OPERATIONAL RESEARCH SOCIETY OF NEW ZEALAND

42nd ANNUAL CONFERENCE

29-30 November 2007

University of Auckland
Auckland
New Zealand

Sponsored by
The Energy Centre (UoA)
The Electricity Commission
Transpower
Optima Corporation
HRS

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

The Energy Centre of the University of Auckland
The Electricity Commission
Transpower
Optima Corporation
HRS
The Department of Engineering Science
University of Auckland

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

Kia Ora and welcome,

The papers in this volume form the Proceedings of the 42nd Annual Conference of the Operational Research Society of NZ (ORSNZ) held 29-30 November 2007 at the University of Auckland, Auckland, New Zealand.

This ORSNZ conference takes advantage of the sabbatical visits of Professor Michael Trick and Professor Gerard Cachon to Auckland. Both Professors Trick and Cachon have kindly agreed to be our plenary speakers. Professor Trick has been the Hood fellow at the University of Auckland during 2007, and throughout his visit to New Zealand he has promoted the subject of Operations Research in several public lectures and advocated the essential involvement of OR in industry. Professor Cachon arrived in Auckland in September and is staying until June 2008.

This conference could not have been organised without the invaluable assistance of staff and graduate students from the Department of Engineering Science at the University of Auckland. I would like to specially thank Andrew Mason, Hamish Waterer, Tony Downward, Ziming Guan, Amir Joshan and Oliver Weide. I would also like to thank Mark Wilson for organizational and technical assistance. Many thanks also to all the authors who have contributed to this volume.

We are most grateful for the strong support provided by our sponsors; The Electricity Commission, The Energy Centre of the University of Auckland, Transpower, Optima Corporation, HRS and the Department of Engineering Science at the University of Auckland.

Golbon Zakeri
Conference chair
November 2007

Table of Contents

Thursday 29 November

09:00–10:00 Plenary 1

10:30–12:10 Session T1: OR in the electricity industry

13:30–14:50 Session T2: Tools for the application of OR

15:20–17:00 Session T3: YPP – Young Practitioner Prize

Friday 30 November

09:00–10:00 Plenary 2

10:30–12:10 Session F1: Optimization over networks

13:30–14:50 Session F2: YPNP – Young Practitioners Not Participating in the YPP

15:20–16:40 Session F3: OR applications

Author	Session	Time
Anderson, Eddie	T1	10:50-11:10
Bava, Vitesh	T3 (YPP)	15:20-15:40
Cachon, Gerard	Plenary Friday	09:00-10:00
De Pont, Nick	T3 (YPP)	16:00-16:20
Downward, Anthony	F2	13:30-13:50
Hartigan, Colm	F2	13:50-14:10
Johnston, Mark	F1	11:50-12:10
Joshan, Amir	F2	14:10-14:30
Leyland, Geoff	T2	13:50-14:10
MacLeod, Mr Craig	T2	14:30-14:50
Mason, Andrew	F1	10:50-11:10
McIntosh, Gregory	T2	14:10-14:30
McNaughton, Alastair	F3	16:20-16:40
Mitchell, Stewart	T2	13:30-13:50
Paynter, John	F3	15:20-15:40
Peat, Mr Martin	T3 (YPP)	15:40-16:00
Philpott, Andy	T1	10:30-10:50
Pritchard, Geoff	T1	11:10-11:30
Raith, Andrea	F2	14:30-14:50
Rouse, Paul	T1	11:30-11:50
Ryan, David	F1	11:30-11:50
Shao, Ms Lizhen	F3	16:00-16:20
Suri, Ratneesh	F3	15;40-16:00
Tam, Bassy	T3 (YPP)	16:20-16:40
Trick, Michael	Plenary Thursday	09:00-10:00
Van Ryzin, Garrett	F1	10:30-10:50
Weide, Oliver	F1	11:10-11:30
Zakeri, Golbon	T1	11:50-12:10
Zhang, Lei	T3 (YPP)	16:40-17:00

The Science of Better and Better Together

Michael Trick
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA, USA

Abstract

Operations Research (OR) is the art and science of better decision making. By using mathematical models, organizations and individuals create value through these better decisions. Historically, OR has provided incremental improvement through better scheduling, resource allocation, and distribution planning but these improvements have been limited due to lack of data or limited computational power. With the recent trends towards massive data sets and significant computational power, combined with algorithmic advances in the field, OR is becoming much more relevant to practice. In contrast to these trends, OR as a field appears to be struggling. Professional society membership is decreasing, and there is a perception that academic course offerings are also on the decline. ”OR groups” in industry are a vanishing breed. How can we reconcile this dichotomy, and what can we do about it? I'll bring some thoughts from my experiences in the US and New Zealand, along with consulting work I have done with Major League Baseball and the United States Postal Service and others.

Michael Trick is Professor of Operations Research at the Tepper School of Business, Carnegie Mellon University, where he has been on faculty since 1989. His research interests are in computational integer programming and applications in sports and social choice. In 2002, he was President of INFORMS he is currently the Vice-President/North America for the International Federation of Operational Research Societies. In 2007, he was Hood Fellow and Honorary Research Fellow at the University of Auckland.

Uniform-price auctions versus pay-as-bid auctions

Andy Philpott
Department of Engineering Science
University of Auckland
(joint with Eddie Anderson, UNSW)

Abstract

We consider the problem of optimizing a supply function bid into a discriminatory auction in which each agent is paid their bid price on each increment of offered capacity. The efficiency of pay-as-bid auctions in comparison with uniform-priced auctions has been debated, as it is conjectured that in the former auction, agents will simply bid their offer prices up to an anticipated clearing price. Using market distribution functions, we derive optimality conditions for each agent in a pay-as-bid auction, and compute Nash equilibria in supply functions. In most realistic cases, there are no pure-strategy equilibria in this game. In the absence of capacities and price-caps, there are infinitely many mixed-strategy equilibria, which become uniquely determined when generators have limited capacities and are subject to a price cap.

Learning implicit collusive behaviour in electricity markets.

Eddie Anderson
Australian School of Business
University of New South Wales

Abstract

When oligopolies involve repeated play (as in wholesale electricity markets) we can expect collusive outcomes to emerge. We model the learning of such collusive behaviours using a genetic algorithm in a stylised model of an electricity market. We show that collusion occurs even though the system may not settle into a single stable collusive equilibrium. We show that implicit collusion has the most importance in markets in which there is an intermediate amount of market power. It is hard to learn collusive patterns of behaviour in markets which are either highly competitive, or in which one player has substantial market power.

AN APPLICATION OF A MALMQUIST PRODUCTIVITY INDEX TO NEW ZEALAND ELECTRICITY LINE BUSINESSES

Paul Rouse and Annie Wing Huen Leung
Department of Accounting and Finance
University of Auckland
New Zealand
p.rouse@auckland.ac.nz

Abstract

This paper describes the different treatments of the Malmquist productivity index such as the Quasi-Malmquist DEA, the generalised Malmquist productivity index and Desli and Ray (1997) Malmquist productivity index. The base and adjacent period Malmquist productivity indexes are explained and an application of the base period Malmquist DEA on the New Zealand electricity line companies during 1999 to 2003 is carried out. This analysis enables us to decompose the conventional Malmquist productivity index into technological and efficiency changes to test whether the technology has changed throughout the sampling period. The results indicate a positive shift in technology between 1999 and 2003 with improvements in technical efficiency during 1999 to 2000, 2001 and 2002, but a decline in technical efficiency during 1999 to 2003. The evidence also shows that capacity has a significant impact on efficiency but has an insignificant impact on the changes in efficiencies or technology of the line companies.

.

1 Introduction

In 1987, the electricity industry was restructured as part of the Labour Government’s plan for substantial changes of key New Zealand Government services. Since then, various modifications have been implemented by successive New Zealand Governments, with a major preoccupation with managing the monopoly on line operations in the electricity industry. In April 1999 in a major change aimed at reducing operating costs and improving efficiency, the industry was split into generation, transmission and retail components in order to reduced monopoly operation and encourage competition.

Applying Data Envelopment Analysis (DEA) to data from 1999 to 2003 we estimate a Malmquist (1953) index which separates the effects of changes in productivity into technical efficiency and technological changes. We also describe different index decompositions focusing on the Quasi-Malmquist DEA, the generalised Malmquist productivity index and the Desli and Ray (1997) Malmquist productivity index. Specifically, the base and adjacent period Malmquist productivity indexes are examined in more detail with the former applied to New Zealand electricity line companies (ELBs).

2 Malmquist DEA

The Malmquist index was introduced by Malmquist in 1953 and further developed using DEA by Färe et al. (1994). The variable returns to scale output oriented Malmquist productivity index (M_o^{t+1}) can be written as:

$$M_o^{t+1}(x^t, y^t, x^{t+1}, y^{t+1}) = \left[\frac{D_o^t(x^{t+1}, y^{t+1})}{D_o^t(x^t, y^t)} \times \frac{D_o^{t+1}(x^{t+1}, y^{t+1})}{D_o^{t+1}(x^t, y^t)}\right]^{1/2}$$

$$M_o^{t+1}(x^t, y^t, x^{t+1}, y^{t+1}) = \left[\frac{D_o^{t+1}(x^{t+1}, y^{t+1})}{D_o^t(x^t, y^t)}\right] \times \left\{\left[\frac{D_o^t(x^{t+1}, y^{t+1})}{D_o^{t+1}(x^{t+1}, y^{t+1})} \times \frac{D_o^t(x^t, y^t)}{D_o^{t+1}(x^t, y^t)}\right]\right\}^{1/2}$$

$$= TE\Delta(x^t, y^t, x^{t+1}, y^{t+1}) \times T\Delta(x^t, y^t, x^{t+1}, y^{t+1})$$

where period t is the benchmark technology period, x is the input vector and y is the output vector. As shown, this can be decomposed into two components, where the first denotes the Debreu-Farrell technical efficiency change (TEΔ_c (x^t, yt, x^{t+1}, y^{t+1})), and the second denotes the technological change (TΔ_c (x^t, yt, x^{t+1}, y^{t+1})), i.e., the shift in production frontier.

Although this index enables the decomposition of technical changes and technical efficiency changes, it ignores non-radial slacks and the effects of returns to scale. This gives rise to the possibility of misallocating productivity changes to technological change or efficiency change.

Grifell-Tatjé, Lovell and Pastor (1998) introduced the quasi-Malmquist productivity index that incorporates the non-radial form of inefficiency defined in Koopmans (1951) definition of efficiency. They use a weighted additive output oriented DEA model instead of the conventional DEA model to calculate the non-radial efficiency scores for the quasi-Malmquist index, and obtained the total output slacks from the DEA model, where $(\phi - 1)y_m^t$ represents radial slack and r_m^t represents non-radial slack:

$$S_m^t = (\phi - 1)y_m^t + r_m^t \qquad \text{m = 1, …, M}$$

Using the total output slacks (S_m^t) obtained, the non-radial technical efficiency is defined as:

$$Q_o^t(x^t, y^t) =: \left[1 + \frac{\Sigma_m(s_m^t / y_m^t)}{M}\right] ,$$

The quasi-distance functions ($Q^t_0(x^t, y^t)$) used to calculate the quasi-Malmquist productivity index is the inverse of the above non-radial technical efficiency measure. In the case where there are no output slacks, $Q^t_0(x^t, y^t)$ will equal $D^t_0(x^t, y^t)$ in the conventional Malmquist index and the quasi-Malmquist productivity index will be equivalent to the conventional Malmquist productivity index.

Grifell-Tatjé and Lovell (1999) provide a generalised Malmquist productivity index which decomposes into the conventional Malmquist productivity index and a Malmquist scale index as follows:

$$G_0^t(x^t, y^t, x^{t+1}, y^{t+1}) = M_0^t(x^t, y^t, x^{t+1}, y^{t+1}) \times E_0(x^t, y^t, x^{t+1})$$
$$= M_o^t(x^t, y^t, x^{t+1}, y^{t+1}) \times \left[\frac{S_0^t(x^{t+1}, y^t)}{S_0^t(x^t, y^t)}\right]$$

where $M_0^t(x^t, y^t, x^{t+1}, y^{t+1})$ is the conventional Malmquist productivity index and $E_0(x^t, y^t, x^{t+1})$ is the period t Malmquist scale index defined as the ratio of a pair of output oriented scale efficiency measures relative to the period t technology.

If $E_0(x^t, y^t, x^{t+1}) > 1$, it means that (x^{t+1}, y^t) is more scale efficient than (x^t, y^t), i.e., scale economies contributed positively to the productivity change over the two periods, t and t+1, and vice versa if $E_0(x^t, y^t, x^{t+1}) < 1$. When $E_0(x^t, y^t, x^{t+1}) = 1$, it means that the degree of scale efficiency is the same between the benchmark period (t) and the period t+1.

In contrast, Desli and Ray (1997) introduce the variable returns to scale frontier as a benchmark and measure technical change using ratios of variable returns to scale distance functions:

$$M_{0c}^t(x^t, y^t, x^{t+1}, y^{t+1}) = \left[\frac{D_{0c}^t(x^{t+1}, y^{t+1})}{D_{0c}^t(x^t, y^t)}\right]$$
$$= \left[\frac{D_0^t(x^{t+1}, y^{t+1})}{D_0^{t+1}(x^{t+1}, y^{t+1})}\right] \times \left[\frac{D_{0c}^{t+1}(x^{t+1}, y^{t+1})}{D_0^t(x^t, y^t)}\right] \times \left\{\left[\frac{D_{0c}^t(x^{t+1}, y^{t+1})}{D_0^t(x^{t+1}, y^{t+1})}\right] \Big/ \left[\frac{D_{0c}^t(x^t, y^t)}{D_0^t(x^t, y^t)}\right]\right\}$$

where D_{0c}^{t+1} represents constant returns to scale and D_0^{t+1} represents variable returns to scale distance functions. The first term in the above equation represents the technical change measured by a ratio of variable returns to scale distance functions; the second term represents the technical efficiency change measured also by a ratio of variable returns to scale distance functions; the final term, the only term that is different from the Grifell-Tatjé et al. (1998) method of decomposition, represents the scale efficiency change relative to period-t technology using different data (i.e., (x^t, y^t) and (x^{t+1}, y^{t+1})) from Grifell-Tatjé *et al.*'s scale efficiency change term, which uses data (x^t, y^t) and (x^{t+1}, y^t).

2.1 Base and Adjacent period Malmquist productivity indexes

The change in productivity over time can be analysed in two ways (Althin, 2001; Asmild et al., 2004): a base-period comparison, which compares data for different years with a base year (usually the first year in the sampling period), and an adjacent-period analysis, which compares data for different years with the prior year's data.

