuses on the percentage a special product is responsible in terms of organisational profit.
Switching Cost	Some authors call switching cost level of specific investment (Bensaou, 1999; Monczka, Callahan, & Nichols Jr, 1995) however, if an organisation invested highly in a relationship then the switching cost are high and therefore the organisation is dependent (Gadde & Hakansson, 2001).
Real time customer demand information	To be efficient the manufacturer depends on EPOS data and information transparency. Having ownership of the data is a power source and makes the supplier depend on the buyer (Burt & Sparks, 2003).
No. of alternative available customers	In an oligopolistic market the number of alternative available customers is often limited. This increases the level of supplier dependency.

Source: (Authors)

On the supplier dependency side four key variables have been identified. Table 3 lists the variables responsible for buyer dependency.

Table 3: Buyer Dependency Variables

<i>Variable</i>	<i>Description of supplier dependency</i>
Capabilities / Supplier Skills	The supplier can have certain skills (Gadde & Hakansson, 2001). Those skills can be performance (Kraljic, 1983), or technical related. Technical complexity describes the equipment a supplier has to manufacture a product (Cox, 2001) as well as the skills to produce a special product or product component (Goffin, Lemke, & Szwejczewski, 2005).
Switching Cost	see Table 2
Resources available by supplier	Resource availability can be related to the final product, added value services, advertising support, and risk sharing (Gadde & Hakansson, 2001; Olsen & Ellram, 1997). Goffin et al. (2005) further identified the form of involvement in new product development as a resource a certain supplier offers.
Branding / Reputation	Branding is linked to the reputation a product/organisation has. If customers demand a special brand, the buying organisation can depend on its suppliers (Cox, 2001; Olsen & Ellram, 1997).
Number of alternative suppliers available	This variable describes the number of alternative available suppliers capable of delivering the same product. Olsen and Ellram (1997) look at product substitutability, however if the product is not substitutable then fewer alternative suppliers are available, hence the purchasing organisation highly depends on the supplier (Geyskens, Steenkamp, Scheer, & Kumar, 1996).
Agility	Innovative products have a higher frequency of design changes and shorter life cycles than functional products do. Also agility enhances communication between buyer and supplier and an increase in interaction (Bensaou, 1999; Harrison & van Hoek, 2005).

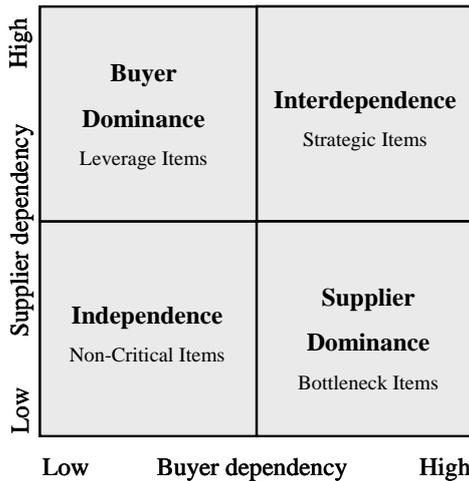
Source: (Authors)

The following section introduces the research model capable of capturing all of the above variables in one model.

2.4 Power & Dependency Dyadic Relationship Model

Cox's (2004) and Kraljic's (1983) portfolio approaches have a major influence on the developed strategic supplier relationship model in Figure 1. Cox (2004) uses broad classification dimensions to cover different power and dependency variables (see Table 1) however his approach is incomplete by focusing on five different variables only. Kraljic (1983) instead focuses on the product criticality.

Figure 1: Power & Dependency Dyadic Relationship Model



Source: Adapted from Cox (2004); Kraljic (1983)

Figure 1 shows the four supplier dependency variables identified in Table 2 arrayed on the y-axis and the six buyer dependency variables (of Table 3) are reflected on the x-axis. The combination of dimensions allows the type of relationship to be defined. Thus, if both supplier dependency and buyer dependency are limited, it is advisable to maintain independence, however if the supplier highly depends on the buying organisation and vice versa interdependency is existent. If supplier dependency is high and a low level of buyer dependency is present the buying organisation is dominating the relationship. If buyer dependency increases and a low supplier dependency exists, certainly the supplying organisation is dominating the relationship. Therefore, organisations have to develop sourcing relationships that are appropriate given the power and dependency circumstances in which they find themselves (Cox, 2004). Some key behaviour patterns are identified in Figure 2.

Figure 2: Characteristics of the four dyadic relationship types



Source: (Authors)

The independence category includes relationships that are straight forward to manage and indicate a low strategic importance. The key considerations when managing these purchases are standardisation and consolidation. The organisation should reduce the number of suppliers and the number of duplicate products if the products have been identified as non-critical (Olsen & Ellram, 1997). In a buyer dominance situation, the buying organisation appropriates most of the commercial value and sets price and quality trade-offs (Handfield, 2004). Independence as well as buyer dominance relationships have a strong price focus since those products are ideally non-critical. Procurement staff therefore requires good negotiating skills.

The focus of most interdependence relationships is achieving the simultaneous objectives of continuous improvements that lead to improved market share and better profit margins (Buzzell & Ortmeyer, 1995). However, interdependence relationships are very resource intense and therefore not applicable to every relationship/organisation (Das, 2005). An effort to shift the relationship is required when cornered in a supplier dominance situation. Supplier dominance and interdependence relationships require proactive employees with relationship management skills.

3 Methodology

Many researchers (Frankel, Naslund, & Bolumole, 2005; Mentzer & Kahn, 1995; New & Payne, 1995) conclude that supply chain management problems are often unstructured, even messy, real-world problems. They suggest that to gain relevance for supply chain researchers, “a one paradigm, one approach” perspective should not automatically be the obvious choice (Frankel et al., 2005). Therefore, two different qualitative methods have been applied namely Quick Scan Audit Methodology (QSAM) and case study research. The QSAM approach is a robust supply chain diagnostic tool to identify the change management opportunities in the supply chain and more detail on the Quick Scan approach can be found in (Naim, Childerhouse, Disney, & Towill, 2002). A Quick Scan was conducted in February 2006 in a major New Zealand forestry organisation. One major change management opportunity identified was the level of external integration with its key suppliers; therefore a follow up in-depth case study was undertaken in April 2006. This method then allowed for more flexible data collection. Case study research is increasingly used as a research tool (Yin, 1994) especially for research into social phenomena like relationships (Checkland & Holwell, 1998). Case study research is, like supply chain management, more process orientated and supports the researcher to understand why certain characteristics or effects occur (Meredith, 1998). Table 4 provides the outline of the research with a brief description of the applied data collection method.

Table 4: Outline of the research

<i>Method</i>	<i>Step</i>	<i>Data Collection Method</i>
QS Scoping	1) QSAM	A team based supply chain audit with quantitative and qualitative elements to identify the level of supply chain integration.
Case study in-depth investigation	2) AS-IS supplier relationship management	Semi-structured interviews to identify the current buyer supplier relationship supported by performance data.
	3) Identification of key dependency variables	Structured interviews to identify the most relevant dependency variables (see Tables 2 and 3) related to the particular case study.

4) Evaluation of supplier base	Sample of 30 suppliers from the supplier base picked by experts from the focal organisation. Identification of supplier and buyer dependency scores for each supplier.
5) Identification of improvement opportunities	Feedback presentation and an expert discussion amongst people that have been involved in the research.

Source: (Authors)

The research applied Table 4's five step model with predominantly qualitative tools. The following section provides the findings of every step together with background information about the forestry organisation.

4 Findings

4.1 Background

The supplier relationships between a New Zealand manufacturer of forestry products, referred to as WoodOrg, and the supplying organisations have been undertaken. The identities of the focal organisation and its suppliers/customers have been changed for proprietary reasons. The case study is an offshoot of a research investigation into supplier relationships/external integration predominantly in New Zealand's agricultural sector. The case description has been induced principally from interviews with key informants at WoodOrg and key informants of their third party maintenance supplier who is also involved in managing WoodOrg's supplier relationships. Eleven people have been interviewed in total including WoodOrg's and Maintenance's supply chain and procurement managers together with further purchasing and contracting staff.

WoodOrg is a wholly owned subsidiary of ParentCorp, and is one of New Zealand's larger manufacturers. WoodOrg produces a broad range of forestry products at several manufacturing sites in New Zealand and Australia, with nearly 60 percent of revenue earned in overseas markets. Currently ParentCorp and WoodOrg are undergoing a major restructuring process. One outcome is that WoodOrg outsourced the maintenance function to a global operating third party maintenance supplier in 2003. In 2004 WoodOrg implemented a supply chain management function to better control its production processes.

4.2 Findings in step 1 and 2: QSAM and 'AS-IS' supplier relationship management

The Quick Scan findings in this section are related to supplier relationship and external integration only. The Quick Scan team identified a total supplier base of 1607 suppliers where 80% of the total supplier base had an annual volume between 1.000 and 10.000 NZD. Nearly all of the suppliers are New Zealand based. WoodOrg and MaintenanceOrg employ four people responsible for the administrative side of procurement. Further, three people are responsible for managing the relationships with key suppliers.

Currently WoodOrg is measuring the performance of only twenty suppliers. These twenty suppliers were identified as key suppliers based on volume only. A lack of internal resources and capabilities to manage more relationships have been identified therefore, strategic procurement is very limited. Much procurement knowledge is tacit as procurement processes are not mapped and supplier information seldom shared and/or stored. In some cases the choice of a specific supplier was neither logical nor clear. Multiple unstructured and non transparent interfaces between suppliers and WoodOrg were identified. At this point, two key

suppliers delivering highly critical items (wood and energy) are presented in more depth. WoodOrg procurement personnel feel they continuously struggle with both suppliers via their dependent relationship yet they do not fully understand all nuances of the power/dependency situation. Relationship managers especially identified the energy supplier as very critical due to the fact that only two supply options are available, however only one supplier has the resources to deliver the amount required. The situation is similar on the wood supply side. Most of their wood suppliers are surprisingly part of ParentCorp, however they run their business as independent business units and therefore strive for the highest market price rather than “in house” cooperation. The wood and energy suppliers have been further evaluated in section 4.5.

4.4 Findings in step 3: Identification of key variables

The identification of the key variables is a crucial step since these variables are the basis for the evaluation of the supplier base and future action plans. In total six people were interviewed including supply chain managers, procurement manager, contract manager and procurement staff. The interviewees clearly identified six variables. On the supplier dependency side these are purchasing volume, level of specific investment, and number of alternative customers. On the buyer dependency side the key variables are capabilities and supplier skills, resources available by supplier, and number of alternative available suppliers. In the next step, the evaluation process, these variables were applied to each individual supplier.

4.5 Findings in step 4: ‘Should be’ supplier relationship

Both supply chain managers identified thirty suppliers to be evaluated. A 4-point Lickert scale was applied to each variable to measure the trend for each variable. The final score is based on a weighted average for buyer and supplier dependency. Each supplier was evaluated by the responsible procurement staff and by the supply chain manager, who also provided the weights for each variable, in order to achieve a more accurate evaluation of the relationship. Table 5 provides the final score that the interviewees agreed on.

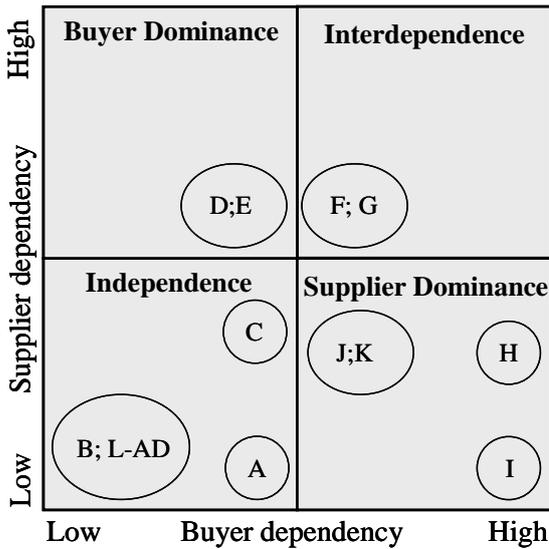
Table 6: Supplier evaluation scores

ID	A	B	C	D	E	F	G	H	I	J	K	L-AD
Supplier Dependency	1	1	2	3	3	3	3	4	4	2	2	1
Buyer Dependency	2	1	2	2	2	3	3	2	1	3	3	1

Source: (Authors)

It is not surprising that the Wood (ID H) and the Energy (ID I) suppliers scored highest on the supplier dependency side. Also none of the suppliers’ scored highest in both dimensions, buyer dependency and supplier dependency, therefore no strong interdependence between WoodOrg and their suppliers was identified. Figure 3 graphically displays the scores calculated in Table 6. All scores of one and two are assigned to low dependency whereas the scores of three and four were assigned to high dependency.

Figure 3: WoodOrgs' 'Should-Be' supplier relationship management



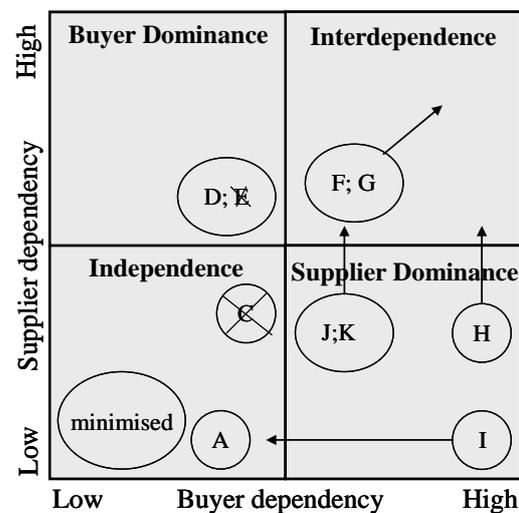
Source (Authors)

The power and dependency structure analysis summarised in Figure 3 highlights a mismatch of the AS-IS (step 1 and 2) and the Should-Be situation. Currently WoodOrg procures predominantly on a transactional/independence basis. The research identifies however that eight out of thirty suppliers require a more strategic approach.

4.6 Findings in step 5: Identification of improvement opportunities

Part of the research was a feedback presentation including a one hour discussion amongst all participants. Supplier dominance is a worry to WoodOrg especially because the products produced by these suppliers are highly critical. Further, the supplier base needs to be reduced dramatically. The long term target is 350-400 suppliers meaning a total reduction of 1200 suppliers. Figure 4 highlights improvement opportunities to achieve a “healthy” supplier relationship portfolio.

Figure 4: WoodOrgs' long-term strategic plan



Source: (Authors)

At the beginning is the reduction of the supplier base. The possible solution discussed is that WoodOrg needs to fully evaluate their supplier base to identify

their key strategic suppliers in order to move many of the independence suppliers to the second tier. This step can be clearly seen in Figure 4 where ID F is the key supplier and ID E and ID C have been removed from the portfolio into the second tier. Secondly the supplier dominance needs to be reduced. WoodOrg agreed that the Wood suppliers (ID H) and 3rd party logistics supplier (ID J) need more attention and a more strategic relationship emphasis. Shifting these suppliers to a more strategic and interdependence level remains questionable at this stage. However, the team agreed that by providing ID K with more volume the supplier dependency on WoodOrg will increase; hence the relationship shifts towards interdependence. The situation with the electricity supplier (ID I) is slightly different. WoodOrg has started reducing their dependency on this particular supplier by insourcing, as WoodOrg is generating their own energy by burning scrap gained through their production process. The generation of their own energy will be extended over the next couple of years to a level that more sourcing opportunities become available and therefore WoodOrg becomes more independent.

5 Conclusion

The research identified that the focal organisation is weakly integrated with their suppliers. Often power and dependency limit the level of integration. The focal organisation is highly dependent on some of its key suppliers including wood and energy. This buyer dependency situation has been identified as one of the key barriers for external integration. Ways have been identified to overcome this situation. The 2x2 relationship model has been identified as a highly valuable decision making tool. However, the study focused only on one main manufacturer in New Zealand. The question remains how other manufacturers or other members in the supply chain are externally integrated. Are other organisations highly dominated by their suppliers? Further research in New Zealand is needed, both to generalise the purchasing portfolio model developed here, and to comment more generally on power and dependency between organisations and their suppliers.

6. Acknowledgement

Thanks the AGMART Research team/fund for supporting this study. Our gratitude to Quick Scan Audit Team from Waikato University especially Dr Eric Deakins, Dr Peter Sun and Oliver Ma.

List of References

- Bensaou, M. (1999). Portfolios of buyer-supplier relationships. *Sloan Management Review*, 40(4), 35-44.
- Burt, S., & Sparks, L. (2003). Power and competition in the UK retail grocery market. *British Journal of Management*, 14, 237-254.
- Buzzell, R., & Ortmeyer, G. (1995). Channel partnerships streamline distribution. *Sloan Management Review*, Spring, 85-96.
- Caniels, M. C. J., & Gelderman, C. J. (2005). Power and independence in buyer supplier relationships: A purchasing portfolio approach. *Industrial Marketing Management*, Article in press, 1-11.
- Checkland, P., & Holwell, S. (1998). Action research: Its nature and validity. *Systemic Practice and Action Research*, 11(1), 9-21.
- Childerhouse, P., & Towill, D. R. (2003). Simplified material flow holds the key to supply chain integration. *Omega The international Journal of Management Science*, 31, 17-27.
- Cox, A. (2001). *Supply chains, markets and power: Mapping buyer and supplier power regimes*. London; New York: Routledge.
- Cox, A. (2004). The art of the possible: Relationship management in power regimes and supply chains. *Supply Chain Management: An International Journal*, 9(5), 346-356.
- Das, T. K. (2005). Deceitful behaviors of alliance partners: Potential and prevention. *Management Decision*, 43(5), 706-719.
- Ellram, L. (1992). International purchasing alliances: An empirical study. *International Journal of Logistic Management*, 2(1), 83-95.
- Frankel, R., Naslund, D., & Bolumole, Y. (2005). The "white space" of logistics research: A look at the role of methods usage. *Journal of Business Logistics*, 26(2), 185-209.
- Frohlich, M., & Westbrook, R. (2001). Arcs of integration: An international study of supply chain strategies. *Journal of Operations Management*, 19, 185-200.
- Gadde, L., & Hakansson, H. (2001). *Supply chain strategies*. Chichester, UK: John Wiley & Sons Ltd.
- Geyskens, I., Steenkamp, J.-B. E. M., Scheer, L. K., & Kumar, N. (1996). The effects of trust and interdependence on relationship commitment: A trans-Atlantic study. *International Journal Of Research in Marketing*, 13, 303-317.
- Goffin, K., Lemke, F., & Szejczewski, M. (2005). An exploratory study of 'close' supplier-manufacturer relationships. *Journal of Operations Management*, Article in Press, 1-21.
- Handfield, R. B. (2004). Trust, power, dependence, and economics: Can SCM research borrow paradigms? *International Journal Integrated Supply Management*, 1(1), 3-25.
- Harrison, A., & van Hoek, R. I. (2005). *Logistics management and strategy: (2nd ed.)*. New York: Financial Times Prentice Hall.
- Kraljic, P. (1983). Purchasing must become supply management. *Harvard Business Review*, 61(5), 109-117.
- Mentzer, J. T., & Kahn, K. B. (1995). A framework of logistic research. *Journal of Business Logistics*, 16(1), 231-250.
- Meredith, J. (1998). Building operations management theory through case and field research. *Journal of Operations Management*, 16, 441-454.
- Monczka, R. M., Callahan, T. J., & Nichols Jr, E. L. (1995). Predictors of relationships among buying and supplying firms. *International Journal of Physical Distribution & Logistics Management*, 25(10), 45-59.
- Motwani, J., Larson, L., & Ahuja, S. (1998). Managing a global supply chain partnership. *Logistics Information Management*, 11(6), 349-354.
- Naim, M. M., Childerhouse, P., Disney, S., & Towill, D. (2002). A supply chain diagnostic methodology: Determining the vector of change. *Computers & Industrial Engineering*, 43, 135-157.
- New, S. J., & Payne, P. (1995). Research frameworks in logistics: Three models, seven dinners and a survey. *International Journal of Physical Distribution & Logistics Management*, 25(10), 60-77.
- Olsen, R. F., & Ellram, L. M. (1997). A portfolio approach to supplier relationship management. *Industrial Marketing Management*, 26, 101-113.
- Sanderson, J. (2004). Opportunity and constraint in business-to-business relationships: Insights from strategic choice and zones of manoeuvre. *Supply Chain Management: An International Journal*, 9(5), 392-401.
- van der Vaart, T., & van Donk, D. P. (2004). Buyer focus: Evaluation of a new concept for supply chain integration. *International Journal of Production Economics*, 92, 21-30.
- Yin, R. (1994). *Case study research: Design and methods* (Vol. 5). London: Sage Publications.

A New Aggregation-Based Method with Improved Processing Selectivity for the Open Pit Mine Production Scheduling Problem

Natashia Boland, Irina Dumitrescu[†], Gary Froyland

[†]School of Mathematics

The University of New South Wales

Sydney, NSW 2052

Australia

irina.dumitrescu@gmail.com

Abstract

In our talk we consider an orebody that contains only one base metal of interest, which is exploited using open pit mining methods. The processing of the base metal requires separation of the ore from waste. The orebody is represented as a discretisation of the volume of earth into blocks. In the presence of mining and processing costs, upper limits on mining and processing capacities and precedence constraints with respect to the order in which the blocks can be excavated, the Open Pit Mining Production Scheduling Problem (OPMPSP) consists of finding the sequence in which the blocks should be removed from the pit, over the lifetime of the mine, such that the net present value of the operation is maximised.

Due to the large number of blocks and of precedence constraints linking them, aggregation of blocks is used. We propose an approach to solving the OPMPSP which uses the aggregates to schedule the mining process, but allows the excavated ore to be processed differently within aggregates. In this model, the maximum processing selectivity is reached when each block inside an aggregate can be processed independently of the rest of the blocks that belong to the same aggregate. In other words, the processing decisions are made at the level of blocks. However, this type of selectivity may complicate the problem too much and make it difficult to solve.

We look at several ways of creating groups of blocks inside any aggregate, so that processing decisions are made at the level of groups. The methods we propose can be proved to attain the same degree of selectivity as the block level processing. We will also propose a heuristic method which tries to maximise selectivity while keeping all aggregates split into at most two groups of blocks. We will provide numerical results for the algorithms proposed.

Sportsbet 21: a successful application of statistical modelling

Stephen R Clarke
Faculty of Life and Social Sciences
Swinburne University of Technology
Australia
sclarke@swin.edu.au

Abstract

Universities are under increasing pressure to increase funding from non government sources. Swinburne University is no exception, and has as one of its strategic themes to be an entrepreneurial university and to encourage innovation and commercialisation of research results. As part of the then School of Mathematical Sciences, Swinburne Sports Statistics was an early entrant into the world of consulting. Computer predictions for AFL football were sold to the media as early as 1980. The expansion of sports betting from the traditional horse, harness and dog racing into a wide range of sporting events has created a market for sports prediction expertise. This paper discusses the development of Sportsbet 21, a Swinburne startup company which provides to bookmakers computer generated odds driven by a statistical model for 'betting in running'- betting on events which occur within a sporting contest. Models have been developed for both cricket and tennis, and the product has been operated successfully by Ladbrokes in the UK for some time. The product has achieved profit targets on growing turnover, and demonstrated the robustness of the mathematical models

Key words: Statistical modelling, Gaming, football, cricket.

1 Introduction

As part of the then School of Mathematical Sciences, one of Swinburne Sports Statistics areas of research was in the prediction of sporting events. This paper details some of our efforts at commercialisation of this activity, in particular the creation of a startup company Sportsbet 21.

2 AFL Football

Swinburne's forays into sport prediction began in 1980, when I wrote a computer program to predict Australian Rules football. A Melbourne daily newspaper, The Sun, agreed to publish the weekly predictions in competition with their chief writer. A consultancy agreement was signed, one of the first outside consultancies undertaken by the Mathematics department, which at that time was mainly a teaching school. Since then, the tips have been published in various media outlets. Print outlets include The Sun from 1981 to 86, The Age from 1990 to 95, The Herald Sun in 1986 and The Australian Financial review in 1987 to 99. The tips made their television debut in 1995, and were broadcast each Thursday on Today/Tonight Adelaide from 1995 to 2002. All these contracts were for a fixed weekly fee for the provision of the tips. From 1997 the

tips have also been published late in the week on our web site www.swin.edu.au/sport, and some other media outlets have picked the predictions up unofficially. The program has spawned many imitations and has led to increased interest in automatic prediction and sport ratings. Clarke (1988, 1993) contains some details of the program and experiences in publishing in the daily press.

Although in the main newspapers only published predicted winners, the program predicts not only the winner of an upcoming match, but also the expected margin in points, the chance of each team winning, and via a simulation of the remainder of the year, the chance of each team finishing the home and away series in any position. With the growth in sports betting this information has become of interest to punters. Right from the beginning, I had envisaged the possibility of marketing the predictions direct to interested parties via a mail subscription, but lacked the marketing skills to implement. With the financial policies in place at the time, the main motivation for marketing the tips was the publicity the school and the University received.

The program was ahead of its time in publishing margins and chances of winning long before sports betting on these outcomes were legal. Betting is now allowed on Australian rules football. In addition to head to head betting on the winner, various bets can be made on the margin of victory and of course on teams making the finals or winning the premiership. In 2002, following an approach from Ozmium Ltd the program was marketed to punters. Ozmium were already marketing horse racing tips through their web site www.smartgambler.com.au and were keen to diversify. The tips are emailed to the list of subscribers each week, along with a spreadsheet showing bookmakers prices, overlays, recommended bets and bet sizes according to the Kelly formula. The punters placed their own bets, based on the programs recommendations, their own information and betting strategies. A discussion list was set up, and subscribers discussed the tips, their own systems, the pros and cons of various bookmakers, good deals etc. In line with my suggestions, it was quite clear that most subscribers were not using the computer's predictions as a black box, but as a starting point to which they added their own opinions. It was also clear that they all had their own betting systems. While some bet on all predictions which showed some set percentage overlay, others required a certain probability of winning etc. Clarke & Clarke (2006) contains an analysis of the computer's performance over several years. As the detailed predictions are now marketed to punters, media outlets are provided with winners only for publication in return for citing Swinburne. In 2006 outlets included ABC Adelaide radio, several local papers and Campus Review.

In addition to AFL football, over the years Swinburne has predicted many sports and placed predictions on their web site. Sports have included cricket, tennis, Grand Prix motor and cycle racing, horse racing, netball, rugby, rugby league, baseball, basketball, beach volleyball, Olympic games and soccer. With Governments now allowing betting on a range of sports, it became clear our expertise in prediction might be put to profitable use in the gaming industry

3 Sportsbet 21 and Cricket

Early in 1998 Mark Solonsch of Synaval Pty Ltd approached me with an idea of developing an automated betting system to allow real time bets on events that occur during a sporting event. The initial idea was to bet on the number of runs in an over of cricket. The system would be driven by computer-generated odds that would take into account the state of the game and the past betting pattern of the punters.

There are several problems with sports betting. Consider one-day cricket. With approximately 100 international matches per year, betting opportunities are limited. This contrasts with horse and harness racing where there might be that many races in one day in Australia. A bet takes a day (or 5 days in test cricket) to decide, and this reduces churn, the opportunity for punters to 'reinvest' their winnings, reducing turnover for the bookmaker. Furthermore, with head to head betting there are only two outcomes, and this restricts the bookmaker to a margin of about 5%. A small number of bets and a small margin both decreases profit and increases the risk for the bookmaker. On the other hand, betting on the number of runs in an over of cricket provides 100 bet opportunities in one match, and with over 10 outcomes allows higher betting margins than head to head. Difficulties to be overcome included the small betting time between overs to set odds and take and validate bets. This necessitated the development and testing of a suitable statistical model that would set the odds virtually instantly, and suitable computer protocols to allow a large number of bets in a small time interval. Statistical modelling was undertaken by Michael Bailey and myself, with Myles Harding and Geoff Lewis developing the computer implementation. Most of this computer and statistical development work was made possible by appointments and grants through the Chancellor's strategic funds.

The statistical model was developed by analyzing data from all one day matches played up to that time. Logistic and linear regression methods were used to estimate the probability of each of 11 outcomes, being the number of runs from 1 to 10 and more than 10 runs. Variables used included scores, run rates and wickets fallen over various periods, over number and target score. Obviously separate models had to be developed for first and second innings. Bailey & Clarke (2004, 2006) give examples of the sort of statistical modeling undertaken, in these cases to predict the scores of individual batsmen and the chances of each team ultimately winning the game. Once true probabilities were predicted, these were converted to bookmakers odds, allowing for any given percentage profit. As soon as the operator entered the score for the over, the model virtually instantly calculated the prices for each of the 11 outcomes for the next over. This allowed punters approximately 30 seconds to place bets before betting was closed at the start of the next over.

In early 2000 a trial was run with volunteer punters given a certain amount of play money to bet on a one-day match. The results were positive in that they achieved a return to the operators and positive feedback from players. The startup company Sportsbet 21 Pty Ltd was set up with Swinburne and staff holding 50%, and Synaval Pty. Ltd. 50%. The system underwent several further trials, and a License agreement was finally struck with Ladbrokes, which allows for Sportsbet 21 to receive a proportion of net profits generated when Ladbrokes offered the system to their punters. The system was run live with Ladbrokes under the name 'between overs' during the 2003 World Cup. The product achieved profit targets on small turnover, and demonstrated the robustness of the mathematical models. Subsequently Tasmania Tote bought 25% of the company, which provided funds for further development work. The product has continued to operate under the name 'betting in running'. Figure 1 shows a screen dump. Turnover has grown, although the limited time allowed for betting had been a restricting factor.