The main difference between the two methods is that the base period measure of technical change depends on the fixed base period's technology. This can be problematic when there are significant changes in the technology of the dataset during the sampling period, or when a long sampling period is employed. For example, when technical changes are measured for the year 2000 to 2005, the technical changes when employing the year 2000 as base year and the efficiency scores when employing the year 2005 as base year can be significantly different, especially when there are dramatic technology changes during the 5-year period. This creates a dependency on the base period's technology as a reference for measuring the technical changes, which might not be relevant to the periods that are under examination. On the other hand, this method fulfils transitivity, where the product of the index number of period-0 relative to period-1, and the index number of period-1 relative to period-2, equals the index number of period-0 relative to period-2 (Althin, 2001). Althin

(2001) finds that the change in efficiency from one period to another period differs significantly if the base period changes. This suggests that the measure of changes in productivity is dependent on the base-period technology and may not “truly” measure changes in productivity across the years. However, Fisher (1922) suggests that the circular test is not strictly necessary in order to obtain an accurate analysis.

The adjacent period method measures the shift in frontier at time t and t+1 for technology changes which are then averaged geometrically. This method does not fulfil the circular tests but it does not depend on a base period technology benchmark when measuring changes in technology.

Since the effects of the two methods are uncertain, both are considered in the study of the electricity line industry below. As the two methods provide similar results, only the base period Malmquist productivity index results will be reported below to study the changes in efficiency after the 1999 electricity reform.

3 Literature Review

The Malmquist productivity index has been used in a number of empirical studies to measure performance of industries such as the health sector (Giuffrida, 1999; Sommersguter-Reichmann, 2000), electricity industry (Edvardsen and Forsund, 2003; Klein and Yaisawarng, 1994), productivity in OECD countries (Fare, Grosskopf, Norris and Zhang, 1994), telecommunications (Uri, 2001), multi modal bus transit (Viton, 1998) and the banking industry (Asmild, Paradi, Affarwall and Schaffnit, 2004; Grifell-Tatje, Lovell and Pastor, 1998).

There are several studies on the effects of the introduction of new regulations and controls on productivity change in the electricity industry worldwide, using conventional DEA and the Malmquist productivity index. Klein and Yaisawarng (1994) studied the effects of sulphur dioxide controls on the productivity change in the U.S. electric power industry using the Malmquist input-based productivity index. They decomposed the index into change in technical efficiency, changes in technology and changes in scale efficiency and found that productivity had decreased from 1985 for three of the four target years.

Scully (1998) studied the effects on efficiency gains of the transformation of New Zealand electrical supply industry, from state-owned to commercially-oriented power companies. A translog cost function was used to measure the efficiency gains which indicate that the reforms had substantial cost reducing effects.

Lawrence’s (2003) analyses employed total factor productivity (TFP) analysis to look at changes in the industry productivity growth from 1996 to 2003. He used three outputs, namely throughput, system line capacity and connection numbers, and five inputs, i.e., operating costs, overhead line capacity, underground line capacity, transformer capacity and other capital. He found an average of 2.1% annual increase in the TFP growth trend during 1993 to 2003, and an increase of 5.6% during the year 1999 to 2000. Lawrence suggests that this change in the TFP index is due to the change in electricity regulations in the year 1999, leading to an improvement in the efficiency of the industry. However, given that the TFP index does not take into account the change in technology during the sampling periods, it cannot be certain whether the increase is due to efficiency or technology changes, or a combination of the two.

4 New Zealand Electricity Industry Data

The sample analysed in this study includes 29 New Zealand electricity line companies over the years 1999 to 2002 and 28 New Zealand ELBs in 2003. 15 of the ELBs operate in the North Island and 13 in the South Island. One company in the North Island was removed in 2003 because of the purchase of United Networks by Vector. The data was collected from the Commerce Commission (2003) and verified using the independent database established in Lawrence (2003) and data published in the PriceWaterHouseCoopers (2003) Electricity Line Business Information Disclosure Compendium for the years 2001 to 2003. In order to adjust for the effects of inflation, all monetary data, e.g. direct and indirect costs are adjusted to 2003-dollar value using the Customer Production Index (CPI)'s electricity group index.

The DEA model comprises three outputs: total electricity distributed, number of connections and a quality factor (SAIDI); and three inputs: direct and indirect costs, loss ratio and the optimised deprival value as a measure of the capital investment in the line businesses.

5 Results

The first DEA analysis pools the data over the five years with results reported in Table 1 showing an increasing trend with the exception of 2002.

	1999	*2000*	*2001*	*2002*	*2003*	*Average*
Average	82.4%	85.8%	87.6%	85.6%	88.4%	**86.0%**
Median	84.1%	86.0%	87.8%	84.2%	88.2%	**86.1%**

Table 1 – Pooled DEA efficiency scores

Table 2 reports the results of a Malmquist productivity index with 1999 as the base year with decompositions into changes in technical efficiency and technology. 21 (8) ELBs have efficiency changes greater (less) than 100% for the years 2000, 18 (11)ELBs for 2001, 19 (10) ELBs for 2002 and 10 (18) ELBs for the years 2001 and 2003. Aurora Energy shows a decline in technical efficiency over the entire period, with an average of 91.9%. Nelson Electricity and Network Tasman display improvements in efficiency with an average of 132.55% and 123.7% respectively.

There are 22 ELBs who have progressive technology changes (where $T\Delta > 1$) 1999 to 2000, 25 ELBs for 1999 to 2001, 22 ELBs for 1999 to 2002 and 25 ELBs for 1999 to 2003. ELBs such as Network Tasman, Orion New Zealand and Vector show considerable improvement in technology over the sampling period, with an average of 107.1%, 119.4% and 116.5% respectively. Other ELBs such as Nelson Electricity have considerable regressive technological changes (average of 86.3%). The variation for technology changes is significantly lower among the ELBs than for technical efficiency changes. Vector and

	Table 2	Malmquist productivity Index				Changes in Technical Efficiency				Technology Change			
		99-00	99-01	99-02	99-03	99-00	99-01	99-02	99-03	99-00	99-01	99-02	99-03
1	Alpine Energy	104.82%	108.48%	134.78%	126.12%	104.9%	103.6%	132.6%	114.4%	99.95%	104.76%	101.64%	110.20%
2	Aurora Energy (Dunedin Electricity)	96.14%	98.48%	98.85%	100.90%	95.1%	93.1%	88.2%	91.2%	101.14%	105.81%	112.03%	110.64%
3	Buller Electricity	89.18%	98.53%	109.14%	124.16%	94.1%	98.4%	112.4%	124.9%	94.82%	100.18%	97.06%	99.39%
4	Centralines	120.52%	112.34%	109.47%	112.12%	118.0%	107.8%	110.1%	99.3%	102.13%	104.19%	99.44%	112.88%
5	Counties Power	104.54%	99.94%	129.46%	96.68%	100.5%	93.4%	121.3%	78.5%	104.04%	106.98%	106.76%	123.09%
6	Eastland Network	98.92%	124.13%	110.40%	116.42%	101.5%	125.2%	109.3%	95.1%	97.47%	99.12%	100.98%	122.38%
7	Electra	102.90%	92.96%	102.17%	97.84%	98.8%	93.1%	102.0%	89.0%	104.19%	99.82%	100.14%	109.96%
8	Electricity Ashburton	118.91%	110.45%	117.81%	118.92%	118.1%	105.3%	109.7%	103.7%	100.68%	104.85%	107.41%	114.66%
9	Electricity Invercargill	134.15%	129.90%	95.23%	127.62%	132.0%	132.0%	100.7%	132.0%	101.64%	98.42%	94.53%	96.69%
10	Horizon Energy Distribution	106.68%	118.82%	107.46%	125.10%	102.1%	104.8%	94.9%	106.5%	104.48%	113.33%	113.19%	117.48%
11	MainPower New Zealand	105.56%	111.64%	115.49%	112.91%	97.4%	100.2%	107.1%	84.7%	108.33%	111.36%	107.86%	133.35%
12	Marlborough Lines	109.28%	101.04%	96.79%	99.70%	105.3%	95.4%	93.4%	76.5%	103.81%	105.96%	103.62%	130.39%
13	Nelson Electricity	111.97%	144.59%	93.21%	105.40%	135.6%	167.9%	100.0%	126.7%	82.56%	86.12%	93.21%	83.19%
14	Network Tasman	110.68%	135.05%	142.09%	142.78%	107.5%	130.3%	129.9%	127.1%	102.94%	103.64%	109.37%	112.32%
15	Network Waitaki	117.55%	121.32%	99.39%	90.99%	112.2%	108.4%	108.2%	74.8%	104.77%	111.96%	91.87%	121.71%
16	Northpower	103.40%	111.82%	109.55%	119.23%	95.4%	100.4%	101.8%	106.8%	108.34%	111.39%	107.65%	111.65%
17	Orion New Zealand	103.31%	100.56%	113.36%	99.02%	82.5%	74.6%	100.0%	95.0%	125.22%	134.72%	113.36%	104.23%
18	Otago Power	113.30%	117.34%	117.10%	120.56%	110.7%	113.4%	108.7%	95.8%	102.35%	103.46%	107.74%	125.82%
19	Powerco	100.97%	102.21%	94.99%	105.63%	100.2%	96.6%	89.3%	90.6%	100.77%	105.80%	106.39%	116.64%
20	Scanpower	110.68%	119.99%	100.93%	104.70%	114.7%	111.2%	116.4%	101.4%	96.48%	107.87%	86.72%	103.30%
21	The Lines Company	107.02%	114.15%	108.72%	90.92%	103.6%	107.2%	106.1%	74.3%	103.28%	106.50%	102.49%	122.44%
22	The Power Company	118.85%	119.92%	103.30%	110.38%	119.4%	116.8%	94.9%	99.8%	99.52%	102.68%	108.84%	110.61%
23	Top Energy	100.21%	106.44%	93.72%	78.06%	96.4%	97.2%	94.5%	68.6%	103.99%	109.46%	99.16%	113.87%
24	Unison Networks	105.87%	97.57%	102.53%	107.77%	101.2%	94.4%	99.2%	99.7%	104.63%	103.40%	103.33%	108.14%
25	UnitedNetworks	107.09%	112.22%	114.52%		100.0%	100.0%	100.0%		107.09%	112.22%	114.52%	
26	Vector	115.90%	119.25%	121.25%	109.40%	100.0%	100.0%	100.0%	100.0%	115.90%	119.25%	121.25%	109.4%
27	Waipa Networks	105.06%	103.44%	106.84%	97.19%	100.4%	97.7%	106.2%	87.4%	104.68%	105.86%	100.60%	111.2%
28	WEL Networks	103.21%	99.08%	100.55%	100.89%	100.2%	96.7%	98.5%	93.8%	103.02%	102.43%	102.10%	107.57%
29	Westpower	95.70%	105.19%	106.59%	90.28%	98.5%	100.9%	105.3%	72.3%	97.12%	104.26%	101.19%	124.84%
	Average	**107.7%**	**111.6%**	**108.8%**	**108.3%**	**105.0%**	**105.7%**	**104.9%**	**96.8%**	**102.9%**	**106.4%**	**103.9%**	**113.1%**
	Median	**105.9%**	**111.6%**	**107.5%**	**106.7%**	**101.2%**	**100.4%**	**102.0%**	**95.5%**	**103.0%**	**105.8%**	**103.3%**	**112.0%**
	Maximum	**134.2%**	**144.6%**	**142.1%**	**142.8%**	**135.6%**	**167.9%**	**132.6%**	**132.0%**	**125.2%**	**134.7%**	**121.2%**	**133.3%**
	Minimum	**89.2%**	**93.0%**	**93.2%**	**78.1%**	**82.5%**	**74.6%**	**88.2%**	**68.6%**	**82.6%**	**86.1%**	**86.7%**	**83.2%**
	Standard Deviation	**9.0%**	**12.2%**	**12.0%**	**14.2%**	**11.3%**	**16.8%**	**10.6%**	**17.2%**	**7.0%**	**8.1%**	**7.5%**	**10.6%**

Table 2: Malmquist Index Decomposed into Technical Efficiency and Technology Changes

United Networks have no changes in efficiency over the five-year period because in the calculation of the Malmquist index, these ELBs had infeasible super-efficiency scores for all five years and a proxy has been used to calculate the Malmquist Index.

Some interesting insights can be gained from the Malmquist productivity index. For Electricity Ashburton (Table 3) the pooled DEA analysis for 1999 to 2002 shows an average efficiency of 70.4%. However, in 2003, there is a significant increase in efficiency to 80.2 %. The pooled DEA results would attribute all the increase in its efficiency scores to that year. However, the Malmquist productivity index shows the significant increase is not caused by technical efficiency improvements, (which declines in 2003 although it is larger than 100%), but is due to significant technological improvement for 2003. (Recall that 1999 is the base year).

			1999-2000	***1999-2001***	***1999-2002***	***1999-2003***
Malmquist Index	**Tech Efficiency Changes**		118.1%	105.3%	109.7%	103.7%
	Technology Changes		100.7%	104.9%	107.4%	114.7%
		1999	**2000**	**2001**	**2002**	**2003**
Pooled DEA	**Efficiency Score Mean (70.4%)**	69.1%	75.3%	68.2%	69.0%	80.2%

Table 3 –Summary of Efficiency of the ELB, Electricity Ashburton

Overall, the results indicate a positive shift in technology between 1999 and 2003 with improvements in technical efficiency during 1999 to 2000, 2001 and 2002, but with a decline when comparing 1999 to 2003. In order to see whether these changes in technology and technical efficiency are caused by factors not analysed in the above models, further sensitivity analysis is carried out to investigate the variations in efficiency in the pooled analysis and the changes in technology and technical efficiency in the Malmquist approach. Factors investigated include capacity, percentage of lines underground and location differences.

5.2 Regression results

To evaluate possible relationships between efficiency and potential environmental factors, Table 4 reports the results of regressions of the pooled DEA and Malmquist DEA efficiency scores using the following explanatory variables:

TRANS_CAP = Total installed distribution transformer capacity, in kilovolt amperes

(TRANS_CAP)^2 = TRANS_CAP squared, in kilovolt amperes

TOT_SYS_LENG = Length of overhead line and underground cable measured in kilometres

UNDER = Percentage of lines underground (= total underground lines (km) divided by total system length (km))

For the pooled efficiency scores, YR_00, YR_01, YR_02, YR_03 = Dummy variable for 2000, 2001, 2002 and 2003

For the Malmquist measures, YR_9901, YR_9902, YR_9903 = Dummy variable for 1999 compared with 2001, 1999 with 2002 and 1999 with 2003

NORTH = 1 if line business is located in North island.