Subsequent developments have allowed for setting prices an over in advance, so prices on over 51 (say) will be set and betting commence as soon as over 50 starts. Models have also been developed for test cricket, betting on partnership length and

20/20 cricket is on the drawing boards. Some work has also been done on making the models less generic and more player and situation specific.

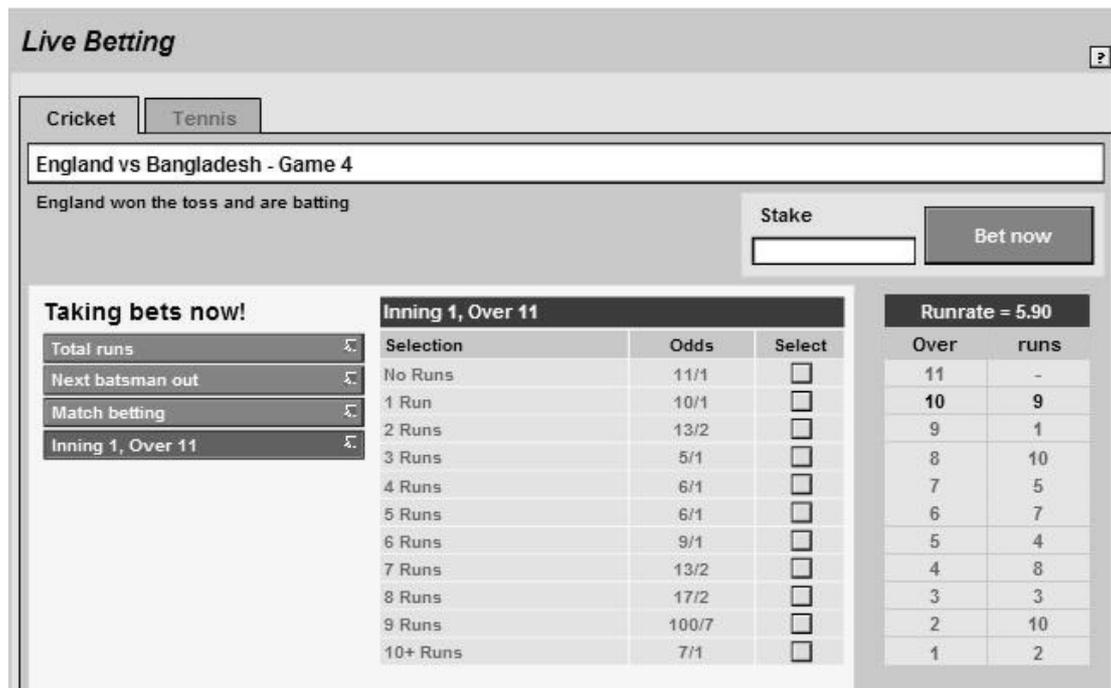


Figure 1. Early screen dump of betting in running on cricket

4 Tennis

With the success of the cricket model, it was decided in 2003 to extend into other sports, and tennis was chosen. We decided to allow betting on each game score. Thus the server could win to love, 15, 30, or deuce, or lose to love, 15, 30, or deuce, giving 8 outcomes and approximately 10 bets in each set of a tennis match. The approach taken differed significantly from that taken in cricket. Firstly, the model was player specific, and the past player statistics of the two participants was taken as input data. Barnett & Clarke (2005) discuss the player statistics available, and Tristan Barnett has provided regular updates of player statistics to the system. Secondly, once the relevant parameters were determined, a probability model was used to calculate the chances of all possible outcomes. The input parameters were updated as the match progressed. Since the prices for a server's next service game were determined as soon as they finished serving their current game, the system always allowed for betting a game ahead. This has allowed for betting to be virtually continually open. Figure 2 shows an early screen for tennis betting. Recently the ability to bet on the winner of the game has been added to the product.

5 Conclusion

Acceptance by the public of betting in running continues to grow, with the models generally standing up to punters' skills. The company recently gained a new licensee in Vanuatu, which should see turnover increase further. Clearly there is a need to develop further products, but funding development of new applications is a continuing difficulty

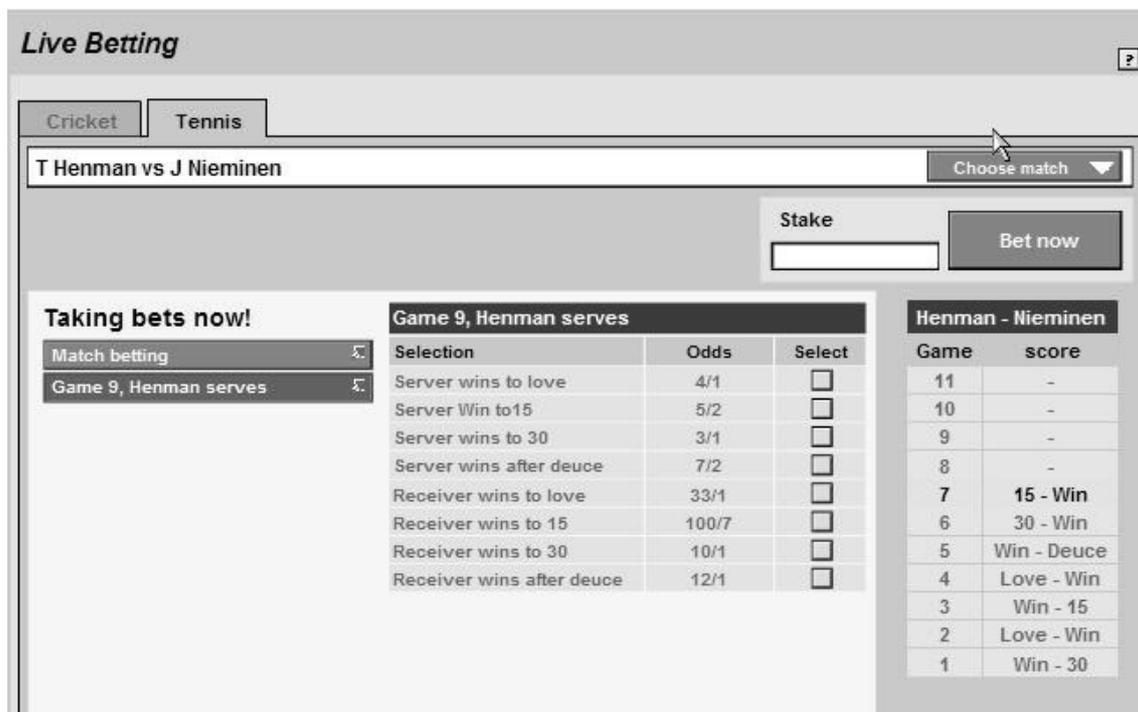


Figure 2. Early screen dump of betting in running on tennis

Acknowledgement

Thanks to Mark Lowy, Sportsbet 21 director, for reviewing the draft of this paper.

6 References

- Bailey, M. and S. R. Clarke, 2004. "Market Inefficiencies in Player Head to Head Betting on the 2003 Cricket World Cup". In *Economics, Management and Optimisation in Sports*, P. Pardalos, S. Butenko et al, (Eds.), Springer. pp. 185-201.
- Bailey, M. and S. R. Clarke, 2006. "Predicting the match outcome in One Day International cricket matches, while the game is in progress". In *Proceedings of the Eighth Australasian conference on Mathematics and Computers in Sport*, J. Hammond and N de Mestre, (Eds.), MathSport (ANZIAM): Coolongatta. pp 160-169.
- Barnett, T. J. and S. R. Clarke, 2005. "Combining player statistics to predict a long tennis match at the 2003 Australian Open". *International J. of Management Mathematics*. **16**: 113-120.
- Clarke, S. R., 1988. "Tinhead the tipster". *OR Insight*, **1**(1), 18-20.
- Clarke, S. R., 1993. "Computer forecasting of Australian Rules football for a daily newspaper". *Journal of the Operational Research Society*, **44**(8), 753-759.
- Clarke, S. R. and R. C. Clarke, 2006. "A Comparison of computer and human predictions of AFL". In *Proceedings of the Eighth Australasian conference on Mathematics and Computers in Sport*, J. Hammond and N de Mestre, (Eds.), MathSport (ANZIAM): Coolongatta. pp 141-149.

The Travelling Tournament Problem: Neighbourhoods and Visualisation

Mark R. Johnston
School of Mathematics, Statistics and Computer Science
Victoria University of Wellington
New Zealand
Mark.Johnston@mcs.vuw.ac.nz

Abstract

The Travelling Tournament Problem (TTP) is a difficult combinatorial optimization problem. It involves constructing a double round robin sports tournament (every team plays every other team once at home and once away). Each team begins at its home city and travels to play its games at the chosen sequence of venues (some home and some away) and returns to its home city at the end of the tournament. The objective is to minimize the total distance travelled by the teams subject to a number of side constraints. We propose novel local search neighbourhoods, including some involving four rounds of the tournament. Visualisation of neighbourhoods and solutions leads to interesting insights into the geometric structure of good solutions found by local search.

1 Introduction

Suppose there is an even number of sports teams, say n teams. In a *round robin tournament* each team plays every other team exactly once (requiring $n - 1$ rounds). In a *double round robin tournament* each team plays every other team exactly twice, once at home and once away (requiring $2n - 2$ rounds). A simple example with $n = 6$ teams is given in Table 1. An alternative representation, often seen in the sports results section of a newspaper, is given in Table 2, in which the number in the body of the table is the round (or date) of the corresponding match. It is relatively easy to produce a double round robin tournament with no additional constraints using the polygon method (see, e.g., de Werra 1981).

The *Travelling Tournament Problem* (TTP) is to determine a double round robin tournament that minimizes the total distance travelled by all teams subject to a number of side constraints (Easton, Nemhauser, and Trick 2001). Each team begins at its home city and travels to play its games at the chosen sequence of venues (some home and some away) and returns to its home city at the end of the tournament. We assume that each team has a single home city and that the distance between home cities is known. The *location* (home or away) of each team in each round is the city

Table 1: Simple representation of a double round robin tournament.

		Round									
		1	2	3	4	5	6	7	8	9	10
	1v6	2v6	3v6	4v6	5v6	6v1	6v2	6v3	6v4	6v5	
	5v2	1v3	2v4	3v5	4v1	2v5	3v1	4v2	5v3	1v4	
	4v3	5v4	1v5	2v1	3v2	3v4	4v5	5v1	1v2	2v3	

Table 2: Newspaper representation of a double round robin tournament.

		Away team					
		1	2	3	4	5	6
	1	–	9	2	10	3	1
	2	4	–	10	3	6	2
Home	3	7	5	–	6	4	3
team	4	5	8	1	–	7	4
	5	8	1	9	2	–	5
	6	6	7	8	9	10	–

where team i plays its game in that round (i if it plays at home or its opponent's home if it plays away).

The principal side constraints are the “*atmost*” constraints (no more than three consecutive home or away games are allowed for any team) and the “*norepeat*” constraints (a game between team b at team a 's home cannot be followed immediately by a game between team a at team b 's home). An additional constraint considered by Ribeiro and Urrutia (2004) is the *mirror constraint* whereby a tournament is composed of two consecutive round robin tournaments which are identical apart from reversing the venues.

The TTP was introduced by Easton, Nemhauser, and Trick (2001, 2003) as an abstraction of Major League Baseball in the United States where teams play several games “on the road” before returning home. Trick (2006) maintains a webpage of challenging TTP problem instances. Knust (2006) maintains a useful collection of references on the TTP and related problems in sports tournament scheduling. Some very recent papers have appeared from the PATAT 2006 conference (Burke and Rudova 2006).

The Travelling Salesman Problem (TSP) provides a convenient lower bound on the optimal unconstrained TTP solution. The minimum possible distance travelled by any one team is to start at home, play a sequence of zero or more home games, followed by an optimal TSP tour of all the other team locations “on the road”, before returning home for a final sequence of zero or more home games. Thus if z_{TTP}^* is the optimal TTP total distance and z_{TSP}^* is the optimal TSP distance, then $z_{TTP}^* \geq nz_{TSP}^*$. An upper bound on z_{TTP}^* is given by 4 times the sum of all distances between home cities, corresponding to all teams playing alternate home and away matches.

The intention of this paper is to determine whether good (near-optimal) solutions to the unconstrained TTP exhibit near-TSP optimal individual routes, i.e., does each

Table 3: Travel sequence representation of a double round robin tournament

Team	Round																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1:	-9	-8	-7	-4	-5	-3	-2	-6	10	-10	9	8	5	7	6	3	2	4
2:	-4	-9	-8	-7	-6	-5	1	10	-3	9	8	5	7	4	3	6	-1	-10
3:	-8	-7	-6	-5	-4	1	10	9	2	8	5	7	4	6	-2	-1	-10	-9
4:	2	10	-5	1	3	8	-6	-7	9	5	7	6	-3	-2	-8	-10	-9	-1
5:	10	6	4	3	1	2	9	8	7	-4	-3	-2	-1	-10	-9	-8	-7	-6
6:	-7	-5	3	9	2	10	4	1	8	7	-10	-4	-9	-3	-1	-2	-8	5
7:	6	3	1	2	10	9	8	4	-5	-6	-4	-3	-2	-1	-10	-9	5	-8
8:	3	1	2	10	9	-4	-7	-5	-6	-3	-2	-1	-10	-9	4	5	6	7
9:	1	2	10	-6	-8	-7	-5	-3	-4	-2	-1	-10	6	8	5	7	4	3
10:	-5	-4	-9	-8	-7	-6	-3	-2	-1	1	6	9	8	5	7	4	3	2

team follow an efficient sequence of away games? In order to address this question, we consider a very simple version of the TTP with *no side constraints* and Euclidean distances, i.e., we abandon the atmost, norepeat and mirror constraints.

The remainder of this paper is structured as follows. Section 2 presents ways of visualising a tournament (solution). Section 3 defines a number of local search neighbourhoods that can be classified by how drastically they alter a given solution. Section 4 reports some preliminary computational experience. Finally Section 5 offers some conclusions and recommendations for future research.

2 Tournament Visualisation

Text Visualisation. A common text representation of a tournament schedule is illustrated in Table 3 (see, e.g., Anagnostopoulos et al. 2006). This representation highlights the travel sequence (round-by-round trajectory) of each team (row), with the number being the opponent in that round (a positive indicating the game is at home and a negative indicating the game is away).

Graphical Visualisation. The objective of the TTP is to minimize the total distance travelled by all teams. Figure 1 gives a graphical visualisation of the tournament from Table 3 where the teams are located at the vertices of a regular decagon and the distances between teams is Euclidean. The lower part of Figure 1 shows the movement of each individual team, with the team number shown in bold and the distance travelled by that team shown underneath. For example, Table 3 shows that team 1 begins at home, plays 9 – 8 – 7 – 4 – 5 – 3 – 2 – 6 in a sequence of away games, plays 10 at home, then 10 away, then back home for all the remaining games. The upper part of Figure 1 shows the sum of all the lower figures, where the thickness of an edge is proportional to the number of times that edge is traversed by any team. In this example, a good solution would include very thick edges around the convex hull (the TSP solution) and only a few interior edges as necessary to form a feasible double round robin tournament.

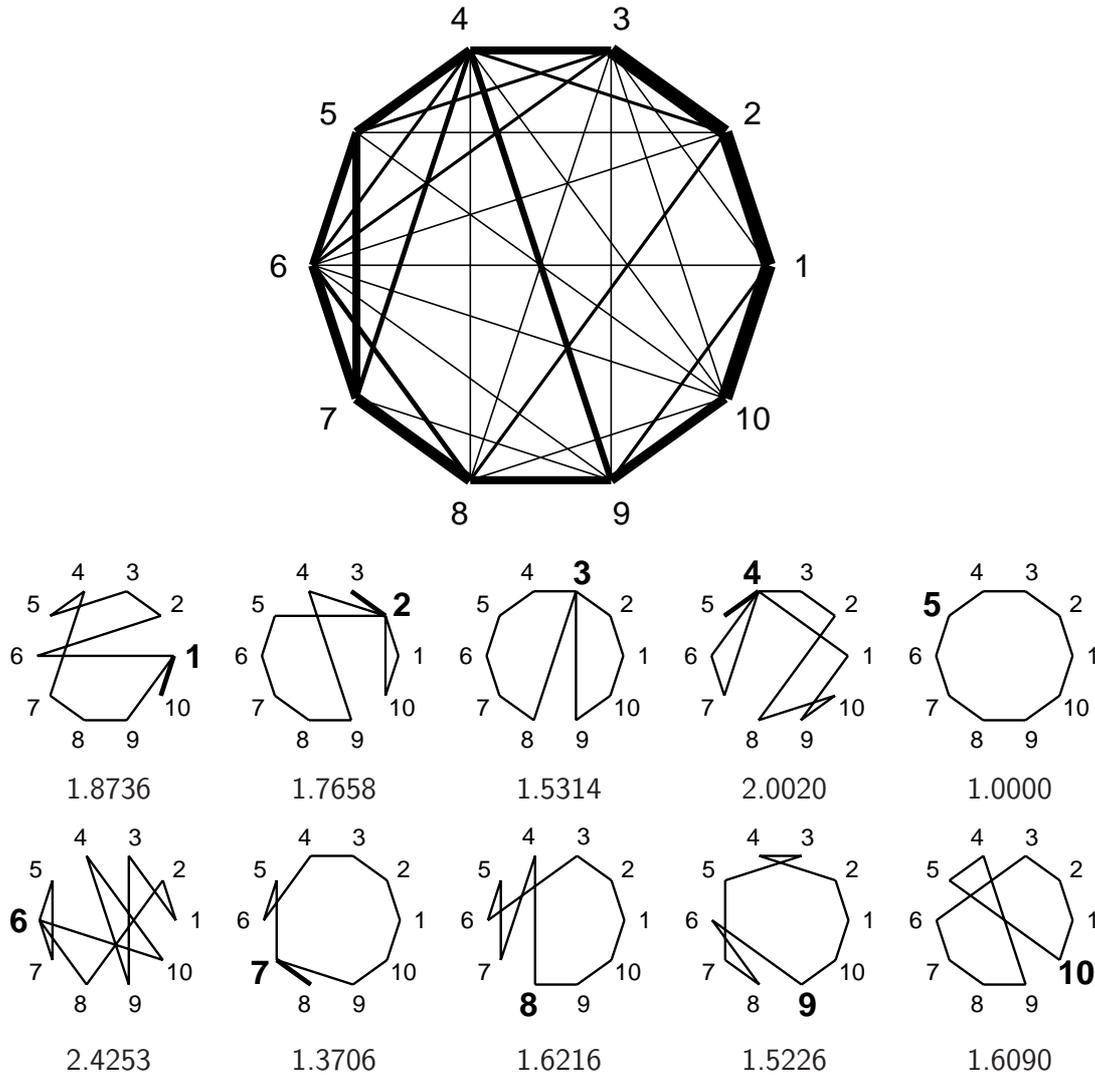


Figure 1: Visualisation of a tournament with $n = 10$ teams and $z_{TTP} = 16.7220$.

3 Local Search Neighbourhoods

The TTP (with atleast and norepeat constraints) is a very difficult combinatorial optimization problem. Easton, Nemhauser, and Trick (2001, 2003) approach its exact solution via constraint programming (CP) and integer programming (IP). However, due to the complexity of the TTP, many authors have sought approximate solutions via local search (Ribeiro and Urrutia 2004; Urrutia and Ribeiro 2006), simulated annealing (Anagnostopoulos et al. 2006; Lim, Rodrigues, and Zhang 2006), tabu search (Adriaen, Custers, and Berghe 2003; Di Gaspero and Schaerf 2006; Henz 2004) and ant colony optimisation (Adriaen, Custers, and Berghe 2003; Crauwels and Oudheusden 2003). Given a current solution \mathbf{x} , these local search methods (Aarts and Lenstra 1997) search a *local neighbourhood* of solutions $\mathcal{N}(\mathbf{x})$ that are in some sense close to \mathbf{x} . The authors above all suggest local neighbourhoods (although possibly under different names) summarised in Table 4. These are briefly defined below, and new neighbourhoods **GeneralTwoRound**, **SwapRoundTwoExchange** and **GeneralFourRound** are shown to generalise these neighbourhoods. We also classify the local neighbourhoods according to how drastically they alter a given solution and the

Table 4: Summary of neighbourhoods (those marked \diamond are new to this paper)

Level	Neighbourhood	Rounds involved	Teams involved	Evaluation complexity
1	SwapHomes	2 rounds	2 teams	$O(1)$
1	PartialSwapRounds	2 rounds	4 teams	$O(1)$
1	GeneralTwoRound \diamond	2 rounds	all teams	$O(n)$
2	SwapRounds	2 rounds	all teams	$O(n)$
2	SwapRoundTwoExchange \diamond	4 rounds	all teams	$O(n)$
3	PartialSwapTeams	4 rounds	all teams	$O(n)$
3	GeneralFourRound \diamond	4 rounds	all teams	$O(n)$
4	SwapTeams	all rounds	2 teams	$O(n)$

complexity of calculating the change in z_{TTP} .

3.1 Level 1: Two-Round Neighbourhoods

SwapHomes involves 2 rounds and 2 teams. Suppose team a plays team b at home in round r and away in round s . Then the new tournament has a plays b away in round r and at home in round s . To evaluate the change in z_{TTP} requires at most 8 additions and 8 subtractions of distances between team locations, i.e., regardless of n the calculation is $O(1)$.

PartialSwapRounds involves 2 rounds and 4 teams. Suppose a plays b and c plays d in some round r and that a plays c and b plays d in some round s (this is often unlikely). Then the new tournament has a plays c and b plays d in round r and a plays b and c plays d in round s , with home-away assignments preserved. To evaluate the change in z_{TTP} requires at most 16 additions and 16 subtractions of distances between team locations, i.e., again the calculation is $O(1)$.

GeneralTwoRound is a *new* neighbourhood that generalises both **SwapHomes** and **PartialSwapRounds**. Consider any two rounds r and s of a tournament and represent each match in these two rounds as an edge coloured by its round (some edges may appear twice if two teams play each other twice in these two rounds). This graph is a set of $c \geq 1$ disjoint cycles, where each cycle has an even number of edges which alternate between colours. A rotation of a cycle in this graph is defined by swapping the colours of the edges in that cycle and the corresponding matches in rounds r and s are interchanged. This leads to a neighbourhood of 2^c new solutions defined by those cycles chosen to be rotated. For each neighbouring solution, the change in z_{TTP} can be evaluated in $O(n)$.

3.2 Level 2: Inter-Round Neighbourhoods

SwapRounds simply interchanges two rounds of the schedule.

More generally, consider any two rounds r and s . We can define the distance between round r and round s as the sum of the travel distance of each team moving from its location in round r to its location in round s . We wish to select a permutation of the rounds, so consider the Travelling Salesman Problem (TSP) instance in which each TSP-city corresponds to a TTP-round and the distance between TSP-

cities is the corresponding distance between TTP-rounds just described. We must also include a dummy TSP-city which corresponds to each team beginning (before round 1) and ending (after round $2n - 2$) the tournament at their home city.

`SwapRoundTwoExchange` is a *new* TTP-neighbourhood that applies a *TSP-two-exchange* (Lin and Kernighan 1973) to the TSP-cities corresponding to the TTP-rounds. Suppose the rounds are labelled $(r_0, r_1, \dots, r_i, r_{i+1}, \dots, r_j, r_{j+1}, \dots, r_n, r_0)$ where r_0 is the dummy TTP-round. Then a `SwapRoundTwoExchange` move simply reverses the subsequence of rounds (r_{i+1}, \dots, r_j) .

The change in z_{TTP} that results from a single `SwapRounds` or `SwapRoundTwoExchange` move can be evaluated in $O(n)$. However, if the complete inter-round distance matrix is calculated, requiring $O(n^3)$ to calculate, the subsequent change in z_{TTP} that results from a single `SwapRounds` or `SwapRoundTwoExchange` move can then be calculated in $O(1)$. Alternatively, for small n , we could find the optimal permutation of the rounds using a fast TSP-solver.

3.3 Level 3: Four-Round Neighbourhoods

`PartialSwapTeams` involves at least 4 teams. Suppose that a plays b and c plays d in round r . Preserving home-away assignments, we then force a plays d in round r , remove a plays b and c plays d from round r and remove a plays d from some other round s . Now suppose b plays e in round s . We then force a plays b in round s , remove b plays e from round s . Proceeding recursively, we eventually recover a feasible double round robin tournament.

`GeneralFourRound` is a *new* neighbourhood that generalises `PartialSwapTeams`. Consider any four rounds of a double round robin tournament. Consider the matches as edges which are coloured by their round, say red (r), green (g), blue (b) and black (k). Recall that the edges of any two colours form a set of disjoint even-length cycles. Given the edges of all four colours, we can find a set of four alternative rounds such that any two rounds include edges of at least 3 different colours. These can be discovered efficiently by depth-first search. An example is given in Figure 2, showing only four rounds of a tournament involving 10 teams.

3.4 Level 4: Inter-Team Neighbourhoods

`SwapTeams` simply interchanges the schedule of matches of two teams over all rounds. This is a highly drastic change to the schedule as every team and every round is affected. However, it only requires at most $O(n)$ additions and subtractions to evaluate the change in z_{TTP} .

3.5 A Proposed Improvement Logic

We propose an improvement logic within local search taking the neighbourhoods in layers.

Initial Tournament Construction. A round robin tournament can easily be constructed by the polygon method (de Werra 1981). However, the resulting tournament is very highly structured. For the purposes of this paper, we wish the initial tournament to be more random. Hence we construct two round robin tournaments

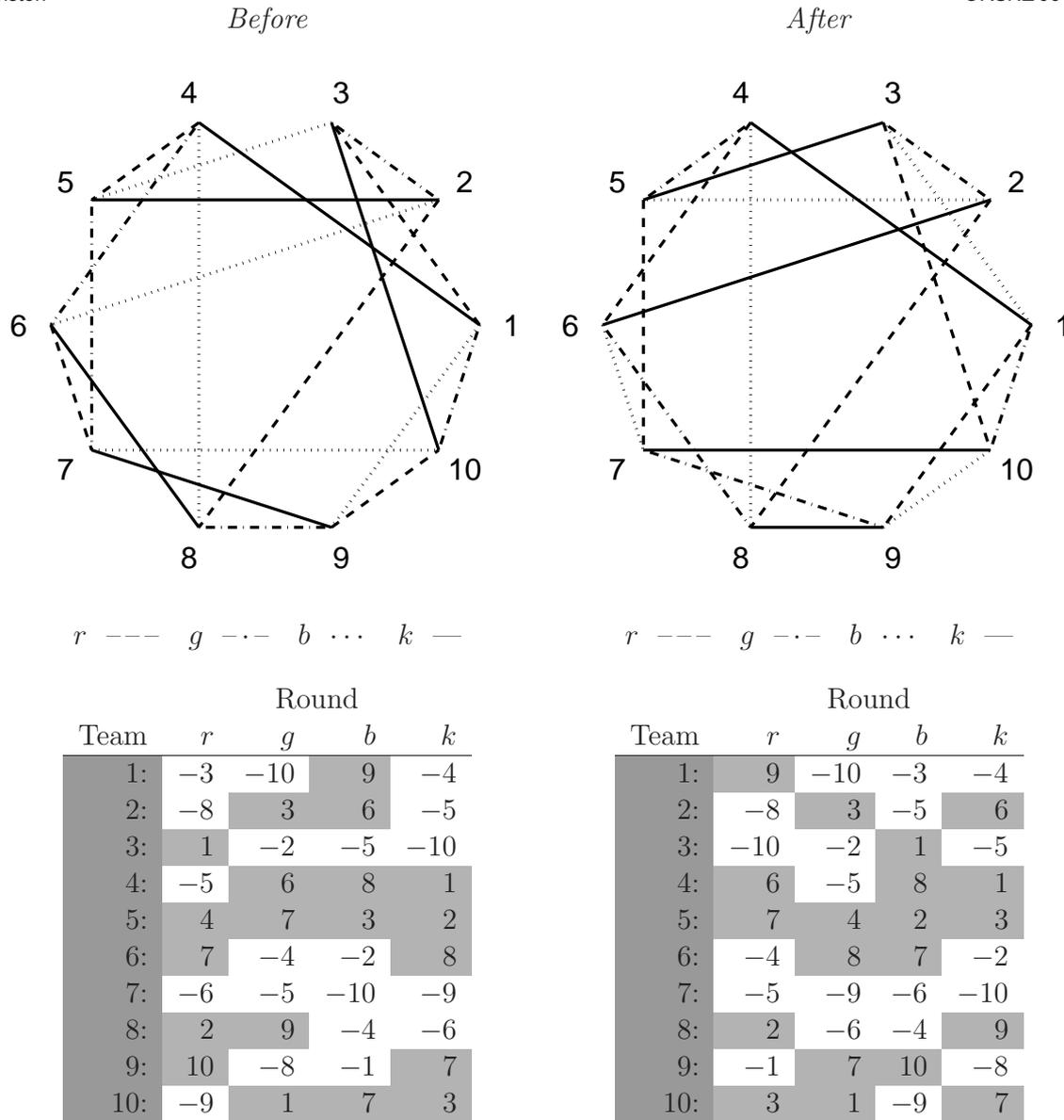


Figure 2: Example of GeneralFourRound move (before and after).

using the polygon method, randomly permute the team labels of the first tournament, randomly permute the team labels of the second tournament, put the two tournaments together while randomly assigning home-away status of each pair of matches between two teams, and finally randomly permute the rounds.