The pooled DEA efficiency scores regression models show transformer capacity, total system length, the percentage of lines underground and North Island location as significantly related to the efficiency scores in the pooled DEA model. In particular, as the percentage of underground lines increase, there is an increase in the efficiency scores. North Island ELBs are on average 23.2% more efficient than South Island ELBs possibly because there are more high density urban areas in the North Island. . Transformer capacity appears to have a detrimental effect on efficiency scores and the positive quadratic terms suggests a U-shaped relationship implying diseconomies associated with mid-sized ELBs.

	Pooled DEA Results		**Malmquist Technical Efficiency**	**Malmquist Technogical Change**
TRANS_CAP	-0.445*	**TRANS_CAP**	-0.397	-0.139
TRANS_CAP2	0.672***	**TRANS_CAP2**	0.441	0.067
TOT_SYS_LEN	-0.229**	**TOT_SYS_LEN**	-0.067	0.034
UNDER	0.385***	**UNDER**	-0.059	-0.004
YR_00	0.093	**YR_9901**	0.030	0.169
YR_01	0.144	**YR_9902**	0.010	0.057
YR_02	0.084	**YR_9903**	-0.222*	0.502***
YR_03	0.128			
NORTH	0.232***	**NORTH**	-0.233**	0.072
Adjusted R-Square	**0.242**		**0.056**	**0.135**
Significance	**<1% *** <5% ** <10% ***			

Table 4: Regression Results of Efficiency Scores and Explanatory Variables

For the Malmquist changes in technical efficiency, technical efficiency improvement is significantly lower between the year 1999 to 2003 than 1999 to 2000, 2001 and 2002. In addition North Island ELBs have lower improvement in technical efficiency than ELBs in the South Island (23.6%). In contrast, the only significant variable for technology change is between 1999 and 2003.

6 Conclusion

In contrast to the pooled DEA model, the Malmquist index shows both efficiency and technology changes over time. Overall, North Island ELBs perform better than the South Island ELBs confirming results reported by Banicevich (1998). However efficiency improved at a higher rate in the South Island than the North Island. Although the percentage of lines underground and capacity significantly affect the pooled DEA scores, no significant relationships were found with efficiency and technology changes over time. Total system length does not appear to have any significant impact on the efficiency of ELBs.

Omitted variables are always an issue, but the model has used combined inputs and outputs employed in most other DEA studies and the DEA analysis on the New Zealand Electricity Industry by Banicevich (1998). Nonetheless, managerial aspects of companies are also important factors which might influence the efficiency of the companies.

7 References

Althin, Rikard. 2001. *"*Measurement of Productivity changes: Two Malmquist Index Approaches", *Journal of Productivity Analysis*, 16.

Asmild, Mette, Joseph C. Paradi, Vanita Aggarwall and Claire Schaffnit. 2004. "Combining DEA Window Analysis with the Malmquist Index Approach in a Study of the Canadian Banking Industry"*, Journal of Productivity Analysis*, 21: 67-89.

Banicevich, Mark. 1998. "Efficiency, Ownership, and Economies of Scale in the New Zealand Electricity Distribution Industry: An Application of Data Envelopment Analysis", Unpublished Honours Dissertation, Department of Accounting and Finance, the University of Auckland.

Commerce Commission. 2003. *Regulation of Electricity Lines Businesses Targeted Control Regime Draft Decisions – Resetting the Price Path Threshold",* 5 September.

Debreu, G. 1951. *"The Coefficient of Resource Utilization"*, *Econometrica* **19**:3.

Desli, Evangelia and Subhash C. Ray. 1997. "Productivity growth, technical progress, and efficiency change in industrialised countries: comment", *The American Economic Review*; **87**: 5

Edvardsen, Dag Fjeld and Finn R Førsund. 2003. "International benchmarking of electricity distribution utilities"*,* Resource and Energy Economics, **25** (4).

Fare, Rolf, Shawna Grosskopf, Mary Norris and Zhongyang Zhang. 1994. "Productivity growth technical progress, and efficiency change in industrial countries", *American Economic Review*, March.

Farrell, M. J. 1957.*"The Measurement of Productive Efficiency"*, *Journal of the Royal Statistical Society* Series A, General, **120**: 253-281.

Fisher, I.. 1922. *The Making of Index Numbers: A Study of Their Varieties, Tests, and Reliability*, Boston: Houghton Mifflin Company.

Giuffrida, Antonio.1999. "Productivity and efficiency changes in primary care: a Malmquist index approach"*, Health Care Management Science* 2.

Grifell-Tatjé, E., C.A. Knox Lovell and J.T. Pastor. 1998. "A Quasi-Malmquist Productivity Index", *Journal of Productivity Analysis*, **10**: 7-20.

Grifell-Tatjé, E. and C.A. Knox Lovell. 1999. "A generalized Malmquist productivity index", *Top* **7** (1): 81-101.

Klein, J Douglass and Suthathip Yaisawarng. 1994. "The effects of sulphur dioxide controls on productivity change in the U.S. electric power industry"*, The Review of Economics and Statistics*, Cambridge: Aug, Vol. **76** (3)

Koopmans, T. C.1951. *Activity analysis of production and allocation*, Cowles Commission for research in economics monograph, 13, New York: Wiley.

Lawrence, Denis. 2003. *"Regulation of Electricity Lines Businesses, Analysis of Lines Business Performance – 1996-2003"*, Meyrick and Associates Pty Ltd.

Malmquist, S. 1953. "Index Numbers and Indifference Surfaces."*Trabajos de Estadistica* 4, 209–242

PriceWaterHouseCoopers. 2002. *Electricity Line Business 2001 Information Disclosure Compendium*, March.

PriceWaterHouseCoopers. 2003. *Electricity Line Business and Gas Pipeline Business 2002 Information Disclosure Compendium,* January.

PriceWaterHouseCoopers. 2003. *Electricity Line Business 2003 Information Disclosure Compendium"*, November.

Scully, Gerald, W. 1998. “Reform and efficiency gains in the New Zealand Electrical Supply Industry”, *Journal of Productivity Analysis*, 11: 133-147.

Sommersguter-Reichmann. 2000. “The impact of the Austrian hospital financing reform on hospital productivity: empirical evidence on efficiency and technology changes using a non-parametric input-based Malmquist approach”, *Health Care Management Science* **3**.

Uri, Noel. D.2001. “Productivity change, technical progress, and efficiency improvement in Telecommunications”, *Review of Industrial Organization* 18: 283-300.

Viton, Philip A. 1998. “Changes in multi-mode bus transit efficiency, 1988-1992”, *Transportation* 25: 1-21.

PuLP a Linear Programming Tool Kit for Python

Stu Mitchell
Department of Engineering Science
University of Auckland

Abstract

Mathematical Programming Languages such as AMPL and GAMS are widely used within operations research. These languages encourage students and practitioners to clearly state their models in an algebraic form. The model is then clearly readable and separated from instance data.

PuLP is a algebraic mathematical programming toolkit for the Python programming Language. PuLP was originally developed by Jean-Sebastien Roy (js@jeannot.org) and extended by Stuart Mitchell (s.mitchell@auckland.ac.nz). Users of PuLP can state their models clearly through the use of an algebra of mathematical programming specific objects including problems, variables and constraints. Currently, three different solvers (CPLEx, LpSolve and Coin-or) are supported. As Python is a full programming language users are not restricted to the limited functionality of AMPL and GAMS.

There are a number of algebraic mathematical programming toolkits for other languages including Concert (CPLEX) and FLOPC++ (COIN-OR). PuLP uses Python language constructs (list comprehensions and modules) to make the construction of mathematical programs much easier.

Daisy: A Detailed Simulation of Milk Collection

Geoff Leyland
Incremental Limited
Auckland

Abstract

The dairy industry is fundamental to the New Zealand economy. Collecting the milk from farms and transporting it to processing sites occupies one of the country's largest transport fleets. Daisy is a detailed simulation of milk collection and transportation that can be used to study existing collection operations and what-if scenarios over extended periods of time. Daisy is composed of a discrete-event simulation engine, a shortest- path algorithm, a heuristic route planner, domain-specific logic and a scripting language. Combined with detailed data on milk supply, the transport fleet, production requirements and a road map, Daisy becomes a powerful tool for studying collection operations and a decision support tool for making improvement decisions. This talk will give an overview of Daisy's design, cover some interesting aspects of its construction and a discussion of other scenarios where it could be used.

New Tools for Military Operations Analysis: Agent-Based Models Combined with Genetic Algorithms

Gregory C. McIntosh
Defence Technology Agency
Devonport Naval Base
Auckland
New Zealand
g.mcintosh@dta.mil.nz

Abstract

MANA (Map Aware Non-uniform Automata) is an agent-based model developed by the Operations Analysis group at Defence Technology Agency. It has been used for a number of studies including: modelling civil violence management, maritime surveillance, coastal patrols, and developing future land operating concepts for the NZ Army. MANA purposefully leaves out detailed physical attributes of the entities being modelled if this is not expected to have any bearing on the study at hand. Hence, the model runs relatively fast, allowing fitness landscapes to be built up over parameter sets within a reasonable time frame. Agents are ascribed a set of personality weightings to guide their behaviour on a battlefield map. These settings can change state, depending on the presence of other agents and terrain features on the map. Hence, a rich variety of behaviour can emerge from a seemingly straightforward scenario, allowing novel military solutions to be uncovered. The agent personality weightings lend themselves to being incorporated into a genetic algorithm (GA) scheme for evolving superior military tactics. We present such a GA scheme and an example of its use.

1 Introduction

Genetic algorithms have recently found acceptance as a viable tool for solving a variety of problems such as designing electronic circuits, automated software development and designing efficient communications networks. More generally, genetic algorithms can be used to solve problems requiring some type of optimization where a large number of parameters are involved and the mathematical structure of the fitness function is not well behaved or is unknown beforehand.

The genetic algorithm derives its inspiration from the way species are thought to evolve in nature. A species' physical and behavioural characteristics are encoded by their genes. The 'solution' for a particular species corresponds to those individuals who are fittest to survive in their environment. Our GA scheme is illustrated in Figure 1. In order to evolve a squad of agents to achieve dominance in a scenario, a population of equivalent squads is defined starting from a random gene pool. Each squad from the population is pitted against the scenario's enemy in turn to establish fitness values. Fittest squads are retained in the gene pool while less fit squads are allowed to fall by

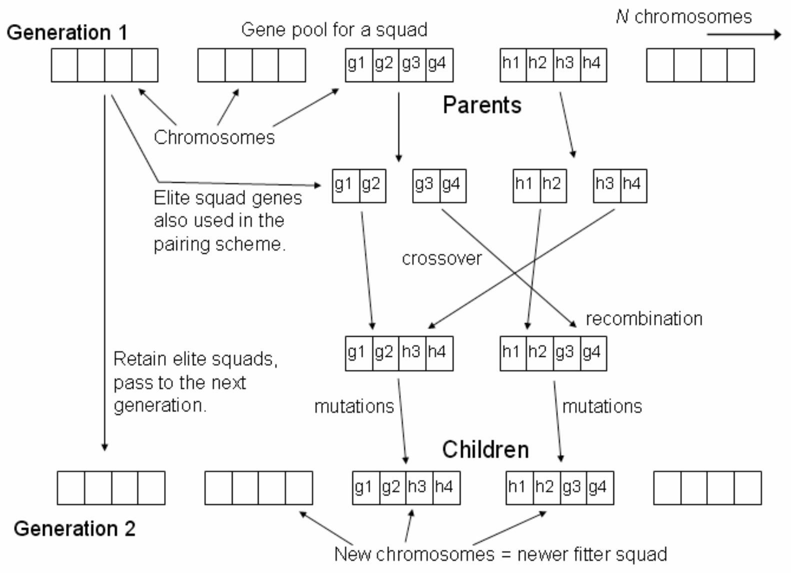


Figure 1. The genetic algorithm scheme, where $g_1,\ldots,g_4$ and $h_1,\ldots,h_4$ are genes corresponding to agent personality weightings in the MANA model.

the wayside. Analogous to evolutionary biology, recombination of chromosomes is carried out to determine the next generation of squads. Repeating the process yields a superior squad which optimally solves the scenario at hand. This typically occurs after just a few generations. The genes in our scheme are integer valued; in contrast to the binary genes found in evolutionary biology. Our scheme focuses on evolving clever tactics and behaviour with the military hardware already specified. Hence, the emphasis is on optimization of operating procedures, as opposed to, say, equipment procurement.

2 Example

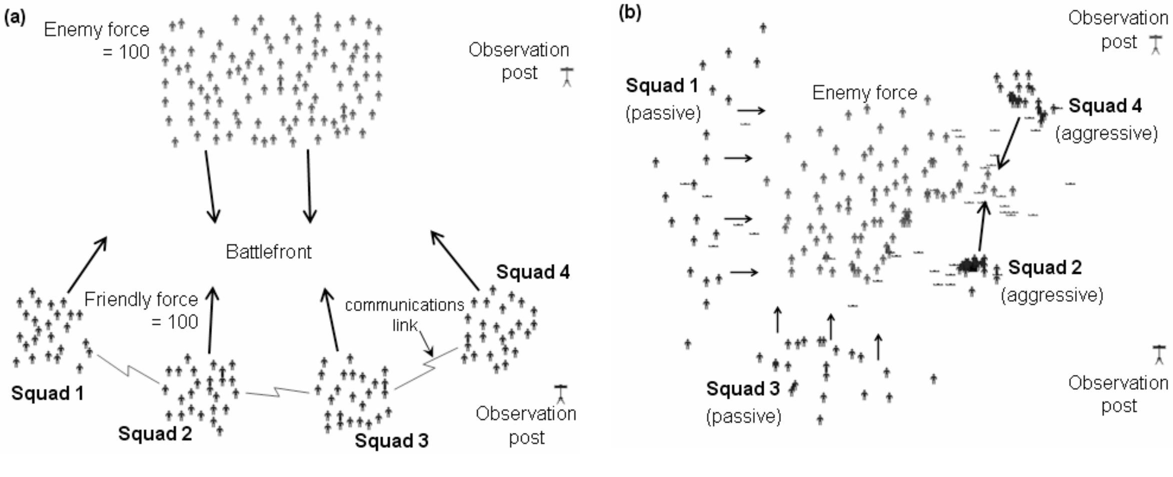


Figure 2. (a) Two-sided battle. Both sides are equivalent except the Friendly force has been split into 4 squads. (b) Tactics evolved by the Friendly force: flanking manoeuvre.

The example shown in Fig. 2 illustrates one particular scenario evolved using our GA scheme. Two armies of agents are pitted against each other. Both sides are equivalent except that the Friendly force has been split into 4 equal squads. Without some type of coordination amongst the four squads, the Friendly force is at a disadvantage since each squad by itself is outnumbered by the enemy force. The GA scheme has been applied to find tactics which can give an advantage to the Friendly force. Tactics evolved are shown in Fig. 2(b), and correspond to a standard flanking manoeuvre. Two squads adopt a more passive stance and maintain a stand-off distance from the enemy. This keeps the enemy occupied and fixed in place. Meanwhile, the other two squads adopt a more aggressive stance. They sweep around and, in a pincer-type movement, achieve local numerical superiority at the enemy's flank. This allows the Friendly force to gradually 'wear down' the enemy and gain the upper hand. Communication links between the four squads (which are modelled in MANA) were essential for this stategy to work.

3 Summary

The genetic algorithm performs extremely well and usually arrives at some sensible solution for outperforming a scenario's enemy. The GA tool often brings up unexpected solutions which may not have been intuitively obvious at the outset. Our GA tool usually converges to a solution within approximately 10 – 20 generations. For smaller scenarios, such as the one illustrated in this article, a population size of $N \sim 10$ has been found to be more than adequate. For larger scenarios with multiple squads involved and many personality weightings selected to build larger chromosomes, we have found $N \sim 20$ to be quite adequate for bringing up viable solutions. On the other hand, for $N > 100$ we begin to experience problems with the GA scheme not converging. In future, this type of software tool could be used for rapid course-of-action development in a front-line operations analysis environment.

4 References

Galligan, D.P., M.K. Lauren, 2003. "Operational Aspects of Imaging Radar Systems in Maritime Reconnaissance Aircraft", *Journal of Battlefield Technology* **6**(3): 29-32.

Graham-Rowe, D., 2002. "Radio emerges from the electronic soup", *New Scientist*, 31st August, p31.

Lauren, M.K., R.T. Stephen, 2002. "Map-aware non-uniform automata (MANA) a New Zealand approach to scenario modelling", *Journal of Battlefield Technology* **5**(1): 27-31.

McIntosh, G.C., 2006. "Genetic Algorithms Applied to Course of Action Development Using the MANA Agent-based Model", *Journal of Battlefield Technology* **9**(3):27-32.

Mitchell, M., 1996. *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Massachusetts, USA.

Yiu, S.Y., A.W. Gill and P. Shi, 2003. "Using Agent Based Distillations to Model Civil Violence Management", *Journal of Battlefield Technology* **6**(1): 27-32.

Operations Research in Practice Experiences from an OR Consultancy

Craig
MacLeodOrbit
SystemsNew
Zealand
craig.macleod@orbitsystems.co.nz

Abstract

Operations Research (OR) methods are not widely understood or well known in thebusiness community. So what constitutes a problem that an OR Based model mightsolve isn't always clear.

This paper works through some of the problems that might require an OR professional, the types of approaches that could be used and what the potential payoff'smay be. As OR practitioners, Orbit are equipped to produce excellent solutions to some of these problems.

Key words: optimisation, heuristic, knapsack

1 Introduction

Problem solving with mathematical models is extremely effective, as many ORproblems are numerical. Some simple questions will signal if an OR-based solution isrequired:

How difficult is this problem? Is getting any feasible solution going to bedifficult?

If the problem is solved without a model of some kind (e.g. by hand), how farwill the solution be from the best possible solution?

If the answer to the previous question is 'I'm not sure', what method would be

used to gain a better answer?When problems are difficult to solve, any feasible solution can seem acceptable. But diga little deeper and it often becomes clear problem solvers have no way of knowing how close to the best possible answer they're coming. And very often the person chargedwith solving problems is ill-equipped to improve on the quality of the solutions.Because:

The problem really is a difficult one to solve.

There's no time to work on a better technique because the current solutionprocess takes all available time.

The tools and techniques for creating a better solution are not available, or notwell understood by the problem solver.

2 What Makes it an OR Problem?

There are two basic reasons to undertake an OR modelling exercise:

1. A very big decision must be made. Whatever it is, it's going to cost a lot ofmoney and spending a certain amount of time figuring out if the best possible decision is being made is likely to be worth the effort. Even confirmation the correct course is being taken might be worth the development of a model.

2 The same type of decision is made repeatedly. In this case, people potentiallyspend a lot of time coming up with the answer, and even a small improvement in each result will make model development worthwhile.

2.1 The Big Decision

To know whether an OR model is needed, start by deciding the 'size of the prize'. Ask,'if we made the best possible decision, how much would that gain?' and conversely, 'ifa poor decision were made, how much would that cost?'

The best solution may be apparent, even if we're not sure how it's achieved -or ifit's possible -everyone knows the answer to the Rubik's Cube, even if they can't do it.So knowing where you are and how far from a solution you might be is usefulinformation.

If the solution is not known, it's a question of asking whether an answer that was 1%better than the current one justifies further study.

Finally, if the decision is big enough, with enough options, a good model is ofbenefit.

2.2 Repeated Decisions

Average behaviour is the key with repeated decisions; achieving the optimal outcomefor each decision is not necessary if we routinely make at least 'good' decisions. An example of this is eating. No one gets healthy by eating the right food for one day.Moreover people won't get unhealthy by eating junk food for one day, or several days ina row; as long as they make good choices most of the time.

Having a model can save planning time. If it takes a planner until 4:30pm to calculate the daily plan, a company may have been operating poorly most of the dayeven though they had information to enable them to do better at 9am. And what happens to the quality of those solutions when the planner, who knows what they're doing, is away?

Many decisions require large quantities of information to be retrieved, possibly fromdifferent systems; an answer therefore must have multiple objectives balanced againsteach other.

3 What Type of Model

So what type of models might be possible?

1 Direct optimisation. Though somewhat rare, problems do exist that can besolved directly via mathematical programming.

2 An optimisation hybrid. This type of model uses a math model with some typeof 'wrapper' to gain meaningful results.

3 A heuristic. The size and complexity of a problem may dictate using this method. It can be used in parallel with one of the other methods if there isuncertainty as to whether a solution is possible or not.

4 Optimisation Case Study – NZ Post

NZ Post approached Orbit with the goal of developing a model of their mail collection,sorting, transport, and delivery systems; NZ Post believed a mixed-integer programmingmodel would be ideal. The size of potential savings, estimated in the millions, warranted

4.2 Problem Size

Unsure whether the problem could be optimised, Orbit created a detailed formulation.This formulation determined the data required from NZ Post, as well as giving ananticipated model size. Our initial formulation indicated approximately 200,000 constraints. Since we'd solved problems much larger than this using GAMS andCPLEX on a 32 bit Windows PC, it was worth developing the model.

4.3 Exploding problem size

Without building an actual model it's difficult to test specifications and formulationsand errors or omissions can be made. For large MIP's, a seemingly small omission cancreate significant changes in the size of the model.

NZ Post standards for mail required Orbit to track the day of pickup for each type ofmail.

Our initial formulation assumed recipients of mail don’t need to know the source of that mail. The formulation overlooked the issue of across-town mail deliveries. NZ Post’s standard for mail delivery for certain types of mail is two days. However, acrosstown should only take one day. Therefore knowing the destination is not enough, withlocal mail, the origin is also significant.

This type of omission is common in problem specifications. Given a list of items,most people can easily pick ‘the odd man out’. But picking a missed item is extremelydifficult. Orbit’s formulation was correct as far as it went, but an important issue hadbeen excluded. Such omissions may remain undetected for an entire development process.

To implement this ‘new’ restriction, tracking both pickup date and source of mailwas necessary and each type of mail needed to be tagged as being across-town or not.

Many constraints have an exponential increase in size due to the effects of theseadditions. The initial estimation of 200,000 constraints rose to 850,000 for the fullmodel.

In combination with the variables and non-zero elements created, this problem wasabout 50% bigger than the maximum sized problem that could be attempted on a 32 bitWindows PC using CPLEX. Even if it fitted, the difficulty in solving an MIP meant themodel was probably twice as large as the biggest model that could be successfullysolved.

A seemingly small omission converted a tractable problem into an impossible one.

4.4 Problem Compression

While it may have been possible to reduce the timeframe of the model, the number ofnodes, or the number of mail types, Orbit felt this would make the model much lessuseful in terms of giving meaningful results. We therefore tackled the size of the modelon a number of fronts:

Limiting model options for clearly unlikely solutions.

Analysing the data to see whether knowledge about the inputs could help us.

• Being very careful not to create redundant constraints or variables.Major savings were possible by limiting the sorting options available for each node.The across-town requirement actually helped here, as many sorting options would notmake across-town delivery by the next day possible. If each node was limited to having2 or 3 possible sorting node options, what was a 20 * 20 problem size became a 20 * 3problem size. If the options are picked carefully, the optimal solution will not be excluded.

Analysing the data provided further savings. If a particular type of mail had not been picked up at a node at any time, then no constraints or variables needed to be generatedfor that type of mail for the pick-up period. These types of savings would normally bemade by the LP pre-processor, but the problem needed to be reduced to some feasiblesize before the pre-processor could act. The final problem size after compression wasapproximately 1/3 its original size.

4.5 Summary

Directly optimising a problem is an attractive option, especially if savings are likely tojustify finding the best possible solution, even if the effort required is quite large.

However, small omissions, or perhaps a desired increase in functionality, can have adramatic effect on the problem size. If sacrifices in detail are made to solve the problem,the usefulness of the results may be compromised.

Orbit solved NZ Posts problem, but it took considerable effort to make the problemsmall enough. Solve times were about 2 hours per scenario. Using a heuristic it mightpossibly have taken a few minutes and the answers may have been close to optimal;however, we wouldn’t have known how close to optimal and the potential savings peryear justified significant modelling effort. If further functionality had been required inthe model, it may have become unsolvable.

4.6 In Practice

As a result of the modelling exercise (of which out model was a part), with a relativelysmall investment NZ Post saved millions of dollars per year and:

Announced the consolidation of a number of sorting centres.

Purchased a number of automated sorting machines.

Introduced postal codes (local knowledge of postal areas reduces when sortingcentres become centralised).

5 Hybrid Case Study – CASm

CASm was created for consultants at CRA International to perform market poweranalyses in the USA under appendix A of the Federal Energy Regulatory Commission (FERC) Merger Policy Statement ('Orders 592 and 642'). Appendix A requires thecalculation of market concentration using the Herfindahl Hirschman Index (HHI) foraffected markets under different conditions. This model has been used successfully innumerous merger analyses filed with FERC.

Potential suppliers into any market must pass a 'delivered price test' by comparinggeneration plus transmission costs to a given market price. Any supplier who can deliverpower to the affected market within 5% of the market price is considered a valid competitor. The destination market and market price is pre-defined and not determinedby the model. Total competing generation is limited to the amount that can be deliveredinto the market given existing transmission limits and market prices.

5.1 CASM as a Linear Program (LP)

In its simplest form, this problem can be formulated as an LP. The 5% price thresholdcan be simulated when a large supplier in the destination market is able to supply at 5%above the market price. If the destination market also has a large demand, all potential suppliers will fill up available transmission capacity, and HHI's can be calculated by
looking at the owners of delivered electricity. The formulation for CASm is stated as[1]:

Minimise: cost for supplies at the destination market
subject to: supply cost at destination < system lambda + 5%, for all suppliers
supply < quantity available, for each supplier and tranche
supply + flows in = flows out + "demand", for each node
line flows are adjusted for losses, for all interconnections
line flows < available limit, for all interconnectionsThis LP is solved
for each destination market and for a number of load conditions such as peak, off-peak, shoulder etc. As the LP minimises cost, the cheapest tranches ofelectricity will be delivered to the market, up to the available transmission capacity.HHI's can be calculated by summing shares for the owners of each delivered tranche.

5.2 Economic versus Fair Transmission Allocation

Using this solution means the cheapest electricity is purchased to fill the demand, butmore expensive potential suppliers can be completely excluded from the market; whichis not the intention of FERC policy. All suppliers should be considered valid participants and gain access to the 'market'.

One way of achieving this is by notionally allocating portions of the transmission network to all 'economic' participants, meaning the cheapest suppliers are limited tohow much load they can supply given the transmission capacity limitations. Proratingtransmission capacity using a 'fairness' criterion is not easy to do using a mathematical program[2]. All suppliers will be explicitly ranked by the LP. Further, the problem is quitelikely to be highly degenerate, since prices are entered in cents and similar types ofstation are likely to have the same cost. This degeneracy can cause quite arbitrary-seeming results in the LP output.

This degeneracy had the possibility of making the LP approach to the modelunworkable. However, maintaining the delivered 5% threshold and easy solution of acomplicated network system had considerable appeal. Previously models had madeextreme restrictions; such as limiting suppliers to a single network path to enable asimple solution to be made. A less restricted model was likely to become a market-leader in the field.

Orbit thought it might be possible to restrict inputs to the transmission system byusing a proration system. If each supplier was limited to injecting a ‘fair’ amount ofelectricity, the LP could then solve the model and results could be generated in the sameway as the ‘economic’ allocation; the 5% delivery threshold could still be considered tohold.

These ‘fairness’ constraints could be considered in the same light as constraintsadded by a cutting plane algorithm for non-linear problems. Cut away ‘unfair’ suppliers,means arriving at a solution still feasible when compared to the original unconstrainedeconomic version of the model.

5.3 Conflict Between ‘Fair’ and ‘Economic’

Imposing rules in a marketplace can lead to undesirable outcomes, especially if rulesimpose severely uneconomic restrictions on market participants. This was the case with the proration algorithm developed for CASm. Some types of constraint would destroythe pricing information contained in the model invalidating our 5% price test.

However, testing the constraints for the ‘fair’ solution is actually reasonably easy.

If some suppliers don’t actually supply up to their allocated limit, we know thatsome of the limits calculated are too high. A cheaper supplier is taking some ofthe transmission capacity allocated in theory to someone else.

If the total amount supplied is less than what would be supplied in the economicversion of the model, we could have allocated more transmission capacity toanother supplier.

We thus have both a test criterion for ‘optimality’, and a stopping rule for an iterativealgorithm. Knowing this, we can test our algorithm against a variety of problems.Understanding the ‘size of the prize’ helps because the optimal solution may not beknown but at least we know how close we are to it.

5.4 LP versus Heuristic

This LP model was developed in only a matter of weeks while the transmissionprorating algorithm took months. Though it may have been impossible to solve this problem directly using mathematical programming, the relative effort required to build a heuristic suggested we should try very hard to find a mathematical formulation for thistype of problem.

In terms of execution time, the heuristic part of the model is hundreds of times fasterthan the mathematical model part. In practice, this is a common ratio. The heuristic ispurpose-built for a specific task, and can take advantage of the specific structure of theproblem. The LP solver must be generic enough to solve any type of linear program.Although mathematical programming solvers have made major gains in performanceover the years, a purpose-built algorithm will generally solve in a fraction of the time.Combining an LP solver with custom-built heuristics can be extremely successful ifcarefully managed.