Intensification Phase. At level 1 we take the best-improving GeneralTwoRound move (subsuming the other two operators) until local optimality is attained. At level 2 we first calculate the complete inter-round distance matrix and then take a sequence of best-improving SwapRounds and SwapRoundTwoExchange moves until local optimality is attained. We alternate between levels 1 and 2 until the schedule is both level-1 and level-2 optimal.

Diversification Phase. At level 3 it is unlikely that any GeneralFourRound move will improve the objective. Also, enumerating and evaluating the whole level-3

neighbourhood is prohibitive (including level-1 and level-2 optimality for each solution in this neighbourhood). Hence `GeneralFourRound` is used to *diversify the search* through a random choice of the four rounds. The level-4 move `SwapTeams` is even more drastic as it affects the travel sequence of every team and all rounds, and hence is used more sparingly.

4 Computational Experience

We now report on preliminary computational experience with the TTP without side constraints. The aim is to compare the routes taken by each team with the optimal TSP tour through the corresponding home city locations. As an initial experiment, the teams are located at the vertices of a regular polygon and Euclidean distances are used. Hence the optimal TSP tour is simply the convex hull. We standardise the scale of each problem such that the optimal TSP tour has length $z_{TSP}^* = 1$.

Figure 3 provides a gallery of “good” solutions of different sizes found by local search as described in Section 3.5 but omitting level-3 neighbourhoods. We observe from similarly good solutions that good solutions are a collection of teams following full TSP tours, teams following two near-optimal subtours, teams following a “zigzag” pattern (e.g. team 1 of 8) and often one team following a perfect “star” tour (e.g. team 4 of 6 and team 5 of 8).

Star Conjecture. Anecdotal evidence suggests that most near-optimal tournaments have *exactly one* team following a perfect “star” tour. Hence we bravely conjecture that every optimal solution to an unconstrained TTP on a regular polygon contains exactly one team following a star tour.

5 Conclusions and Recommendations for Future Research

We have shown that visualisation of neighbourhoods and solutions leads to interesting insights into the geometric structure of good solutions found by local search. The compromise between efficient routing of away games and maintaining a double round robin tournament is evident in the visualisations, particularly in the surprising occurrence of perfect star tours.

Future work will involve a full implementation of the level-3 neighbourhood and using a metaheuristic such as tabu search to control diversification and intensification within the search levels. Also, methods for encouraging the presence of a star tour in each solution will be investigated. The computational experiment will be extended to the scenario where the home cities are located anywhere in the Euclidean plane and team routes are then compared to the corresponding optimal TSP tour. Following Anagnostopoulos et al. (2006) we will eventually include the `atmost` and `norepeat` constraints as soft constraints via penalty costs. It would also be interesting to develop further graphical visualisations that give insight into the effect of each type of local search neighbourhood.

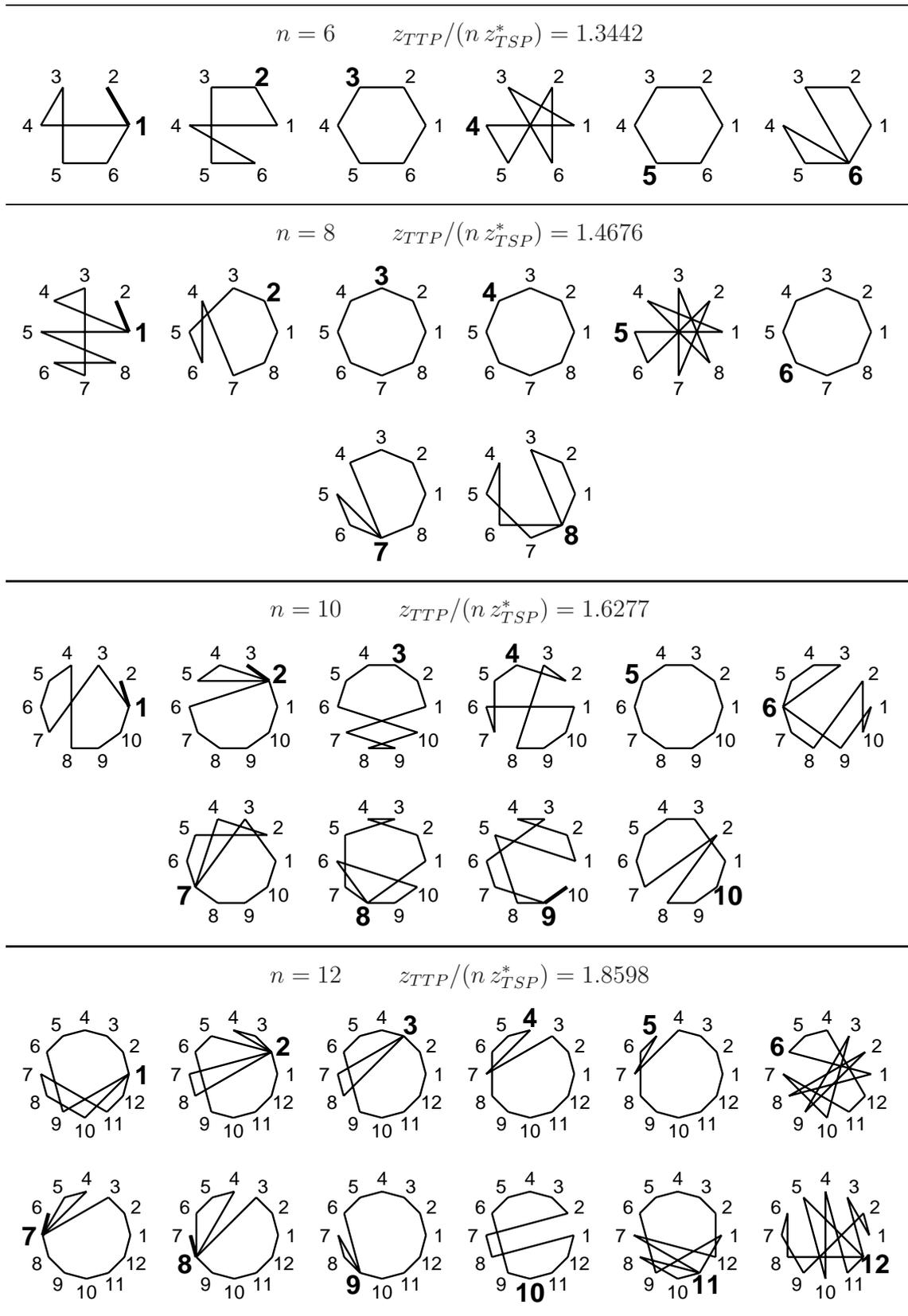


Figure 3: Gallery of “good” tournaments found by local search.

References

- Aarts, E., and J.K. Lenstra, eds. 1997. *Local Search in Combinatorial Optimization*. New York: John Wiley & Sons.
- Adriaen, M., N. Custers, and G. Vanden Berghe. 2003. An agent based metaheuristic for the traveling tournament problem. Working Paper, KaHo Sint-Lieven, Gent, Belgium.
- Anagnostopoulos, A., L. Michel, P. Van Hentenryck, and Y. Vergados. 2006. "A simulated annealing approach to the traveling tournament problem." *Journal of Scheduling* 9:177–193.
- Burke, E., and H. Rudova, eds. 2006. *Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2006)*. Brno, Czech Republic: Masaryk University.
- Crauwels, H., and D. Van Oudheusden. 2003. Ant colony optimization and local improvement. Workshop of Real-Life Applications of Metaheuristics, Antwerp, Belgium.
- de Werra, D. 1981. "Scheduling in sports." Edited by P. Hansen, *Studies on Graphs and Discrete Programming*. North-Holland, 381–395.
- Di Gaspero, L., and A. Schaerf. 2006. A composite-neighborhood tabu search approach to the traveling tournament problem. To appear in *Journal of Heuristics*.
- Easton, K., G. Nemhauser, and M. Trick. 2001. "The travelling tournament problem: description and benchmarks." *Proceedings of CP'01, Lecture Notes in Computer Science 2239, Springer*. 580–585.
- . 2003. "Solving the travelling tournament problem: a combined integer programming and constraint programming approach." *Proceedings of PATAT 2002, Lecture Notes in Computer Science 2740, Springer*. 100–109.
- Henz, M. 2004. "Playing with constraint programming and large neighborhood search for traveling tournaments." *Proceedings of PATAT 2004, Pittsburgh, USA*.
- Knust, S. 2006. Classification of literature on sports scheduling. Webpage: http://www.inf.uos.de/knust/sportlit_class/. Viewed: October 31, 2006.
- Lim, A., B. Rodrigues, and X. Zhang. 2006. "A simulated annealing and hill-climbing algorithm for the traveling tournament problem." *European Journal of Operational Research* 174:1459–1478.
- Lin, W., and B. W. Kernighan. 1973. "An effective heuristic algorithm for the traveling salesman problem." *Operations Research* 21:498–516.
- Ribeiro, C.C., and S. Urrutia. 2004. "Heuristics for the mirrored traveling tournament problem." *Proceedings of PATAT 2004, Pittsburgh, USA*. To appear in *European Journal of Operational Research*.
- Trick, M. 2006. Challenge traveling tournament instances. Webpage: <http://mat.gsia.cmu.edu/TOURN/>. Viewed: October 31, 2006.
- Urrutia, S., and C.C. Ribeiro. 2006. "Maximizing breaks and bounding solutions to the mirrored traveling tournament problem." *Discrete Applied Mathematics* 154:1932–1938.

Tournament Construction Methods for Auckland Bowls

Hamish Waterer
The University of Auckland
h.waterer@auckland.ac.nz

K. Chang, D. Ryan

Abstract

Bowls tournament draws are partial round robin tournaments in which each team plays another team at most once. A game is played between two teams on a single strip of grass called a rink, and each rink can only have one game per round. The number of rinks available is at least equal to the number of games per round. Some rinks may be marked as undesirable, and the assignment of games to undesirable rinks should be equitably shared between all teams. Consecutive games for a team should not be played on the same rink, and games should be arranged to minimize the number of games between teams from the same club.

Auckland Bowls look up tables of known draws to construct their tournaments. However, if they want to play a tournament with a number of teams and rinks that is not given in their tables they must construct the draw manually. We present a matching-based greedy sequential heuristic that constructs a high quality draw for any tournament. The heuristic is implemented in Microsoft Excel and minimizes the number of games where teams from the same club play each other, and the number of times a single team plays on the same rink, or on an undesirable rink is minimal.

Aluminium Production Scheduling Revisited

David Ryan
University of Auckland
d.ryan@auckland.ac.nz

Abstract

Molten aluminium of varying purities is produced in long lines of Heroult-Hall reduction cells in an aluminium smelter. The molten metal is tapped from three cells at a time to form batches. The batches are then mixed in furnaces before the metal is cast into billet or ingot. The cell batching problem can be formulated as a set partitioning optimization model, the solution of which maximizes the total value of metal in batches subject to a number of practical production constraints. Optimized batching results on production data will be reviewed. In a subsequent operation, the molten aluminium from a subset of batches is mixed in a furnace to be cast into a particular product with specified chemical composition as required by the order book. This second furnace scheduling problem can also be formulated as an optimization problem. The constraints ensure first that there is sufficient mechanical crane capacity to produce the required batches within the furnace fill time-window and second that the batches selected to be mixed in the furnace produce the required chemical composition of the product to be cast. This talk will focus on the furnace scheduling problem.

An Iterative Approach to Airline Scheduling

Oliver Weide
Department of Engineering Science
University of Auckland
New Zealand
o.weide@auckland.ac.nz

Abstract

In airline scheduling a variety of planning and operational decision problems have to be solved. We consider the problems aircraft routing and crew pairing: Aircraft and crew must be allocated to flights in a schedule in a cost minimal way. Although these problems are not independent, they are usually formulated as independent mathematical optimisation models and solved sequentially. This approach might lead to a suboptimal allocation of aircraft and crew, since a solution of one of the problems may restrict the solution for the problem solved later. Also, when cost minimal solutions are used in operations, a short delay of one flight can cause very severe disruptions of the schedule later in the day. We generate financially efficient solutions which are also robust to typical stochastic variability in airline operations. We solve the two original problems alternately. In each iteration penalty parameters are used to influence the solutions. Starting from a cost minimal solution, we produce a series of solutions which are increasingly robust. We stop when no further improvement of the robustness measure can be achieved.

1 Introduction

A sequence of planning problems must be solved in airline scheduling: First, marketing decisions in the *schedule design problem* determine which flights the airline operates. Each flight is specified by origin, destination, departure date, and departure time. Given the set of flights in a schedule the solution of the *fleet assignment model* determines which flight is operated by which aircraft type. The objective is to maximise profit with respect to the number of available aircraft and other resource constraints. Next, the *aircraft routing problem* seeks a minimal cost assignment of available aircraft to the flights. A routing is assigned to each individual aircraft such that each flight is covered by exactly one routing. The routings must satisfy maintenance restrictions. Once aircraft types are assigned to flights, the aircraft routing problem can be solved for each aircraft type separately. Similarly to the aircraft routing problem, the *crew pairing problem* (or *tour of duty problem*) allocates crew to flights in a minimal cost way. A set of generic crew pairings is constructed

with respect to a large number of rules such that each flight is covered exactly once. Under the assumption that the crew is only allowed to operate a single aircraft type (which is usually the case for pilots) the crew pairing problem can also be solved separately for each aircraft type. The last of the tactical planning problems is *crew rostering*. Based on the constructed crew pairings a line of work is assigned to each individual crew member. See Klabjan (2005) for a detailed description of the various airline scheduling problems.

In this paper two of the above problems, aircraft routing and crew pairing, are considered. The fleet assignment model is important for large airlines with multiple aircraft types. In the context relevant for this paper, the fleet can be regarded as homogeneous and fleet assignment can be omitted. The crew rostering problem can be viewed as a separate optimisation problem with no influence on the cost of the overall solution and is also not considered.

Traditionally, the aircraft routing problem is solved prior to the crew pairing problem. Both problems are not independent: After the arrival of a flight some minimal time is required for both aircraft and crew until the possible departure of the next flight. While these times are identical when crew stay on the same aircraft, the time needed by the crew increases whenever they change aircraft. Solving the aircraft routing problem first restricts the number of possible connections between two flights a crew is able to operate. This might lead to suboptimal solutions. Conversely, if the crew pairing problem is solved first, the aircraft routing problem might be infeasible.

We solve the aircraft routing problem and the crew pairing problem in one integrated model to obtain a single optimal solution for both problems. We focus on preserving the original set partitioning structures of both problems, as they are well understood and can be exploited in order to solve the individual problems which are already \mathcal{NP} -hard. We extend our model to identify solutions which also behave robust in operations. Robust solutions have a small planned cost and do not cause large additional costs once disruptions occur in operations. The approach follows the idea developed by Ehrgott and Ryan (2002).

We propose to couple the two original problems in a heuristic fashion. Both problems are solved alternately. In each step we change the cost structure of the aircraft routing and crew pairing problem to guide the solution process. We start with a cost minimal solution. We then generate a series of solutions which are increasingly robust. We stop when no further improvement of the robustness measure can be achieved.