5.5 CASm in Practice

The hybrid version of CASm has become a standard tool for Appendix A mergeranalyses in the USA. CRA has used the model, and it has generally been accepted in alarge number of major electricity sector mergers subject to FERC’s jurisdiction.

6 Heuristic Case Study – Vehicle Load Builder (VLB)

Orbit developed VLB to allocate palletised items to trailers for transport. The basicproblem was:

[1] This is the formulation format used in submission documents.

[2] It might be possible using a complementarity formulation

There were multiple trailer loads of items to be sent.

Each item had a priority that indicated the last day it could remain unsent.

Lower-priority items can be sent today, or delayed if it helps the current trailers.This need is traded off with the fact the item will eventually become a ‘must go’and may hinder future trailers.

Trailer rules; such as axle weight, item crush limits, load stability, and item

groupings exist and must all be maintained.From the outset, Orbit were certain this problem could not be solved via mathematicalprogramming in an acceptable time.

6.1 Load Feasibility Creates Problems

If we knew that items allocated to a given trailer would load feasibly, we could use a high-level mathematical programming approach related to the knapsack problem to solve these models. Existing products appear simply to implement weight and volumeconstraints for items allocated to each trailer. Problems with this approach:

Items that look like they might fit on a trailer simply may not. If all the items are very weak, they may not be able to stack high enough for the load to fit insidethe trailer without damage.

Lowering weight & volume guidelines to mitigate the first problem will lose theopportunity to load up to legal limits when the products would fit.

A trailer loaded with heavy items will likely ‘weigh out’ with plenty of sparevolume remaining. Similarly, a trailer loaded with light items will ‘cube’ outwith the weight constraint slack. Only a trailer that weights out and cubes outsimultaneously can be considered to be truly full.

Simple constraints and guidelines cannot capture enough of the restrictions to

reliably test the feasibility of a proposed load.For a company that ships thousands of trailers per day at around USD 1,500 per trailer,even small increases in shipping quantities per trailer are worth millions per year.

The requirement for balancing a large number of conflicting guidelines and legalrequirements within each trailer made a mathematical programming approach at worstimpossible, at best unwieldy and slow. Orbit believed a heuristic-based program would best solve this problem.

6.2 VLB Approach

Using a mathematical programming approach, could potentially solve for every trailersimultaneously. However, by not guaranteeing each trailer in the solution was feasible,the whole solution may have been invalid if any given trailer was infeasible. A heuristicmight solve a trailer at a time, possibly giving away a lot in terms of a global solution byfollowing locally promising solutions instead of the global optimum.

In our favour, however, is the fact that we know what the theoretical minimumnumber of trailers required is, since we know the total weight and total volume ofeverything that needs to be shipped. If our solution is close to this theoretical minimum,we can be confident that our heuristic is working well. If not, we know that further improvements may well be possible.

Our basic method to solve this problem was:

Create a group of items that we theorise will fill a trailer.

Test the trailer for feasibility using a detailed trailer builder which can check thisprecisely.

If the trailer is feasible, possibly attempt to add more to it.

If the trailer is not feasible, either remove some items, or create a new group of

items for testing.Where possible, Orbit developed multiple methods for solving each step, since alternateheuristics can sometimes perform better given special circumstances. Multiple methodstherefore give a better chance of achieving an acceptable outcome, even with difficultcombinations of products.

We know the theoretical best outcome, so we can see how well we do against it, and target development effort in the areas that are performing relatively poorly.

6.3 Additional Types of Problems

Additional difficulty levels in VLB include:

Layer fill – where pallets of items can be broken up into layers so that items willfit into gaps in the trailer.

Temperature classes – where certain items must be kept at particular temperatures, while others can travel in nearly any trailer.

Multiple orders – where groups of items must be kept together on trailers.

Live loading – where items are being produced in a known sequence from a

number of production lines and must be assigned to a trailer immediately.Using a mathematical programming approach one of these types of problems would have 'broken the camel's back'. A heuristic may get more complicated or slower, butwill seldom become impossible to modify.

6.4 Experience with VLB

Based on experience at a number of sites, VLB saves around 10% of trailers (or adds 10% more items per trailer). At a site shipping around 100 loads a day, savings are approximately 3 – 4 million dollars a year.

7 Experience for all Model Types

No matter what type of OR project is being undertaken some things hold true regardless:

7.1 Data

Collection and quality of data is always the biggest issue. Data is normally not available,or not available in the form required. If available, it often has major omissions or otherquality issues. An explicit budget should be created for data collection. On a fixed priceproject, serious consideration should be given to making data collection variable.

7.2 Forest for the Trees

Undertaking an OR modelling project may not be efficient. It might be better to buy afaster PC than to spend weeks making a slow model faster. Or buying a lighter fleet oftrucks would save more than building a model to increase shipping capacity.

7.3 The Budget

Any model can be improved with more time and effort spent on it. Therefore, there's nomodel a client would not like to see improved – as long as the budget remained thesame; so a logical stopping point must be found. This is more art than science whendeciding the right place to stop.

7.4 Get Something That Works

Having something that works is a lot better than having something potentially better thatdoesn't work. A poor model that's available when required is better than a detailedmodel that hasn't been completed. We can make something that's slow run faster, butit's better to get it running before we try for those improvements.

Summary and Conclusions

Approach	**Advantages**	**Disadvantages**

Optimisation	Find the best solution Generally quick to set up Complex trade-offsperformed implicitly by thesolver May uncover quiteunexpected solutions (thatare good ones)	Limited types of problem Limited size and complexity Small omission may causeproblem to becomeunsolvable May need to make majorsimplifications to gain feasibility Possibly slow run times
Hybrid	Balance between optimisation and heuristic	
Heuristic	Can find a solution to any problem No in-built limit to problemsize Additional functionalitycan be added Run times can be relativelyquick	Results may not be optimal,or even close Each heuristic must be created for the purpose Normally a lot more effortthan an optimisation

In practice, no model ends up being ‘pure’. Orbit’s NZ Post model had built in preprocessing heuristics to speed it up and allow for a solution. Our VLB software has a knapsack solver to solve some of the sub-problems to optimality. It’s a matter of balance and picking the best tools for the job.

Full Scale Scheduling, Pricing and Dispatch Software

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

Abstract

The New Zealand Electricity Market (NZEM) operates on a bid clearing system where market participants submit generation offers, purchase bids and reserve offers for electricity. The system matches purchase bids with generation offers to satisfy demand, while also ensuring enough reserve is scheduled to cover the risk of a major contingency event in the network.

Transpower, the operator of the national grid, schedule energy dispatch using a linear program model called the Scheduling, Pricing and Dispatch software (SPD). SPD matches generation offers with purchase bids at minimum cost, while ensuring network and reserve constraints in the system are satisfied.

This project involves building a full scale model of SPD currently implemented by Transpower. The project involves expanding on a current 18 node model of SPD to a full model containing all nodes throughout the country, and all transmission line and reserve constraints across the North and South Island.

This paper will outline the methodologies used in matching nodal prices through the network between the full scale model and actual market prices.

1 Introduction

The New Zealand Electricity Market (NZEM) operates by a system where generation offers from generating participants are taken together with purchase bids from purchasing participants. These bids and offers are then matched in such a way to minimize the cost of electricity generation while meeting demand in an optimal manner, while at the same time complying with transmission constraints and reserve requirements across the national network. Transpower, the national grid operator, determines an optimal dispatch and purchase plan using the Scheduling, Pricing and Dispatch software (SPD), which runs once for ever half hour block (trading periods).

1.1 Nodes

The NZEM is represented as a series of nodes spread out across the North Island and South Island which are connected by transmission lines. These nodes can be either both or one of the following:

- Grid Exit Points – a point in the network where electricity will flow out of the network to local networks and consumers.

- Grid Injection Points – a point in the network where electricity will flow into the network from generators.

A node being classed as a grid exit point indicates that there is demand for electricity at that point in the network. This demand exists in the form of separate purchase bids unique to specific electricity companies. A node being classed as a grid injection point indicates that there is generation of electricity at that point in the network. This generation of electricity exists in the form of separate generation offers unique to specific generation stations and generation units.

1.2 Generation Offers and Purchase Bids

Generation offers and purchase bids are divided into a series of tranches where each tranche has its own MW quantity and price. For generation offers, each tranche indicates a certain amount of electricity on offer at a certain price. Larger offers of electricity are offered at larger prices, creating an increasing piecewise linear stack.

Purchase bids exist as decreasing piecewise linear stacks. Each tranche represents a certain amount of electricity on demand, and the corresponding price that purchase bidder is willing to pay for this amount of electricity. Purchase bidders are willing to pay a higher price for their initial electricity requirements, and will sequentially set lower prices for their demand for subsequent electricity requirements. This creates the decreasing stack.

To satisfy the demand in the network, SPD matches generation offers with purchase bids at minimal cost. This results in the first tranches in each generation offer used first to satisfy the demand for the first tranches in each purchase bid. If a price for a generation tranche is too high, that energy is not cleared. If the price for a demand bid tranche is too low, that demand is not satisfied. The maximised objective function for the linear programme in SPD (only concerning generation offers and purchase bids) is of the following form:

$$NetBenefit = \sum_{p \in BIDS} \sum_{j=1}^{PurchaseBidBlocks_p} Purchase_{p,j} \times PurchaseBidPrice_{p.j}$$

$$- \sum_{g \in OFFERS} \sum_{j=1}^{GenerationOfferBlocks_g} Generation_{g,j} \times GenerationOfferPrice_{g.j}$$

Figure 1: Objective Function (part 1)

Note that if the purchase bid prices are too low, the demand will not be satisfied. Therefore prices for purchase bids are generally very large to encourage the satisfaction of demand.

1.3 Transmission Constraints

The objective function for SPD is maximised subject to network constraints in the model which must be satisfied. These constraints take into account the capacities of lines which connect nodes, voltage angles at each node in the network, and energy losses in each line.

Another important section of the transmission lines is the HVDC link connecting the North and South Islands. The link exists as a set of two poles. Electricity is allowed to flow either north of south through the link, and transmission losses and line capacities are also modelled in the link as constraints to the linear programme.

1.4 Risk and Reserve

An important part of the NZEM concerns accounting for the sudden loss of electricity supply to the network. This occurrence is known as a contingency event, and represents the tripping of a generator in the network, or the failure of one or both poles in the HVDC link. A contingency event will create a certain amount of electricity taken out of the system, known as *Risk*. This Risk must be restored as quickly as possible to maintain system stability. To ensure this is possible, energy is also dispatched in the form of reserve, and is put on schedule in case a contingency even occurs. Reserve is offered in the following forms (called Reserve Types):

- Partially Loaded Spinning Reserve (PLSR) - offered by generators.
- Tail Water Depressed Reserve (TWD) – offered by hydro generators.
- Interruptible Load (IL) – offered by purchase bidders in the form of energy purchased which can be sent back into the system.

These three types of reserve are offered in the form of fast instantaneous reserve (FIR) and sustained instantaneous reserve (SIR) (called Reserve Classes). FIR is offered in 6 seconds and SIR is offered in 60 seconds. Both these types of reserve are necessary to restore the system to a stable state after a contingency event.

Just like generation offers, reserve offers are offered in the form of tranches, where each tranche represents an amount of reserve being offered at a certain price. As the amount of reserve offered increases, so does the price for the reserve. Reserve dispatched is a cost to the system, therefore the objective function now becomes:

$$
\begin{aligned}
NetBenefit = & \sum_{p\in BIDS} \sum_{j=1}^{PurchaseBidBlocks_p} Purchase_{p,j} \times PurchaseBidPrice_{p.j} \\
& - \sum_{g\in OFFERS} \sum_{j=1}^{GenerationOfferBlocks_g} Generation_{g,j} \times GenerationOfferPrice_{g.j} \\
& - \sum_{r\in RESERVEOFFERS} \sum_{j=1}^{ReserveOfferBlocks_r} Reserve_{r,j} \times ReserveOfferPrice_{r.j}
\end{aligned}
$$

Figure 2: Objective Function (part 2)

Constraints in the model for reserve model restrictions on reserve dispatch based on generation and demand, and on the amount of risk in each island. Therefore each reserve offer must be linked with a generation offer (for PLSR or TWD) or purchase bid (for IL).

1.5 Prices

Through finding an optimal solution for a particular date and trading period, SPD will determine generation dispatch levels, purchased demand, transmission line flows and reserve schedules. A particular constraint in the network ensures the amount of electricity flowing into a node is equal to the amount flowing out of the node. When SPD is solved to optimality, the dual variables for this constraint correspond to the prices of electricity at each node within the network. These prices are a measure of how valuable electricity is at the corresponding nodes across the country.

1.6 Project Goals

PEDRO is an 18 node model of SPD which currently exits as a simplified version of the full SPD software without risk and reserve. The goal of this project is to expand PEDRO to a full scale version of SPD as used by Transpower with all transmission line and reserve constraints. This full version of SPD can then replace PEDRO and be used to perform testing, forecasting and benchmarking in other experiments involving the NZEM.

This paper will discuss the current processes undertaken in expanding PEDRO to a full scale model. Current nodal price outputs will be presented together with an outline of the next stages in calibrating the model to match the performance of SPD as used by Transpower.

2 Methodology

To build a full scale model of SPD, the current 18 node model will be expanded to include all nodal, transmission and participant data. Once the model is expanded, the same data set from PEDRO will be used for the Full SPD model and the results will be compared. Debugging and calibration will be performed until both models produce the same objective function value.

2.1 PEDRO

The 18 node model PEDRO is a simplified version of SPD written in the AMPL modelling language and uses the CPLEX 10.0.0 optimizer. The AMPL model file currently reads the data for the model from an AMPL text file data file.

Each node within the data for the model is a representation of a group of nodes in a certain section within the country. Similarly, the transmission lines connecting the nodes are a representation of a set of transmission lines in the full model. The model does not include risk and reserve.

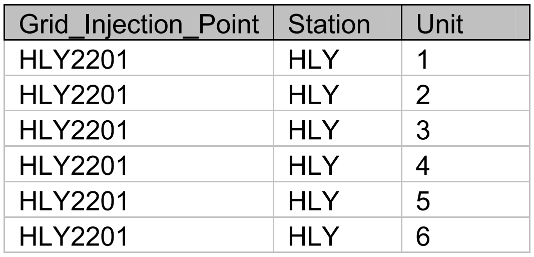
PEDRO matches generation offers with purchase bits at minimum cost subject to transmission constraints in the network. The system is modelled as a supply and demand network problem, where each node in the network has demand for electricity (from purchase bidders) and supply of electricity (from generation offerers). Demand is simplified to a fixed quantity of demand, and supply is broken down into offer stacks for multiple generation offers at each node.