Since only one aircraft type is considered the aircraft routing costs can be regarded as fixed. Hence, the only costs in the formulation are the costs of the crew pairing problem. Therefore it is particularly important to restrict the crew pairing problem as little as possible by aircraft routing decisions.

A number of recent publications address the integration of both problems. We are not aware of a heuristic approach that links the two problems and preserves their original structures.

Despite changing the cost structure, the two problems can be solved as efficiently as in the traditional approach. Also, only few iterations are needed to obtain solutions that are of low cost and operationally robust. Additionally a lower bound for the cost of the optimal solution is obtained which allows us to measure the quality of the generated solutions.

The paper is organised as follows: In Section 2 the problems are formally stated. We review the most recent approaches in the literature in Section 3. In Section 4 the iterative solution approach is presented. Section 5 concludes with numerical results generated by applying the approach to a real life problem.

2 Problem Formulation

In this section we describe the crew pairing and the aircraft routing problems and how they can be solved efficiently. We formulate a model that integrates the two problems and generates financially efficient solutions that are also robust.

2.1 The Crew Pairing Problem

Given a flight schedule the *crew pairing problem* is defined as the problem of assigning crews to flights in the schedule such that each flight is operated by exactly one crew. A sequence of flights which can be flown by a crew on one work day is a *duty period*. An alternating sequence of duty periods and rest periods is called *crew pairing* (or *tour of duty*). Any crew pairing must start and end at the same crew base and is restricted by a number of rules such as rest time regulations or flying time restrictions. Costs are associated with each crew pairing. In the crew pairing problem we seek to find a cost minimal set of pairings that partition the flights in the schedule, i.e. where each flight is contained in exactly one pairing.

The pairings can be represented as columns of a matrix $A^P \in \{0, 1\}^{m \times n^P}$ where m is the number of flights in the schedule and n^P is the number of possible pairings. Entry A_{ij}^P , $1 \leq i \leq m, 1 \leq j \leq n^P$ equals 1 if flight i is contained in pairing j and 0 otherwise. With this matrix representation we formulate the crew pairing problem as a standard *set partitioning model*:

$$\begin{aligned} & \text{Minimise} && c^{PT} x^P \\ & \text{subject to} && A^P x^P = \mathbb{1} \\ & && x^P \in \{0, 1\}^{n^P}. \end{aligned} \tag{1}$$

The element c_j^P of $c^P \in \mathbb{R}^{n^P}$ is the cost associated with pairing j . The decision variable $x_j^P \in \{0, 1\}$ is equal to 1 if pairing j is in the solution and 0 otherwise.

The number of pairings n^P is too large to efficiently consider all possible pairings. In the literature (see Klabjan (2005) for an overview of models used in airline scheduling) column generation and branch-and-bound techniques are identified as the most successful methods to solve the problem. In *column generation* only a small fraction of all possible pairings is considered initially and additional pairings (columns) are generated during the execution of the algorithm.

We model the schedule as a directed *flight network* where arcs represent flights. Additionally, *connection*-arcs are linking two flights that can be operated consecutively by the same crew. In this network each crew pairing is represented by a path. Hence, the column generation problem can be solved by a resource constraint shortest path problem. The rules the pairing must obey are incorporated into the shortest path algorithm.

To solve (1), first the LP relaxation of (1) is solved. Fractions appearing in this solution are caused by different crew pairings competing for the same flights (Ryan and Foster 1981). To eliminate these fractions and obtain an integer solution for (1), a branch-and-bound algorithm using a *constraint branching* strategy is used. Two flights in the solution are chosen that are partially covered by different crew pairings. In one branch only solutions are considered where both flights are contained in the same crew pairing. In the other branch both flights must not be operated by the same crew pairing.

2.2 The Aircraft Routing Problem

The *aircraft routing problem* (also referred to as *tail assignment*) is the problem of assigning aircraft to a given set of flights in a schedule. We seek to find one routing for each aircraft such that each flight of the schedule is contained in exactly one routing. Each routing is subject to maintenance requirements and other flying restrictions. The number of available aircraft is given and each particular aircraft is assigned to one specific routing, similar to the *crew rostering problem* where a line of work is assigned to a particular crew member. In the crew pairing problem on the other hand we seek an unknown number of generic pairings to partition all flights.

Similar to crew pairings, routings can be represented as columns of a matrix $A^R \in \{0, 1\}^{(m+a) \times n^R}$ where a is the number of available aircraft and n^R the number of possible routings. The first m rows are defined similar to those in A^P : The element $A_{ij}^R, 1 \leq i \leq m, 1 \leq j \leq n^R$ equals 1 if flight i is contained in routing j and 0 otherwise. Additionally the element $A_{m+i,j}^R, 1 \leq i \leq a, 1 \leq j \leq n^R$ equals 1 if routing j is operated by aircraft i and 0 otherwise. These constraints are referred to as *aircraft convexity constraints*. With this matrix representation the aircraft routing problem can be formulated similar to the crew pairing problem:

$$\begin{aligned} & \text{Minimise} && c^{R^T} x^R \\ & \text{subject to} && A^R x^R = \mathbb{1} \\ & && x^R \in \{0, 1\}^{n^R}. \end{aligned} \tag{2}$$

The element c_j^R of $c^R \in \mathbb{R}^{n^R}$ is the cost associated with routing j . The decision variable $x_j^R \in \{0, 1\}$ is equal to 1 if routing j is in the solution and 0 otherwise.

This form of the set partitioning model is called the *rostering model*. As in the crew pairing problem, the number of possible routings is very large and column generation techniques are used to obtain a solution for the LP relaxation of the problem. Using the same network representation as in the crew pairing problem the column generation problem is formulated as a resource constraint shortest path problem where we only generate columns which satisfy all rules.

In contrast to the crew pairing problem we branch on aircraft flight pairs to obtain integer solutions. In one branch a particular aircraft is forced to operate a particular flight while in the other branch the aircraft is not allowed to operate this flight. This branching strategy results from the observation that fractions in the solution of the LP relaxation of (2) are caused by different aircraft competing for the same flight.

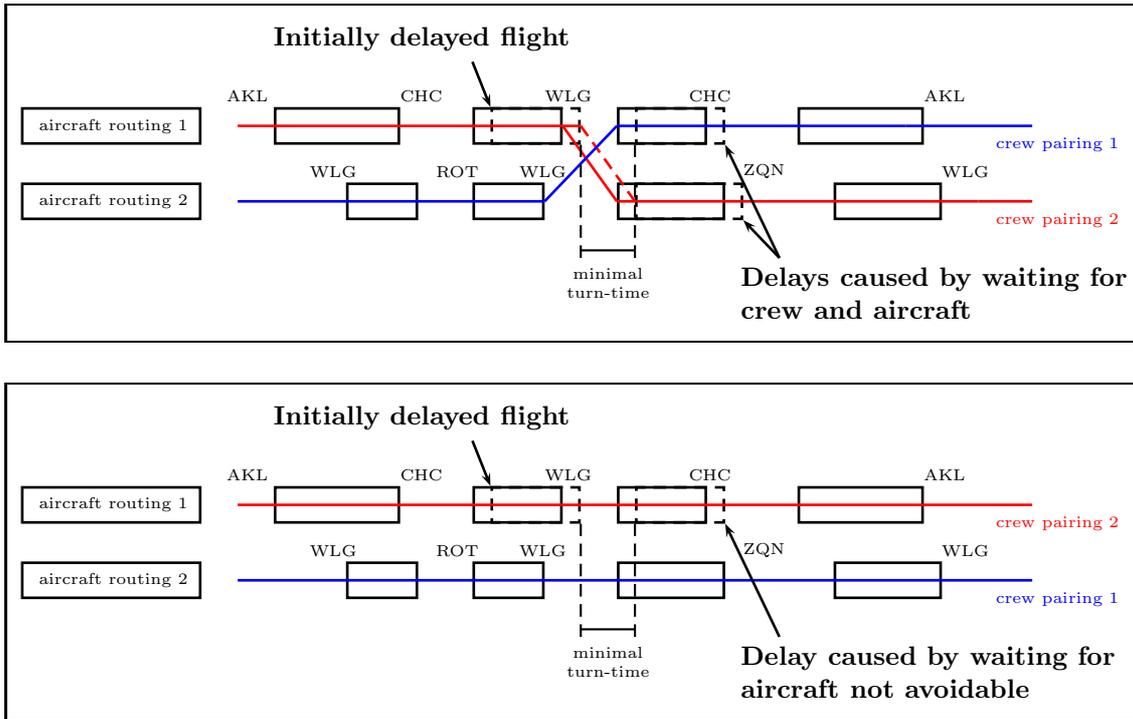


Figure 1: Comparison of a non-robust and a robust solution

2.3 Robust and Integrated Crew Pairing and Aircraft Routing Problem

If two flights can be operated in sequence by the same crew or aircraft (i.e. there exists a connection-arc linking both flights), the time between arrival of the first and departure of the second flight is called *turn-time* for aircraft and *sit-time* for crew.

The minimal time required for an aircraft or crew to operate a connection is called *minimal turn-time* or *minimal sit-time* respectively. The minimal sit-time usually exceeds the minimal turn-time. If the crew stays on the same aircraft both are identical. A connection between flights i and j is called *short* if

$$(\text{minimal turn-time})_{ij} \leq (\text{sit-time})_{ij} \leq (\text{minimal sit-time})_{ij}.$$

Thus, in a feasible solution, short connections are only allowed if the crew stays on the same aircraft. This condition might result in sub-optimal or infeasible solutions if the two problems are solved separately.

Also, we prefer solutions where crew are not changing aircraft when the turn time is below some *restricted time*. A connection between two flights i and j is called *restricted* if

$$(\text{minimal sit-time})_{ij} \leq (\text{sit-time})_{ij} \leq (\text{restricted time})_{ij}.$$

The reason for this is that in reality hardly anything works as planned. Delays occur frequently in airline operations due to late passengers, unscheduled maintenance requirements or bad weather, to name a few. Minimal turn-times are usually used in aircraft routings to keep costs low and connection times attractive for passengers. Hence, if a flight is delayed, the flight operated next by the aircraft is probably also delayed. But if the crew is also changing aircraft on a restricted connection after the delayed flight, one more flight might be affected by the initial delay. Because of the small buffer to compensate the delay (see Figure 1), the crew is likely to be late for the next flight they operate.

This behaviour can propagate to a large number of delayed flights in a short amount of time. To avoid this we try to find solutions in which crew are changing aircraft as rarely as possible if the connection is restricted. We call these aircraft changes *restricted aircraft changes*.

Such a solution, where effects of potential delays are minimal, is called *operationally robust*. The concept of robust solutions is important since these effects incur large additional costs, caused by additionally required crews, compensations for passengers affected by delayed or cancelled flights and damaged reputation of the airline. These costs may by far exceed the savings of using a solution with slightly less planned cost than using a solution that is more expensive but also more robust. We try to identify low cost solutions which are operationally robust, i.e. where disruptions will result in minimal recovery costs.

Crews are allowed to change aircraft if the connection is restricted but this will incur a penalty in the objective function. Our goal is to find solutions containing few restricted aircraft changes and are therefore expected to be robust.

In order to integrate the concepts of short and restricted connections into our formulation we enumerate all short and restricted connections.

We define a matrix $B^P \in \{0, 1\}^{m^B \times n^P}$ where m^B is the number of short connections. Each pairing is associated with one column of B^P , where $B_{ij}^P, 1 \leq i \leq m^B, 1 \leq j \leq n^P$ equals 1 if short connection i is contained in pairing j and 0 otherwise. Analogously for restricted connections, we define a matrix $D^P \in \{0, 1\}^{m^D \times n^P}$ where m^D is the number of restricted connections. $D_{ij}^P, 1 \leq i \leq m^D, 1 \leq j \leq n^P$ equals 1 if restricted connection i is contained in pairing j and 0 otherwise. For aircraft matrices $B^R \in \{0, 1\}^{m^B \times n^R}$ and $D^R \in \{0, 1\}^{m^D \times n^R}$ are defined in an analogous way.

With this matrix representation the *robust and integrated crew scheduling and aircraft routing problem* can be formulated as follows:

$$\begin{aligned}
& \text{Minimise} && c^{PT} x^P + c^{RT} x^R + c^{DT} d \\
& \text{subject to} && A^P x^P &= & \mathbb{1} \\
& && & A^R x^R &= & \mathbb{1} \\
& && B^P x^P - B^R x^R &\leq & 0 \\
& && D^P x^P - D^R x^R - d &\leq & 0
\end{aligned} \tag{3}$$

where $x^P \in \{0, 1\}^{n^P}$, $x^R \in \{0, 1\}^{n^R}$, $d \in \{0, 1\}^{m^D}$ and $c^D \in \mathbb{R}_+^{m^D}$ is some positive penalty parameter. Variable d_i equals 1 if restricted connection i is operated by a crew but no aircraft and 0 otherwise. The first two sets of constraints are identical to the original problem formulations. The third set of constraints enforces that short connections which are operated by some crew are also operated by some aircraft. The last set of constraints provokes additional cost in the objective function if restricted connections are operated by a crew and no aircraft.

3 Literature

Airline scheduling problems have been addressed in an extensive number of publications. See Klabjan (2005) for a detailed overview of the single as well as the integration of various airline scheduling problems. Recently, Sarac, Batta, and Rump

(2006) consider the aircraft routing problem and give a literature overview. We refer to Barnhart et al. (2003) for a detailed description of the crew pairing problem and a review of the literature addressing the problem. Also recently, Gopalakrishnan and Johnson (2005) give a comprehensive overview on state-of-the-art methods to solve the crew pairing problem. See Weide (2005) for an overview of models that integrate airline scheduling problems and incorporate robustness measures.

4 Iterative Solution Approach

The currently most successful approach in the literature to solve the integrated crew pairing and aircraft routing problem seems to be Benders decomposition (Mercier, Cordeau, and Soumis 2003). However, large computation time is needed to solve the problem to optimality.

We do not attempt to solve the integrated problem to optimality. Instead we couple the two problems in a heuristic fashion only solving the aircraft routing and crew pairing subproblems to optimality. We quickly obtain good quality solutions together with a measure of the solution quality.

We propose to solve the crew pairing problem and the aircraft routing problem repeatedly. We penalise restricted aircraft changes. Only the aircraft routings from the current iteration are taken into account. Discount factors make it attractive for aircraft to use the same restricted connections the crew is using in the previous iteration. The procedure is embedded into two loops. In the inner loop the penalty for the crew changing aircraft is increased while in the outer loop the restricted time is increased. In the following the single steps are described in detail.

Algorithm 1 Iterative algorithm

```

1: SOLVE crew pairing problem {No aircraft routings are taken into account}
2: while  $restrictedTime \leq maxRestrictedTime$  do
3:   while  $crewPenalty \leq maxCrewPenalty$  do
4:     SOLVE aircraft routing problem {Apply  $discountFactor$  to all restricted
     connections the crew is using in the previous iteration}
5:     SOLVE crew pairing problem {Apply  $crewPenalty$  to each connection used
     by crew and not by aircraft if  $sitTime \leq restrictedTime$ }
6:     INCREASE  $crewPenalty$ 
7:     BREAK if the robustness measure can not be improved
8:   end while
9:   INCREASE  $restrictedTime$ 
10: end while

```

We assume the aircraft routing cost to be fixed and consider the cost of the crew pairings as the only cost of the integrated solution. The robustness measure of each integrated solution is a weighted sum of all restricted aircraft changes. We choose weights which linearly increase when sit-time decreases. We search for a solution with low cost and low robustness measure.

We start with a crew penalty of zero and restricted time equal to the minimal sit-time. The cost of the crew pairing solution of Step 1 yields a lower bound on the total cost since no aircraft routings are taken into account. In Step 5 we solve the crew pairing problem for the aircraft routing solution calculated in Step 4 to obtain

a feasible solution for the integrated problem. The initial aircraft routing solution is driven by the cheapest possible crew pairing solution. After we obtain a solution to the integrated problem in the first iteration we improve the robustness value of this solution by increasing the penalties for restricted aircraft changes. We exit the inner loop if the robustness measure can not be improved. We then increase the restricted time to further increase robustness. We obtain a series of solutions with varying robustness values and costs. We apply this technique to real world data and present solutions in detail in Section 5.

The main advantage of this solution algorithm to the integrated problem is that the original set partitioning formulations of both problems remain unchanged. This preserves the properties that enable us to efficiently solve each subproblem. We observe that only few iterations are needed to obtain good quality solutions.

If aircraft routing and crew pairing solvers are available, the implementation of the iterative approach is straight forward. We influence the characteristics of the solutions via the costs of the underlying network structure. This is easily implemented into the shortest path computations of both problems: Discount factors are applied to restricted connections used by the crew in the network for the aircraft routing calculations. Since there are no costs associated with the aircraft routings the discount factors will enforce paths in the solution that contain as many restricted crew connections as possible. Also, we always obtain a feasible aircraft routing solution since all changes to the problem are in the objective function only. Vice versa in the crew pairing problem, each restricted aircraft change will incur a penalty during the shortest path calculation. The implementation of a master problem that controls both subproblems is straight forward.

5 Computational Experiments

The numerical tests are performed on a Linux (Fedora Core 5) desktop computer with a 2.6 GHz Pentium 4 processor and 1 GB RAM.

Flight schedule We use an interconnected flight network constructed of a domestic airline schedule. The schedule varies each day and we consider a dated time period of one week. The schedule contains 743 flights and 14 aircraft are available. Of 21780 connections present in the schedule, 41 are short and 1175 are restricted for a restricted time of 30 minutes plus min-sit time. All rules and restrictions applicable to aircraft routings and crew pairings were provided by the airline and are implemented in the algorithm. Hence, all generated solutions satisfy these rules and are ready to be used in practice.

Crew pairing We use *adjusted LP costs* as a measure of the cost of the integrated solution, which represent the true crew pairing costs without penalties taken into account. We use the LP costs rather than the IP costs to get a more accurate representation of the solution behaviour. In all calculations the bound-gap of the branch-and-bound algorithm is set to 2%. Setting the bound-gap to 0.5 % does not yield better solutions for the test examples but results in much longer computation time. We use a node limit of 500 since we want to keep computation times short. We do not generate any columns during the branch-and-bound process since the solution quality also does not improve. We store all columns we generated for later

iterations. We start the LP relaxation computation from the basis of the previous optimal LP solution. This results in a large decrease of computation times during the later iterations of the algorithm.

Results In Table 1 we present a selection of results. We list the results of the traditional approach where we solve the crew pairing problem for a given aircraft routing solution together with the results of our iterative approach. For each iteration we list computation times together with adjusted LP costs and the number of restricted aircraft changes for specified sit-times. We alter the restricted time as well as the penalty for crew changing aircraft during the algorithm. Only the aircraft changes with sit-time less or equal to the specified restricted time are relevant.

We observe that for a fixed restricted time and increasing crew penalty the robustness measure generally decreases while the costs increase. This is not always the case since the aircraft routings are changed in each iteration depending on the crew pairing solution of the previous iteration. Hence increasing the crew penalty may lead to solutions that incur less cost and are also more robust. If aircraft routings stay fixed we expect a anti-proportional relationship between costs and robustness measure if we increase the crew penalty.

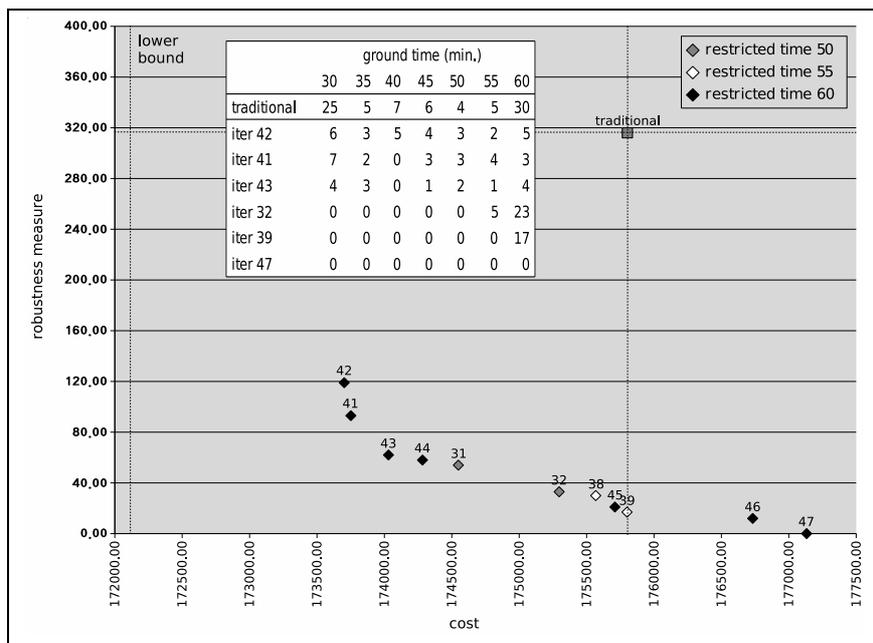


Figure 2: Comparison of traditional, infeasible lower bound and efficient solutions

The first iteration yields a crew pairing solution that incurs 2.03% less cost than the traditional solution. This solution is infeasible since no aircraft routings are taken into account but the solution value is a lower bound for the optimal solution value. The cheapest feasible solution we find incurs 1.20% less cost than the traditional solution. Hence, this solution is at most 0.8% more expensive than the optimal solution. In this solution only 28 restricted aircraft changes occur with sit-time less or equal to 30 minutes plus minimal sit-time while this number is 82 in the traditional solution.

We also find a solution with the same cost as the traditional solution but no restricted aircraft changes with a restricted time of 25 minutes plus min-sit time.

Note that in later iterations even if the crew penalty is small the number of restricted aircraft changes is very small compared to the early iterations. This is caused by crew pairings generated in previous iterations. These pairings do nicely reflect the behaviour of the aircraft routing solutions. This also explains the very small LP/IP gaps in the last iterations of the algorithm.

In Figure 2 the great benefits of the solution method become obvious. We plot efficient solutions generated by the iterative algorithm together with the traditional and infeasible lower bound solution. All solutions of the iterative approach perform better regarding cost or robustness than the traditional approach. Finally, the total running time to obtain these solutions is less than 4.5 hours which is acceptable for a long term planning problem.

6 Conclusion

We integrate the aircraft routing and crew pairing problems by combining already existing solution methods with a straight forward technique. Although optimality of the solutions can not be guaranteed, we obtain solutions that incur less costs and are significantly more robust than solutions currently used. Future research includes the investigation of a more sophisticated way to control the objective functions of both problems and obtain an optimal solution for the integrated problem.

References

- Barnhart, C., A. Cohn, E. Johnson, D. Klabjan, G. Nemhauser, and P. Vance. 2003. "Airline Crew Scheduling." In *Handbook of Transportation Science*, edited by R. Hall. Kluwer Scientific Publishers.
- Ehrgott, M., and D. Ryan. 2002. "Constructing Robust Crew Schedules with Bicriteria Optimization." *Journal of Multi-Criteria Decision Analysis* 11:139–150.
- Gopalakrishnan, B., and E.L. Johnson. 2005. "Airline Crew Scheduling: State-of-the-Art." *Annals of Operations Research* 140:305–337.
- Klabjan, D. 2005. "Large-scale Models in the Airline Industry." In *Column Generation*, edited by G. Desaulniers, J. Desrosiers, and M. Solomon. Kluwer Academic Publishers.
- Mercier, A., J. Cordeau, and F. Soumis. 2003. "A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem." Technical Report G-2003-48, GERARD.
- Ryan, D.M., and B.A. Foster. 1981. "An integer approach to scheduling." In *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*, edited by A. Wren, 269–280. Amsterdam: North-Holland.
- Sarac, A., R. Batta, and C.M. Rump. 2006. "A branch-and-price approach for operational aircraft maintenance routing." *European Journal of Operational Research* 175:1850–1869.
- Weide, O. 2005. "Robust and Integrated Airline Scheduling." *40th Annual ORSNZ Conference*. Wellington, 245–254.

Solving a Multi-level Capacitated Facility Location Problem by DVAM

Hsiao-Fan Wang, Ying-Yen Chen

Department of Industrial Engineering and Engineering Management,
National Tsing Hua University, Hsinchu, Taiwan, 30043, ROC.

hfwang@ie.nthu.edu.tw

Abstract

Facility location problems have drawn much attention in the literature, especially on the single level cases. There were some researches which considered solving the multi-level facility location problems, but they mainly focused on the uncapacitated issues. Because it is unrealistic with the unlimited capacities and multi-level facilities are a general case, therefore in this study, we shall discuss the problem of locating facilities over different levels in a multi-level distribution system where heterogeneous commodities are shipped from the multiple origins to multiple destinations via different levels of distribution facilities under capacity constraints. The objective is to determine the optimal set of facilities opened at each distribution level to satisfy the multiple demands of the clients with minimal total distribution costs including connection cost and opening cost.

The problem is formulated as a mixed integer linear program. An algorithm of Dynamic VAM (DVAM) is developed by adopting the concept of Vogel's Approximate Method (VAM). This polynomial time-complexity algorithm has shown its capability of finding optimal solution of small-scaled problems and near optimal solutions of large scaled problems with considerable efficiency and accuracy. An illustrative example was given in this paper and a real case of South Taiwan Emergent Medical Service System for Enterovirus Infection Patients has been applied to provide a valuable reference for patient allocation.

Keywords: Multi level facility location, Mixed integer programming, DVAM

1. Introduction

Multi-level facility location problem (MFLP) is the problem of locating facilities over different levels in a multi-level distribution system where commodities are shipped from the origin to the destination via a number of intermediate facilities, the higher the

level, the better is the skill. The objective is to determine the optimal set of facilities opened at each distribution level to satisfy the demand so that the total distribution costs including connection and opening costs will be minimized.

MFLP is originated from two-level facility location problem. Two-level facility location problem such as the distribution system of plants-warehouses-customers has attracted research attention for a century. A thorough review was provided by Elson (1972). MFLP includes two kinds of problem, Multi-level Uncapacitated Facility Location Problem (MUFLP) and Multi-level Capacitated Facility Location Problem (MCFLP). MUFLP was first addressed by Tcha (1984). Extending MUFLP to facility capacity constraints is MCFLP. MCFLP is more general and realistic than MUFLP and has been applied to many real cases such as hierarchical logistic system, telecommunication network and health insurance system. However, these problems are usually solved by simple heuristics because of its high complexity. Since most of the studies all were focused on the uncapacitated case, for instance, Aardal, *et al.*,(1999); Ageev (2000); Ageev (2004); Bumb,*et al.*,(2001) and Jain, *et al.*,(2001). Therefore, how to develop an efficient method for a more realistic MCFLP problem which is also able to improve the level of accuracy becomes an urgent issue.

The remainder of this paper is organized as follows. In Section 2, we define the mathematical formulation of the MCFLP. In Section 3, an algorithm based on the concept of Vogel's Approximation Method yet with dynamic cost at each iteration is proposed with its complexity analysis. An illustrative example is given in Section 4 for the comparison with Greedy method and LINGO. Finally, conclusions are drawn with discussions in Section 5.

2. Multi-level capacitated facility location problem

Let D be the set of demand points. Each client $j \in D$ has its own demand d_{jl} in each facility level l which must be served by precisely one facility at level l . And without lose of generality, let $d_{jL} \leq \dots \leq d_{j2} \leq d_{j1}$. Let F^l be the set of sites where facilities on level l , $1 \leq l \leq L$, are located, and assume that the sets F^l are pairwise disjoint with $\bigcap_{l=1}^L F^l = \Phi$, and thus denote $F = \bigcup_{l=1}^L F^l$. The cost of setting up a facility at site i_l is f_{i_l} with $f_{i_l} \geq 0$ for each $i_l \in F^l$. Let s_{i_l} denote the maximum capacity of facility i_l .

Let p denote a sequence of facilities $i_l \in F^l$, $l = 1, \dots, L$, and refer to a path of facilities. The set of all possible paths is denoted by P . Each client must be assigned to precisely one path $p \in P$. The total service cost incurred by assigning client j to path $p = (i_1, i_2, \dots, i_L)$ is equal to $c_{pj} = c_{j,i_1} + c_{i_1,i_2} + c_{i_2,i_3} + \dots + c_{i_{L-1},i_L}$. ($c_{i_0,i_1} \equiv c_{j,i_1}$)

The objective is to determine a set of facilities $Y^l \subseteq F^l$ on each level $l=1, \dots, L$ opened to each demand site so that the total cost of facility-opening and path-connecting is minimized.

Let y_{i_l} be equal to 1 if facility i_l is open at level l ; otherwise it equals to 0. And let x_{pj} be equal to 1 if client j is assigned to path p , and 0 otherwise. Then, a multi-level capacitated facility location problem described above can be formulated into a binary linear programming model as below:

(MCFLP)

$$1) \min \sum_{l=1}^L \sum_{i_l \in F^l} f_{i_l} y_{i_l} + \sum_{j \in D} \sum_{p \in P} \sum_{l=1}^L c_{i_{l-1}, i_l} d_{jl} x_{pj}$$

$$2) \text{s.t. } \sum_{p \in P} x_{pj} = 1, \quad \text{for each } j \in D,$$

$$3) \quad \sum_{p: i_l \in p} x_{pj} - y_{i_l} \leq 0, \quad \text{for each } j \in D \text{ and } i_l \in F^l,$$

$$4) \quad \sum_{j \in D} \sum_{p: i_l \in p} d_{jl} x_{pj} - s_{i_l} y_{i_l} \leq 0, \quad \text{for each } i_l \in F^l,$$

$$5) \quad x_{pj} \in \{0, 1\}, \quad \text{for each } p \in P, \text{ and } j \in D,$$

$$6) \quad y_{i_l} \in \{0, 1\}, \quad \text{for each } i_l \in F^l, \quad l=1, \dots, L$$

Apart from the objective function that minimizes both opening cost and connection cost, constraint (2) imposes each client being assigned to exactly one path; whereas constraint (3) requires a facility i_l opened to each a client j at each level l . In other words, it is impossible to assign client j to a path for using facility i_l unless facility i_l is open. If y_{i_l} equals to 0, then the sum of all assigned variables for client j to use paths containing facility i_l must be 0. Constraints (4) are the capacity constraints for the selected ($y_{i_l} = 1$) facility, which is modified from Aardal' model (1999).