2.2 Expanding PEDRO to Full Scale SPD

Expanding PEDRO to a full scale version of SPD involved redesigning the way data is imported into the model. The model will be required to solve over multiple dates and trading periods, therefore the data will continuously be changing. The data required for the model initially exists within the Centralised Data Set (CDS) as CSV files. The data within these files will be extracted and saved within an excel spreadsheet in a format which an AMPL run file can read using ODBC. The ampl run file will then loop over the number of dates and periods present within the excel data sheet and store the nodal prices and any other required information in a set which can then be outputted back to excel after SPD has solved over the time range. The manner in which the transmission lines, generation offers and purchase bids are modelled in PEDRO is different to that of the CSV files. Therefore as part of the process of expanding the model, the indexing systems for these participants will need to be changed to match the CSV files from the CDS.

2.3 Modelling of Generation Offers

The generation offers within the CDS are indexed by node (grid injection point), station and unit. A unique combination of these three indices represents a particular generation offer within the system. For example, the grid injection point HLY2201 contains 6 generation offers. These are indexed as follows:



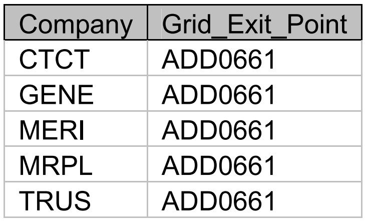
Grid_Injection_Point	Station	Unit
HLY2201	HLY	1
HLY2201	HLY	2
HLY2201	HLY	3
HLY2201	HLY	4
HLY2201	HLY	5
HLY2201	HLY	6

Figure 3: Generation Offers

Therefore at node HLY2201, we have 6 generation offer stacks. These 6 generation offers make up the supply of electricity for this particular node. Later it will be seen that reserve offers for PLSR and TWD reserve are also indexed by node, station and unit, therefore to match each PLSR and TWD reserve offer up with a generation offer, the generation offers need to be indexed in this way.

2.4 Modelling of Purchase Bids

The purchase bids within the CDS are indexed by node (grid exit point) and company. A unique combination of these two indices represents a particular purchase bid within the system. For example, the grid exit point HLY2201 contains 6 generation offers. These are indexed as follows:



Company	Grid_Exit_Point
CTCT	ADD0661
GENE	ADD0661
MERI	ADD0661
MRPL	ADD0661
TRUS	ADD0661

Figure 4: Purchase Bids

Therefore at node ADD0661, we have 5 purchase bid stacks. These 5 purchase bids make up the demand for electricity for this particular node. Later it will be seen that reserve offers for IL reserve are also indexed by node and company, therefore to match each IL reserve offer up with a purchase bid, the purchase bids need to be indexed in this way.

2.5 Modelling of Risk and Reserve

Data for reserve offers is supplied as reserve stacks for each reserve type and reserve class. These stacks are divided into tranches, and are supplied for each participant offering reserve. For PLSR and TWD reserve the participants are indexed by node, station and unit. For IL reserve the participants are indexed by node and company.

Risk is calculated for each island and for each reserve class. Within the SPD formulation, the risk is set equal to the maximum out of the dispatch for each generator and the flow through the HVDC link. If the risk is equal to the largest generating participant, or the flow through the HVDC link (if it is larger than the largest generating participant) than its will also cover the event of a smaller source of electricity failing, therefore this is the risk which the reserve dispatch must meet. The total amount of reserve dispatched across all reserve types must be equal to the risk in each island. This is modelled for both reserve classes within the SPD formulation.

2.6 Model Functionality

The model itself has been programmed in AMPL using CPLEX 10.0.0. However the formatting of the data for the model has been set up in excel using VBA macros. The AMPL run file uses ODBC to read tables containing data for the various sets for the model from the excel data sheet. The prices, generation and reserve dispatch, purchased demand and transmission line flows are then outputted back to the excel data sheet ready for analysis.

The process of running the full scale version of SPD begins within an excel Run Book. This Run Book exists within a directory which also contains the necessary data files needed for the model. This data currently needs to exist in the form of excel files for generation offers, purchase bids, branch data and loss information and reserve offers. Actual historical prices are also included within this data and will be read into the excel data book for comparisons with the prices outputted from the SPD model.

Within the Run Book the user must input the filenames of the data files, and must also ensure that the data contains the correct dates and trading periods. When the user runs the model, VBA macros will open each workbook and extract the required information. These macros also perform checks to ensure the data is valid, and will warn the user if errors are present. The macros link generation offers and purchase bids with the nodes within the model, and also link reserve offers with generation offers and purchase bids.

Once all the data is verified, it is stored within arrays in VBA, and then outputted to tables from the arrays within the excel Data Book. This is to reduce the computational time required for switching between multiple workbooks. When the data has been formatted into tables within excel, the AMPL run file is then called using a batch file run command, which starts the SPD optimisation model. The model will loop over each date and trading period present within the data, and then output the necessary information such as prices and dispatch data back to the excel Data Book. The prices from the model are then compared with the prices from the historical data for analysis.

To verify that the model was functioning correctly, the data used for the original PEDRO model was re-formatted from the text data files into the excel Data Book. Full SPD was then performed for this small data set and calibrating and debugging was performed until both PEDRO and Full SPD were giving the same solutions for the same sets of data.

3 Current Results

At this current stage in the project, the full model of SPD has been coded and VBA macros have been written to format all necessary data, however the reserve section of the model has been commented out until it is certain that the basic model without risk and reserve is functioning correctly.

Full SPD without reserve has been performed for a range of trading periods over two days, and the resulting nodal prices have been compared with the actual nodal prices. It was observed that there are currently major differences between the prices from the model and the actual prices for certain periods, however for other periods the prices match very closely. It is believed that trading periods between 12am and 4am of a day have a minimal reserve requirement. This is because it is a time where generation is low which results in small risks in both islands, and therefore small reserve requirements. Therefore the SPD model without reserve should be performing well for these periods, however several of these periods contain large errors.

The following graph compares the prices from the SPD Model and the prices from the CDS for period 2 for 1st February 2007, a trading period and date where the average absolute error was very small:

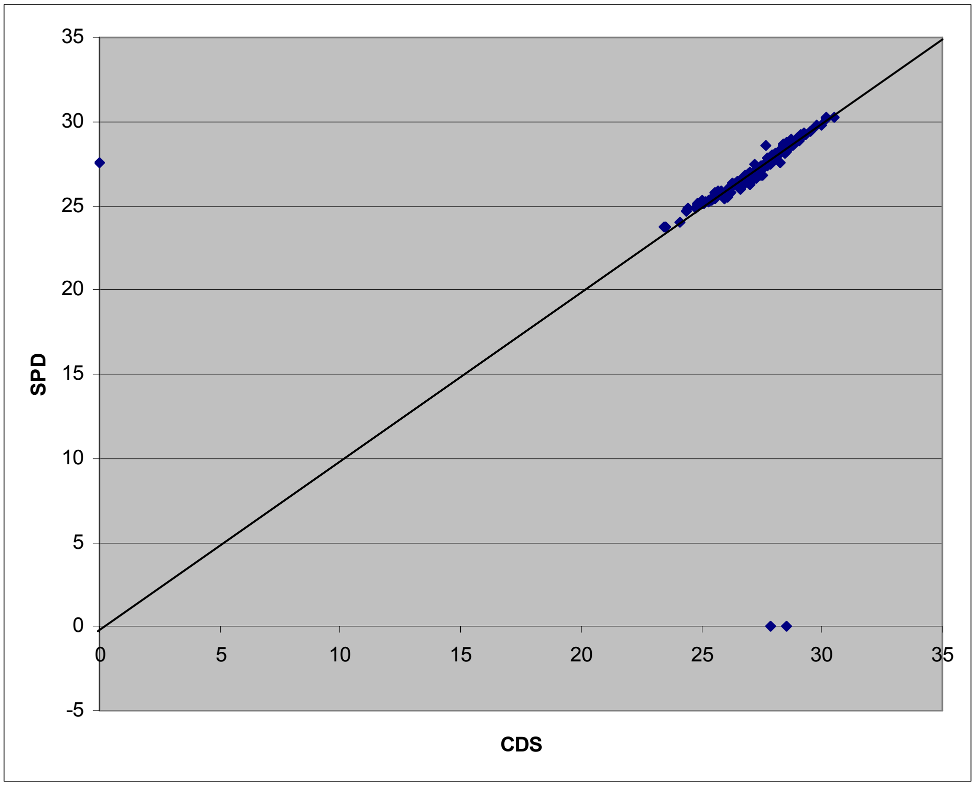


Figure 5: Price Comparison 1st February 2007

It can be seen that there are some outliers: nodes MNI0111, WES0331 and TWC2201. These nodes are in fact spur-lines, meaning they are nodes connected to only one other node in the network. It is possible that the optimal solution found by the SPD model has the same objective function value as the real solution found by Transpower, even though the optimal solutions are different. It is possible that this is the case here due to the fact that all the other nodes are very close, and the only differences involve a price of 0 for both sets of prices.

4 Next Stages

4.1 Debugging

The issues raised in the previous section concerning differences between prices between both sets of data need to be investigated to confirm that the full version of SPD without reserve is functioning correctly. This can be done by investigating the differences for periods where reserve isn't an issue (ie those periods between 12am and 4am of a given day).

Also an investigation needs to be performed for trading periods where the SPD model is performing well except for certain nodes. This will involve breaking the network down into small subsets surrounding particular nodes and observing line flows.

When it is verified that the Full SPD Model is performing correctly, the risk and reserve sections will be incorporated into the model. Comparisons can then be made for all other trading periods.

4.2 Introducing New Sections

Other sections which need to be introduced into the model include Mixed Constraints, Security Constraints and Ramping.

The Mixed Constraints section of the model which allows unique constraints to be applied to any variables within the formulation. Such constraints are used to determine parameters within the risk and reserve sections which are currently approximated.

The Security Constraints section introduces constraints to line flows for groups of transmission lines which ensure that system stability remains intact should a particular transmission line fails within the network. These are very important constraints which are always activated, and need to be incorporated into the full model.

Ramping is a section in the model which takes the generation dispatch from a certain trading period and uses it to constrain the generation dispatch for the next trading period. This section will only have an effect on the solution during peak times of electricity, and should have a minimal effect for trading periods early and late in the day. However to fully replicate SPD used by Transpower, this section will be included.

Loading Congestion at New Zealand Aluminium Smelters

Martin J. Peat
Department of Engineering Science
University of Auckland
New Zealand
mpea051@ec.auckland.ac.nz

Abstract

Rio Tinto Aluminium New Zealand Aluminium Smelters (NZAS) is located at Tiwai Point, Bluff. The smelter produces two main products, ingot and billet, of which 85% are exported by ship to Europe and Asia. Once products are produced they stored in a yard before being transported to the wharf using hired trucks and loaded onto ships.

The wharf is reached by a two lane road followed by a one lane causeway that has two passing bays to allow trucks to pass. The objective of the paper is to develop a simulation model to aid NZAS in determining the number of trucks to hire and also to understand the cause of congestion when loading vessels.

A discrete event simulation model is developed based on the sequence of events undertaken by a truck transporting product, utilising forklifts at the yard to load metal and cranes at the wharf to load the ship. The distributions of time taken to load, unload and drive a truck are estimated and will be implemented by NZAS once data is obtained. The validated model accurately simulates loading productivity allowing policy decisions to be made regarding the number of trucks to hire.

1 Introduction

New Zealand Aluminium Smelters Limited (NZAS) is one of the world's largest aluminium smelters, located on the Tiwai Peninsula, Bluff. Aluminium currently ranks seventh in New Zealand as a commodity export earner with NZAS ranking first as a single operating site. The economic benefit to the New Zealand economy is estimated at NZ$3.65 billion (Rio Tinto Aluminium 2007). High purity aluminium products are produced for use in the electronics and aerospace industry. After the smelting process, 86% of product is exported to markets in Japan, Europe and North America.

At NZAS raw alumina is imported, converted to aluminium, fabricated into metal product and exported for use. Molten aluminium is tapped from the bottom of reduction cells and cast into ingot, rolling block or extrusion billet. Ingot block is produced for remelt and is shipped in one tonne bundles. Standard purity ingot has 99.7% minimum aluminium content. Extrusion billet is cast to specific customer, capable of high speed, consistent extrusion in later processing (Rio Tinto Aluminium 2007).

2 Paper Background

Once produced, aluminium products are stored in an open yard adjacent to the smelter prior to shipping. At the storage yard trucks are loaded with either ingot or billet, with one forklift available at the yard for billet loading and one straddle carrier available for ingot loading.

Aluminium is loaded on vessels at the smelter's wharf, reached by a 1.3km two lane road followed by a 1.3km single lane causeway with two equally spaced passing bays, loaded trucks have right of way. The layout is shown in Figure 1. NZAS contracts trucks to deliver the aluminium to the vessels. Each month two different types of vessels are loaded with the three types of aluminium products. NZAS has provided comprehensive information regarding the loading of vessels (Goomes 2007).

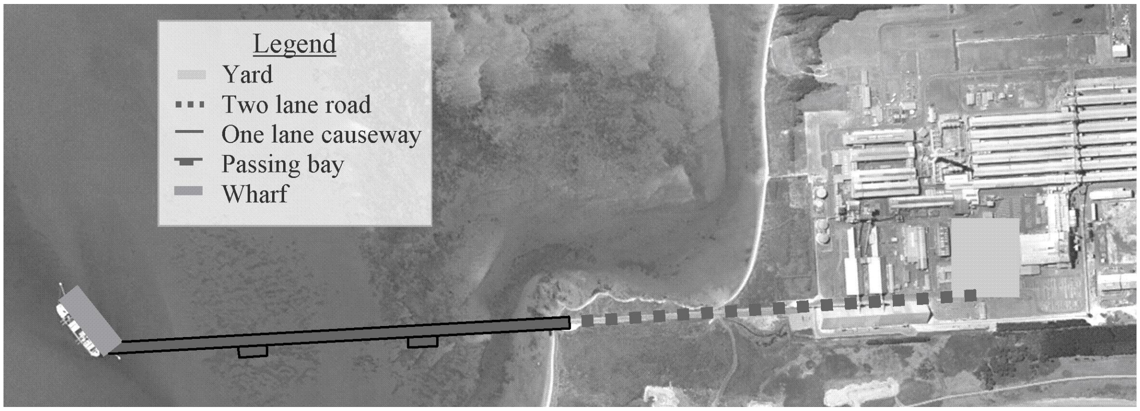


Figure 1: Schematic overlay of yard and wharf locations

Currently two types of vessel are used, gantry craned vessels and slewing craned vessels, with a crane being operated by a gang of workers known as stevedores. Trucks bring aluminium product alongside the vessel at the wharf to be lifted by crane into one of the vessels holds. The loading rates vary for each type of vessel and product type, with billet loaded faster than ingot. Loading is planned to ensure the use of both cranes is maximized at loading and unloading at the destination and also to keep the vessel balanced. The trim of a vessel and the port-starboard stability must be maintained and for these reasons the loading is professionally planned and executed to ensure the stability of the vessel and reduce any cargo movement at sea.

Standard operating procedure when loading vessels at the smelter wharf is to use 7 trucks when loading with two gangs on a gantry craned vessel or six trucks on slewing craned vessels. At times, as determined by a planned loading schedule, only one gang is used with four trucks transporting product.

It takes approximately one minute to unload a truck of ingot or billet at the vessel's side with the whole load being taken off in one lift. The unloading time is determined by how long the stevedores in the cargo hold take to stow each lift. Trucks then return to the yard along the same causeway.

3 Objectives

The paper objective is to produce an accurate simulation representing the traffic flow to the wharf whilst loading aluminium product. The simulation will be used to gain an understanding of the causes of congestion on the loading road by investigating the queuing behaviour and waiting times around the causeway.

The final objective is to produce a tool for NZAS to determine the number of trucks to hire for the loading of each shipment. The simulation tool produced should have the capacity to analyse the different loading configurations outlined by NZAS and produce output which allows the calculation of the optimum number of trucks to use under each configuration scenario.

4 Literature Review

A number of papers have been found which consider the loading and unloading of containers from vessels at a container terminal. Bish (Bish 2003) looks at multiple cranes serving each vessel with containers moved between the vessels and yard using a fleet of vehicles. The paper addresses issues including where to store containers in a yard, dispatching vehicles to pickup containers and the scheduling of loading and unloading operations using cranes on a wharf. Bish's model schedules the loading operations of the cranes to minimize the maximum time to load and unload a set of vessels using a transhipment based heuristic with a focus on easily implemented yet effective policies.

Everett (Everett 2001) applies analytical and simulation methods to the mining of iron ore from mine face through to vessel loading. The simulation is then used to aid scheduling decisions in loading shipments of iron ore. The quality of a shipment is dependent on successive shipments having uniform composition of raw material quality. Whilst system improvements are not reported, the graphical user interface is reported to have aided in the understanding of counterintuitive results obtained

Whilst each of these papers concerns vessel loading, the New Zealand Aluminium Smelter paper is unique in that it handles the transportation of product along a single lane road to be loaded onto vessels based on a pre-determined loading schedule.

Vehicle speed limits are used to ensure road safety. The speed at which drivers travel has been shown to be normally distributed and those drivers who travel at around the 85th percentile of a speed distribution are the safest in terms of accident risk (Ludmann, Neunzig et al. 1997). It has become a guideline for traffic engineers at the Department of Transport in Britain to set speed limits at the 85th percentile of natural driving speed along the given stretch of road (Association of British Drivers 2004).

5 Simulation Tools

A wide variety of simulation tools exist to provide a means of evaluating policies without having to experiment on a real system. The simulation software to be used needs to be easy to use, functional and low cost in order to fulfil the requirement of producing a tool for NZAS to use in determining the number of trucks to hire.

Python (Martelli 2003) is an open source, interpreted, object-oriented, high level programming language with an easy to learn syntax. The portability and free cost of Python made it an attractive option as it can be deployed on a wide range of platforms. The modular approach increases flexibility and allows the easy installation and use of packages, including the SimPy package.

SimPy is a process based discrete event simulation package based on Python. It provides the modeller with three parent class simulation components to model real world interactions between active processes which compete for limited resources. Monitor and tally objects assist in gathering statistics.

6 Developing the model

To formulate a simulation model representing the traffic flow of trucks to and from the wharf whilst loading aluminium product onto vessels, it is necessary to consider the real system in its entirety before making simplifications and developing a simulation in SimPy.

The real system to be evaluated encompasses all the actions involved in moving product already stored in the yard into the holds on vessels and thus ready for export. By isolating this part of the smelter operations, we are able to obtain information about the behaviour of the loading process. In implementing this model the time a truck spends waiting to load, waiting to travel along segments of causeway and waiting to unload are simulated. The accumulation of these waiting times gives an indication of the overall productivity of the vehicle fleet.

The loading configurations included in the model are based on information supplied by NZAS, whereby varying amounts of aluminium product are loaded onto vessels for export under changing loading scenarios. The loading scenarios of interest can be separated by decisions made affecting truck loading operations in the yard and those decisions regarding the vessel loading. Specifically, in the yard both products may be being loaded simultaneously, alternatively sole loading of ingot or billet may occur. The decision at the wharf is how many cranes are being operated.

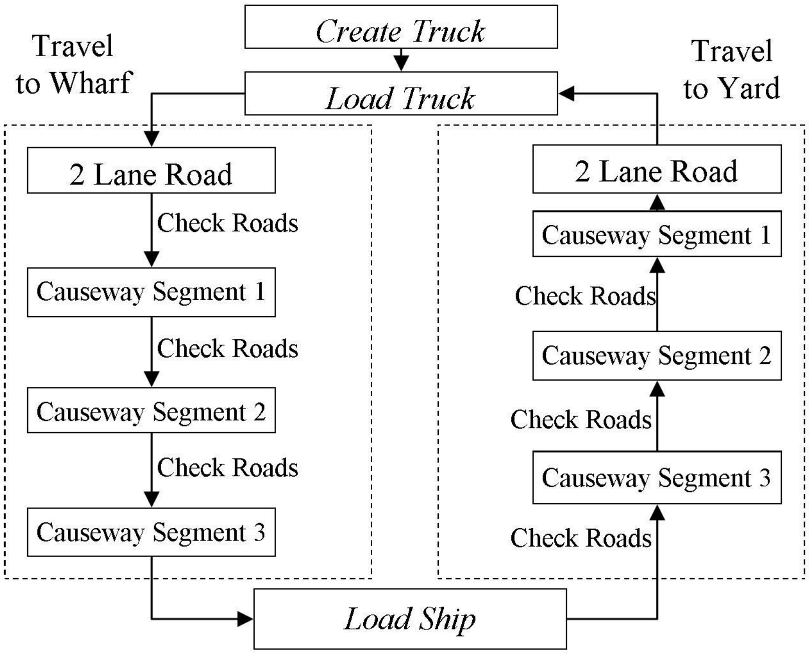


Figure 2: Logic flow diagram of the simulation model

6.1 Modelling

The formal simulation model utilises the classes of Python to cluster related model components. The model can be broken into three geographic parts:

- Yard. Including trucks waiting to load, selecting product and loading the trucks
- Roads. Travelling to and from the wharf along the two lane road and segments of the causeway
- Wharf. This is primarily concerned with loading vessel. Trucks wait for a crane to lift the aluminium product which is then stowed.

The model can be simply represented as a logic flow diagram (Figure 2) to be implemented in the simulation programme. The status of the roads ahead is checked as trucks come to the next segment of causeway.

6.2 Simulation programming

The model was built with increasing complexity as confidence in the Python language developed. The use of classes allows the model framework to be quickly adapted to model any of the loading scenarios at the wharf. A deterministic model was initially developed with the intention of introducing stochastic variables as a final stage of the development. To begin, the simplest loading scenario was implemented, with a truck being loaded and immediately unloading at one crane.

6.3 Single Road to Multiple Roads

To increase the complexity and realism of the model, a road is introduced between the yard and wharf, derived from a class collating the fundamental components including the distance, speed limit and the current usage of the road. Trucks are delayed for a period determined by the combination of the vehicle speed in relation to the speed limit, and the distance travelled. The acceleration and braking of trucks are assumed to be accounted for in the distribution of vehicle speeds. The physical capacity of how many trucks can use a segment of road is unknown and has not been constrained in the model but would be easy to add.

The step to implementing multiple roads proved simple due to the object oriented approach. To achieve this, road class instances are stored in a vector with each truck looping through the four instances of road to travel to the wharf and then completing the loop in the reverse order to simulate driving back to the yard.

6.5 Reducing Causeway Flow to One Lane

To accurately restrict the flow of traffic to one lane on the three segments of causeway a truck observes the state of the roads ahead to assess the traffic flow and determine whether to wait or travel. The implementation of a road component in the SimPy does not change, only the usage is restricted.

In the simulation model the give way logic has the effect of stopping a truck if the traffic flow ahead does not permit moving forward. The standard one way condition is that a truck must give way to any trucks exiting the closest segment of causeway, if this segment if unused or has traffic flowing in the same direction then the truck proceeds onto the segment of causeway. A truck returning to the yard must be able to reach and pull into the next passing bay if necessary before it proceeds to use the causeway.

6.6 Loading Schedule

Implementing a loading schedule is achieved by changing the availability of resources to load and unload trucks. A specific loading schedule for a given vessel is a combination of loading trucks at the yard in addition to a combination of cranes used to unload trucks at the wharf.

Thus by changing the availability of resources we are able to implement different schedules which can either stay constant throughout the simulation or change as a function of the simulation time. Implementing availability as a function of simulation time allows us to implement a model based on the supplied loading schedule.

7 Input Data

As no historical records were available, knowledgeable estimates made by the logistics team were obtained and will later be replaced by more accurate distributions when data recordings are made at NZAS.

Loading times at the yard are estimated using a lognormal distribution as realised values cannot fall below zero. The data provided for the simulation is used as the mean of our distribution with the loading time being a property of the forklift and crane resources.

To model the time taken to travel along a segment of road we refer to the research which found vehicle speeds to be normally distributed and the speed limit at the 85th percentile of speeds. Therefore we assume that the speeds of the truck fleet will be normally distributed, and furthermore that the speed travelled along each segment of road is independent to any other pieces of road and restrict speeds to positive values.

A similar approach is taken to model the time distribution when unloading trucks and stowing product in the vessel at the wharf. A Weibull distribution is used with the range of unloading times obtained from NZAS being used to estimate the shape and scale parameters. We also know that a truck will frequently wait to be unloaded as the crane is busy (Goomes 2007). Using this information in conjunction with the loading rates on which loading schedules are based we can approximate the distribution of loading times for each product and type of crane

8 Evaluation of the Model

The usefulness of a model is determined by the accuracy in which it represents reality and meets the objectives set prior to implementation. The performance of the model should be comparable to the real system at Tiwai Point to be useful for decision making. To test truck movement through the system we review the state of the system at each change, focusing on the actions performed by each truck to ensure that the process flow is correct.

The simulation can be evaluated in three of the five loading scenarios by comparing output to the information provided by NZAS. In each scenario we know the combination of cranes, the type of product being loaded and the time taken to load a given number of loads. To validate the model under each scenario we simulate 500 replications, terminating at the expected loading time. Accuracy is to be measured with regards to the total number of truck loads transported to the vessel.

Table 1: Comparison of Loading Productivity with Real System for Validation

	Expected	Observed Mean	Range (loads)
Ingot One Crane	98 loads	97.49 loads	80 - 118
Ingot Two Cranes	160 loads	153.39 loads	134 - 177
Billet One Crane	119 loads	115.54 loads	91 - 134

Reviewing the summary of simulated loading productivity in Table 1 we find that the mean simulated number of truck loads is lies close to the expected value and is central to the range of loads observed in each scenario. In this respect the model can be considered valid.

In reviewing the sensitivity of road speed and unload times it was found that uncertainty in unload times has a greater impact on the mean trip time. Although the trip times in the standard model are observed to be almost double the expected time, the primary paper objective is to compare the effect of varying the number of trucks used for given scenarios. Thus as the trip times are longer across all scenarios observed, the model may be valid with respect to trip times in a relative sense, even though the absolute magnitude varies from what is encountered in the real world system.

8.1 Testing Policy Implications

The objective leads us to consider the effect of changing the number of trucks as a method of validating our model. We know that eight trucks were previously used to load at two cranes but this was decreased due to commercial reasons concerning the productivity of an eighth truck. The model behaves as expected when the size of the truck fleet is increased from six trucks to eight trucks. The cumulative time the fleet of trucks spends waiting increases because there are more trucks to service at the yard when loading and at the wharf when unloading. Interestingly a change in policy between six, seven or eight trucks does not yield a significant increase in the mean number of loads transported to the vessel.

9 Results

Performance of the simulation system is gauged in terms of the paper objectives, specifically the causes of congestion and determining the number of trucks for NZAS to hire. Measures of efficiency, including the number of loads completed in a fixed simulation time, or conversely the time taken to complete a fixed number of loads are used to produce a guide to the productivity of different size truck fleets.

Table 2: Four simulated loading configurations

	Number of Cranes	
Product	1 Crane	2 Cranes
Ingot	Case 1	Case 2
Billet	Case 3	Case 4

9.1 Productivity

Completing multiple simulation runs of 1000 minutes we are able to gauge the level of productivity of trucks by recording the average number of loads completed, as shown in Figure 3. If there was no delay caused by congestion we would expect a linear increase in productivity.

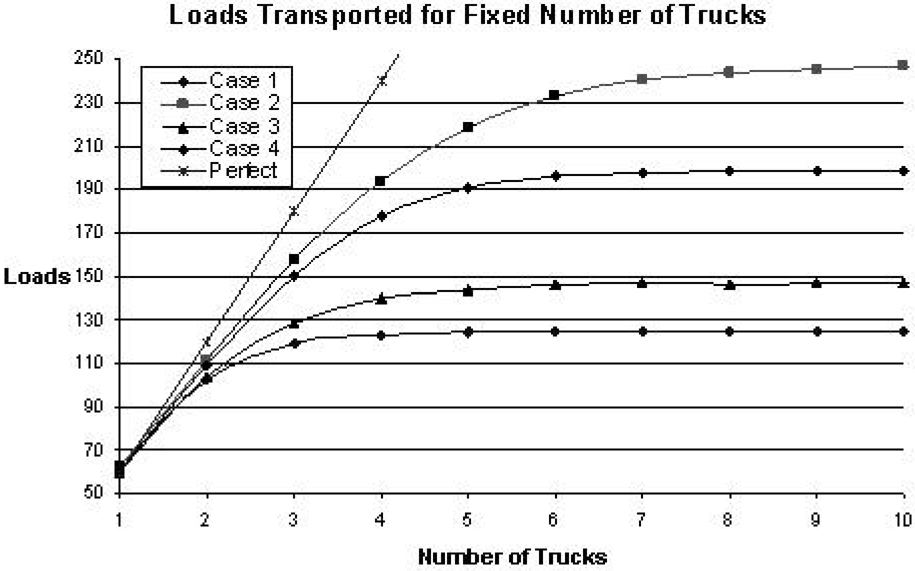


Figure 3: Loads transported in a fixed time with different numbers of trucks

As the gradient of the plotted lines decreases to a plateau the productivity decreases until no improvement is seen when another truck is added. This shows that there is congestion in the system and no benefit will be gained by adding another truck to the system.