Based on the structure of the proposed model above, we can derive the solution properties for the purposes of algorithm development:

Lemma 1 (Infeasibility Conditions). Let $TS_l = \sum_{i \in F^l} s_{i_l}$ and $TD_l = \sum_{j \in D} d_{j_l}$. Then if

one of the following two cases occurs, there is no feasible solution.

(1). For any $l = 1, \dots, L$, $TS_l < TD_l$.

(2). For any $j \in D$ and $i_l \in F^l$, $d_{j_l} > s_{i_l}$.

Lemma 2 (Optimality Condition). When the regret costs are all less than or equal to zero, the solution is optimum. In other words, for a minimization problem, the optimality condition is that the reduced costs are non-positive (Hillier & Lieberman, 2001).

Lemma 3. MCFLP is NP-hard.

If the demands from one client are the same at different levels, a single-demand MCFLP can be easily solved by setting $d_{j_1} = d_{j_2} = \dots = d_{j_L} = d_j$. On contrast to single-demand MCFLP, the ordinary MCFLP is called multiple-demand MCFLP. In practice, the single-demand and multiple demand MCFLPs can be regarded as two kinds of problems which are derived from two different operational strategies, namely, single demand strategy and multi-demand strategy.

Now, let us propose an algorithm for efficiently solving the above model.

3. The Proposed Algorithm

Basically, assigning each client to a set of facilities is a combinatorial problem. There are many kinds of software, such as LINGO, which have been developed for solving such kind of problems. Refer to Alper and Martin (2005), it is known that LINGO solves integer programming problems based on branch and bound algorithm. Due to the complexity of branch and bound algorithm, finding optimal solutions for the large scale problems is time consuming or even impossible.

A general purposed algorithm, Greedy Search, which is developed by Geoffrion and Van Toy (1979) has been an alternative and commonly used method. During the process, Greedy Search assigns paths to clients by the order of demand from the largest to the smallest. At each assignment, Greedy Search will choose the available path with minimal connection cost.

In the following, after a Vogel-Based algorithm is proposed, we shall compare the proposed algorithm with both LINGO and Greedy Search. It can be observed that Greedy method is easier and faster to implement, but is worse in performance. On the other hand, LINGO can find the optimal solution in a small-scaled problem but it can not cope with large scale problems.

Note that MCFLP can be regarded as a transshipment problem of which the clients are the sources, the intermediate-level facilities are transshipment nodes and the

highest-level facilities are the destinations. Since in our problem, each client has to be served by all levels of facilities, the problem can be transformed into a special transportation problem of which the clients are the sources and the paths connecting all levels of facilities are the destinations.

Because of NP nature of MCFLP from Lemma 3, finding an optimal solution for a large scale MCFLP is almost impossible. However, we still can make use of its properties as a transportation problem to find a good solution. Vogel's Approximation Method (VAM) is a method to find an initial basic feasible solution of a transportation problem with the properties of easy implementation and cost effectiveness. For MCFLP, to find or to approach the optimal solution is equivalent to find the most profitable feasible assignment. Based on optimality condition of Lemma 2, to find optimal solution is to get non-positive regret costs. To achieve this it relies on the reduction of the regret values as defined below :

Definition 1 (Regret Value). A *regret value* is defined as the arithmetic difference between the smallest and next-to-the-smallest costs of alternatives.

The concept of the regret value is the essence of VAM. Therefore, we shall develop an algorithm based on the concept of VAM for our MCFLP model.

While applying VAM to our model, it should be noted that in our MCFLP model, the costs in cost table are varied at each iteration, which are the crucial difference from the original VAM method and thus the revised VAM is called Dynamic VAM (DVAM). Such changes of the costs in the cost tables are due to the ingredient of the cost which consists not only the connection cost but also the additional opening cost when one path is first assigned to one client. While assigning path p to client j , the delta cost, denoted by Δ_{pj} , is equal to the connection cost c_{pj} and the opening costs of those facilities are not yet opened in path p . Because the additional opening costs may change when iterations go on, therefore the cost tables should also be updated. Once a facility i_l opens in an iteration, the costs Δ_{pj} in the cost table of the next iteration should be subtracted by f_{i_l} .

With DVAM, for each remaining row and column of cost table, we calculate its regret value as defined in Definition 1. Denote R_j as the regret value of the j th column in the cost table and R_{\max} , the maximal regret value over all j , in the following, the procedure of the proposed DVAM is presented:

Algorithm 1 (DVAM)

Step 1. (Feasibility test 1) Based on Lemma 1(a), If $TS_l < TD_l$ for any $l = 1, \dots, L$, terminate the procedure.

- Step 2.* (Feasibility test 2) Based on Lemma 1(b), if $d_{jl} > s_{i_l}$ for any $j \in D$ and $i_l \in F^l$, terminate the procedure.
- Step 3.* Compute Δ_{pj} for each $p \in P$ and $j \in D$ and list in the cost table.
- Step 4.* Let D' denote the set of all clients which have not been served yet. Let $\Delta_j' = \min_{p \in P} \Delta_{pj}'$, the minimal delta cost. And let Δ_j'' denote the second minimal delta cost. Find Δ_j' and Δ_j'' for each $j \in D'$ in $p \in P$. Compute $R_j = \Delta_j'' - \Delta_j'$ for each $j \in D'$.
- Step 5.* Find the client j^* with the maximal regret value, R_{\max} . Identify the available path p^* with the minimal delta cost.
- Step 6.* If $d_{j^*l} \leq s_{i_l}'$ for each $i_l \in p^*$, assign path p^* to client j^* . Otherwise there exists one $d_{j^*l} > s_{i_l}'$ for any $i_l \in p^*$, find out the available path with the next minimal delta cost and set this path as path p^* . Return to Step 6. If there is no more available path, then the solution can not be found and the procedure terminates.
- Step 7.* Let F^* denote the set of the facilities which have not been opened yet in path p^* . Open each facility in F^* . Update the delta cost in the cost table $\Delta_{pj} = \Delta_{pj} - \sum_{i_l \in p \cap F^*} f_{i_l}$ for each $p \in P$ and $j \in D'$. Update the spare capacity of facilities $s_{i_l}' = s_{i_l}' - d_{j^*l}$ for each $i_l \in p^*$. Return to Step 4.
- Step 8.* Repeat Step 4 ~ Step 7 until all clients are served.

The time complexity of each step in DVAM is $O(L \times |D|)$, $O(|F| \times L \times |D|)$, $O(L \times |P| \times |D|)$, $O(|P| \times |D|)$, $O(|D|)$, $O(L)$, $O(L \times |P| \times |D|)$ and $O(|D| \times L \times |P| \times |D|)$, respectively. Hence, the complexity of DVAM is $O(L \times |P| \times |D|^2)$. Since the complexity of the greedy solution is determined by sorting the connection cost with $O(|D| \times |P| \log |P|)$ and in general, the number of the demand is less than the number of the facility sequences, therefore, the complexity of DVAM is not much higher than the complexity of the greedy solution, but is much less than the complexity of the problem, $O(|P|^{|D|})$. If single demand is required, our proposed algorithm can easily solve it by setting demands d_{jl} of different levels by d .

4. Illustrative example

To compare the accuracy of DVAM and the greedy method, let us take the following example for both illustration and comparison purposes. Although we use an example of the single demand MCFLP for ease of explanation, we treat it as an ordinary MCFLP. There are four clients with demands 400, 600, 900 and 300, two facilities at level 1 with capacities 2000 and 1600 and two facilities at level 2 with capacities 10000 and 7000. The opening costs of these four facilities are 140, 120, 500 and 400. The network of this problem is drawn in Figure 1. The twelve line segments in Figure 1 are all of the legal paths. The locations of these eight nodes are indicated by their coordinates as shown in Table 1 and the connection cost of any legal path segment is assumed to be the Euclidean distance between the end nodes of that path segment.

Table 1. Locations of All Nodes

Facility	Locations			
	F ₁₁ =	F ₁₂ =	F ₂₁ =	F ₂₂ =
Cites	(30,60)	(60,40)	(40,40)	(60,60)
Demand	D ₁ =	D ₂ =	D ₃ =	D ₄ =
Nodes	(10,70)	(30,20)	(80,90)	(90,40)

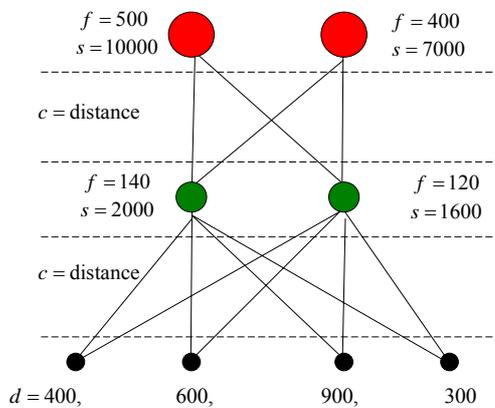


Figure 1. Network of the Illustrative Example

To illustrate the procedure, one iteration of the procedure is shown in details with a parameter table shown below where the resulting basic variable is given in the lower right-hand corner.

Step 1. Feasibility test 1: $TS1 = 3600$, $TS2 = 17000$ and $TD1 = TD2 = 2200$. $TS1 \geq TD1$, $TS2 \geq TD2$.

Step 2. Feasibility test 2: $\text{Max}\{d_{jl}\} = 900, \text{Min}\{s_i\} = 1600. \text{Max}\{d_{jl}\} < \text{Min}\{s_i\}$.

Therefore there does not exist one $d_{jl} > s_i$.

Step 3. Compute Δ_{pj} and list in the cost table as Table 2.

Table 2. Iteration 1 of DVAM

	Client				Capacity	
	1	2	3	4		
	1	684.7	702.4	720.7	725.6	2000 10000
Facility	2	698.3	676.1	693.9	670.0	1600 10000
Sequence	3	592.4	610.0	628.3	633.2	2000 7000
	4	598.3	576.1	593.9	570.0	1600 7000
Demand		400	600	900	300	
		400	600	900	300	$x_{44}=1$
Regret		5.9	33.9	34.5	63.2	

Note : : R_{max} ; : corresponding minimal delta cost

Step 4. $R1 = \Delta_1'' - \Delta_1' = 598.3 - 592.4 = 5.9. R2 = \Delta_2'' - \Delta_2' = 610.0 - 576.1 = 33.9.$

$R3 = \Delta_3'' - \Delta_3' = 628.3 - 593.9 = 34.5. R4 = \Delta_4'' - \Delta_4' = 633.2 - 570.0 = 63.2.$

Step 5. $R_{max} = R4, j^* = 4. p^* = 4.$

Step 6. $d_{41} = 300 \leq 1600 = s_{2_1}, d_{42} = 300 \leq 7000 = s_{2_2}$. Hence, $x_{44} = 1.$

Step 7. Open facility site 21 and 22. Update the delta cost in the cost table. For example, $\Delta_{41} = \Delta_{41} - (f_{2_1} + f_{2_2}) = 598.3 - (120 + 400) = 78.3$. Update the spare capacity of facilities $s_{2_1}' = 1600 - 300 = 1300$ and $s_{2_2}' = 7000 - 300 = 6700.$

Return to Step 4.

After 4 iterations, the allocation finished and the procedure stops with solutions of $x_{44}=1, x_{43}=1, x_{32}=1$ and $x_{31}=1.$

In this example, the final solution obtained from DVAM is the same as the optimal solution solved by the software of LINGO program of which the objective value is 906.2123, but the objective cost of the greedy solution is 1482.3063. In Section 3, we can see the complexity of DVAM is not much larger than the complexity of the greedy method. However, DVAM can reach the optimal solution whereas the greedy solution

is much further away from the optimal solution even in a small problem as shown above.

5. Conclusions

Facility location problems have drawn much attention in the literature, especially on the single level cases. There were some researches which considered the multi-level capacitated facility location problems, but they mainly focused on the uncapacitated issues. Because it is unrealistic of unlimited capacities and multi-level facilities are a general case, therefore in this study, we considered the capacitated case, the MCFLP and formulated the MCFLP as a binary linear program.

Due to the NP nature of the problem, we have developed an algorithm base on the concept of Vogel's Approximation method (VAM). Because the costs in our cost table are varied at each iteration, therefore the developed algorithm is called Dynamic VAM (DVAM).

The algorithm has been illustrated by a small example. By comparing with the results of LINGO and Greedy Method, the proposed algorithm has shown to be more efficient and effective.

The proposed method has been applied to a real case of South Taiwan Emergent Medical Service System for Enterovirus Infection patients. The result provided a useful reference for patient allocations.

Acknowledgment

The authors acknowledge the financial support from the National Science Council, Taiwan, ROC with project no. NSC94 -2213-E007-018.

6. References

- Aardal K, Chudak FA, Shmoys DB.,1999, "A 3-approximation algorithm for the k-level uncapacitated facility location problem." *Information Processing Letters*; **72** :161–167.
- Ageev AA., 2000, "Improved approximation algorithms for multilevel facility location problems." *Operations Research Letters* ,**30** : 327–332.
- Ageev AA, Zhang Y, Ye J.,2004, "Improved combinatorial approximation algorithms for the k-level facility location problem." *SIAM J. Discrete Math.***18**; 207–217.
- Alper A, Martin W P S.,2005, "Integer programming software systems." *Annals of Operations Research* ,**140**: 67-124.
- Bumb AF, Kern W.,2001, "A simple dual ascent algorithm for the multilevel facility location problem" in *Proceedings of the 4th International Workshop on Approximation Algorithms for Combinatorial Optimization. Lecture Notes in Computer Science* ,. **2129**, Springer, Berlin; 55–62.

- Elson DG.,1972, "Site location via mixed-integer programming." *Operational Research Quarterly*,**23**: 31-43.
- Geoffrion A, Van Toy T J.,1979, Common sense planning methods can be hazardous to your corporate health. *Sloan Management Review*,**20**; 30-42.
- Hillier FS, Lieberman GJ.,2001, *Introduction to Operations Research*. North America: McGraw Hill.
- Jain K, Vazirani VV.,2001, "Approximation algorithms for metric facility location and k-median problems using the Primal-dual schema and Lagrangian relaxation." *J. ACM* , **48**: 274–296.
- Tcha D, Lee B.,1984, "A branch-and-bound algorithm for the multi-level uncapacitated location problem." *European Journal of Operational Research* **18**: 35–43.