9.2 Congestion

There is potential for congestion to be caused at a number of places in the system where decisions are made. To quantify how frequently and for how long a truck waits, the queue length and times at these places were investigated under each loading case.

As it takes longer for the crane to stow product than it takes for a truck to be loaded we find that trucks become phased, and little waiting happens at the yard. In our model trucks are found to seldom wait for a significant length of time to use segments of the causeway.

The majority of congestion in our model is observed to be at the wharf where trucks are unloaded and product stowed on the vessel. Queuing times are observed to be on average significantly higher than waiting times at the roads with trucks waiting an average of 10.5 minutes to unload. There is a queue to unload approximately 87% of the time.

The results obtained from the simulation show that most congestion occurs at the wharf where there are normally two trucks waiting to be unloaded. These results hold for each of the four loading cases and any number of trucks. The magnitude of the waiting time increases with an increase in the number of trucks hired.

10 Altered Models

Out of interest the model is altered to represent possible changes to the transportation of product and loading of vessels. We specifically consider scenarios where:

- The causeway is two lanes
- There are no passing bays on the one lane causeway
- Two forklifts or cranes are used for loading trucks in the yard

To implement the scenario with a two lane road the entire distance, the one lane causeway is removed and the distance of the two-lane road increased. Removing both passing bays from the one lane causeway simply requires a change in the give way policy so that all segments of the causeway have no oncoming traffic.

We find no significant difference in productivity under any of the new scenarios when loading ingot using two cranes at the wharf. However we find an amplified fluctuating pattern when no passing bays are present and a damped cycle with two cranes. This confirms earlier predictions that the fluctuating pattern in productivity was due to the average time spent in the yard is less than the average time spent at the wharf.

11 Future Development

Future development of the simulation model requires improvements in the accuracy of the distributions used for loading, unloading and vehicle speeds. This is to be implemented by NZAS once data is gathered for the purpose.

In addition, separating the processes of lifting product off a truck and stowing product in the vessel should lead to trip times which are closer to the fifteen minute estimation. Extending the complexity of the one way policy to give trucks more foresight would also aid in more accurately determining where congestion occurs on the causeway as we have only shown that the most congestion comes from waiting for the crane to finish stowing product on the vessel.

12 Conclusions

The simulation developed in SimPy to model the flow of traffic during the loading of vessels accurately predicts the productivity under the four loading cases derived from the loading schedule provided by NZAS. The validated model appears robust enough to be a useful tool in determining how many trucks to hire. The number of trucks to hire will also be influenced by a trade-off in cost between the costs of hiring trucks and the time constraints on loading the vessel. Depending on costs, the current NZAS policy of using seven trucks to load with two cranes or four trucks to load with one crane is near optimal.

The results found by terminating the simulation at either a fixed number of loads, or a fixed amount of time, provide the same information in a different manner and both may be of use to NZAS. If there are a set number of loads to be taken to the ship, the tool can be used to determine how long it will take to load with a given number of vehicles. Conversely, if the objective is to load as much as possible in a set amount of time then the termination criteria should reflect that.

Queue lengths and waiting times obtained from the simulation show that trucks seldom wait for oncoming trucks to pass, with returning trucks using passing bays effectively. We have found that there is very little congestion on the one way road although the model could be improved in this regard.

Instead congestion occurs on the wharf while waiting to load the ship. The length of time taken to stow product on the vessel was found to be the cause of congestion in the model as trucks were on average required to wait on the wharf for longer than they waited at any other point during a trip. The congestion on the one lane road was compared to a hypothetical model where a two lane road is present with no significant increase in productivity, confirming that congestion is created during the unloading of trucks.

As the number of trucks hired increases, congestion also increases and truck efficiency is observed to decrease. Further modelling of the unloading of trucks and stowage of product on the ship will improve the accuracy of our model.

The model achieves the objective of allowing NZAS to calculate the optimum number of trucks to hire under each loading scenario.

Acknowledgements

I would like to thank my supervisors Professor David Ryan and Dr Stuart Mitchell for their guidance and input throughout the paper. I would also like to acknowledge the assistance of Victor Goomes and Rio Tinto Aluminium New Zealand Aluminium Smelters Ltd for providing information for the paper and answering many questions regarding the loading process.

References

Association of British Drivers. (2004). "Speed Limits." Retrieved 25 August 2007, from http://www.abd.org.uk/speed_limits_85th.htm.

Bish, E. K. (2003). "A multiple-crane-constrained scheduling problem in a container terminal." European Journal of Operations Research **144**(1): 83-107.

Everett, J. E. (2001). "Iron ore production scheduling to improve product quality." European Journal of Operations Research **129**(2): 355-361.

Goomes, V. (2007). Wharf Congestion Project. S. Mitchell, M. Peat and D. Ryan, Rio Tinto Aluminium New Zealand Aluminium Smelter.

Ludmann, J., D. Neunzig, et al. (1997). "Traffic Simulation with Consideration of Driver Models, Theory and Examples." International Journal of Vehicle Mechanics and Mobility **25**(5): 491-516.

Martelli, A. (2003). Python in a nutshell, Sebastopol, Calif. ; Farnham : O'Reilly.

Rio Tinto Aluminium. (2007). "Rio Tinto Aluminium – New Zealand Aluminium Smelter." Retrieved 1 August 2007, from http://www.comalco.com/31_new_zealand_aluminium_smelter.asp.

Rio Tinto Aluminium. (2007). "Rio Tinto Aluminium – New Zealand Aluminium Smelter - Extrusion billet." Retrieved 1 August 2007, from http://www.comalco.com/whoweare/30_507.asp.

Clutha Hydro Electric River Chain Optimisation

Nicholas J. de Pont
Department of Engineering Science
University of Auckland
New Zealand
ndep003@ec.auckland.ac.nz

Abstract

The Clutha River has two hydro stations, as well as a storage lake. The Clyde station has a capacity of 432MW from 4 units. The Roxburgh station has a capacity of 320 MW from 8 units. There is also a major storage lake, the Hawea which has a controlled release that is subject to resource consents. This presentation will detail the methods used to model this system over a week with the objective of optimizing generator revenue. The model must represent the system realistically, thus it takes into account the delays when releasing water from one lake to when the water arrives at its destination, as well as the limited storage in the lakes and restriction of flow along the rivers. It must also model each turbine, and determine which units must be turned on or off. The model also looks at reserve currently setting reserve targets which must be met in each trading period.

1 The Clutha

1.1 Layout

The Clutha system has a main storage lake at Hawea, water leaving Hawea flows down to Lake Dunstan, and takes approximately 9 hours to arrive. Lake Dunstan is a small storage lake which connects to the Clyde power station. Water released at Clyde takes 2-3 hours to reach Lake Roxburgh. Lake Roxburgh is a storage lake which supplies water to the Roxburgh Power station; although this station is on one dam it connects to two separate nodes on the electricity network and thus is treated as separate power stations. Figure 1 on the following page shows a diagram of the Clutha system.

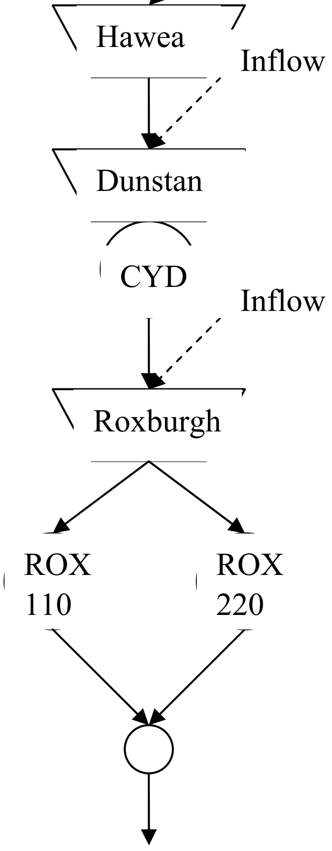


Figure 1: The Clutha System

1.2 Power Stations

As stated previously there are two power stations in the Clutha system the Clyde station and the Roxburgh Station. The Clyde station consists of four turbines and has a maximum capacity of 432 MW. The Roxburgh station has a total of eight turbines and a maximum capacity of 320 MW. However the Roxburgh station connects to two separate nodes, 3 turbines connect to one node, ROX110, and 5 turbines connect to another node, ROX220. All of the turbines are the same so therefore ROX110 has a capacity of 120MW, and ROX220 has a capacity of 200MW. The fact they connect to different nodes is important as the price separation can occur where the price at one node is not equal to the price at another node.

2 The Model

2.1 Objective

The aim of this project is to build a model which maximises revenue over one week. Revenue can be calculated as the profit minus the cost, where the profit is equal to the amount generated multiplied by the price which it was sold for, and the amount of reserve sold, multiplied by the price of that reserve. There are two costs associated with the system, the main cost is the cost of turning turbines on, and there is also a cost of running a turbine on tail water depressed. A final term had to be added which is a water value at Lake Hawea, in order to determine how much water should be released over a week.

Indices

j = stations: Clyde, Roxburgh110, Roxburgh220

t = Period: 1...336

Parameters

P_{jt} = electricity price at station j time t

R_t = reserve price at time t

C_j = cost of turning on a turbine at station j

T_j = cost of running a turbine tailwater depressed at station j

W = water value at Hawea

Decision variables

p_{jt} = generation at station j at time t

r_t = reserve offered at time t

F = final lake level at Hawea

$u^{+}_{j,t}$ = number of turbines turned on at station j at time t

$\lambda_{j,t}$ = number of turbines used for tail water depressed at station j at time t

Model CHERCO

1) Maximise $\sum_j \sum^{336}_{t=1} p_{ij}.P_{i,j}. + \sum^{336}_{t=1} R_t.r_{,t}. + W.L - \sum_j \sum^{336}_{t=1} u^{+}_{j,t}.C_j - \sum_j \sum^{336}_{t=1} T_j.\lambda_{j,t}$

2.2 Flow Constraints

The flow along the river is constrained by resource consents this section will cover these constraints, and also the balance constraints on the river and lakes. It is worth noting for this section that the model for the most part wraps around, thus water released from Hawea at the end will arrive at Dunstan at the beginning. Also the lake levels must return to their initial level; however Hawea is excluded from this specific constraint, due to the water value.

Indices

i = lakes: Hawea, Dunstan, Roxburgh

j = stations: Clyde, Roxburgh110, Roxburgh220

k = stations, lakes and junctions

t = period: 1...336

Parameters

$y0_t$ = initial lake storage at lake i at time t

I_i = inflow into lake i at time t

U_{ij} = upper bound of flow along arc i, j

L_{ij} = lower bound of flow along arc i, j

UB_i = upper bound of storage in lake i

LB_i = lower bound of storage in lake i

τ_{ij} = the delay from node i to node j

Decision variables

y_{it} = lake storage at lake i at time t

x_{ijt} = flow along arc i,j at time t

Model CHERCO

1) $y_{it}. = y_{it-1} + I_i + x_{jit\text{-}\tau_{ij}} - x_{ikt}$ Where ijk represent all stations, lakes and junctions

2) $y_{i,0} = y0_i$ for *i* in lakes
3) $\sum_{ji} \mathrm{x}_{jit\text{-}\tau_{ij}} = \sum_{ik} \mathrm{x}_{ikt}$ for all *t*
4) $L_{ij} \leq x_{ijt} \leq U_{ij}$ for all *t*
5) $LB_i \leq y_{it} \leq UB_i$ for all *t*

Explanation

1) This balances the storage in each lake
2) Sets the initial lake level
3) Balances the flows along the arcs
4) Ensures that the flows along each arc do not exceed the resource consents
5) Ensures that the lake level stays within the resource consent boundaries

2.3 Hawea Release Constraints

There is a specific set of constraints related to the flow from Lake Hawea. Resource consents only allow that the flow be changed by a certain amount depending on the current flow.

Indices

a = the different states for increasing flow rate
b = =the different states for decreasing flow rate
t = period: 1...336

Parameters

Q_b^- = amount flow can be decreased in state *b*
Q_a^+ = amount flow can be increased in state *a*
lob_a^+ = lower bounds of state *a*
lob_b^- = lower bounds of state *b*
upb_a^+ = upper bounds of state *a*
upb_b^- = upper bounds of state *b*

Decision variables

x_t = flow out of Hawea at time *t*
$\partial_{b,t}^-$ = integer variable determining decreasing flow state
$\partial_{a,t}^+$ = integer variable determining increasing flow state

Model CHERCO

1) $\sum_{a \in A} \partial_{a,t}^+ = 1$

2) $x_t \leq x_{t-1} + \sum_{a \in A} \partial_{a,t}^+ . Q_a^+$

3) $\sum_{a \in A} \partial_{a,t}^+ . lob_a^+ \leq x_t \leq \sum_{a \in A} \partial_{a,t}^+ . upb_a^+$

4) $\sum_{b \in B} \partial_{b,t}^- = 1$

5) $x_t \geq x_{t-1} - \sum_{b \in B} \partial_{b,t}^- . Q_b^-$

6) $\sum_{b \in B} \partial_{b,t}^- . lob_b^- \leq x_t \leq \sum_{b \in B} \partial_{b,t}^- . upb_b^-$

Explanation

1) Ensures that it is only in one state for increasing flow
2) Sets the limit which flow can be increased by
3) Determines which state the model is in for increasing flow
4) Ensures that it is only in one state for decreasing flow
5) Sets the limit which flow can be decreased by
6) Determines which state the model is in for decreasing flow

2.4 Unit Commitment

The unit commitment determines how many turbines are switched on and how much power is generated. This generation is a function of the flow through a turbine; this is estimated using linear piecewise functions.

Indices

i = lakes: Dunstan, Roxburgh
j = stations: Clyde, Roxburgh110, Roxburgh220
m = pieces: 0,1,2
t = period: 1...336

Parameters

$\varphi_{j,m}$ = the width of piece m for station j
a_j = the y intercept of the turbine curve at station j
$cc_{j,m}$ = the gradient of piece m at station j

Decision variables

$z_{j,t}$ = the number of turbines running at station j at time t
$u^{+}_{j,t}$ = the number of turbines turned on at station j at time t
$u^{-}_{j,t}$ = the number of turbines turned off at station j at time t
$qcc_{j,m,t}$ =the amount of flow through piece m in station j at time t
$p_{j,t}$ = the generation at station j at time t

Model CHERCO

1) $z_{j,t} = z_{j,t-1} + u^{+}_{j,t} - u^{-}_{j,t}$

2) $qcc_{j,m,t} \leq \varphi_{j,m}.z_{j,t}$

3) $qcc_{j,0,t} \leq \varphi_{j,0}.z_{j,t}$

4) $x_{h,j,t} = \sum_{m=0..M} qcc_{j,m,t}$

5) $p_{j,t} = \sum_{m=0..M} cc_{j,m}.qcc_{j,m,t} - a_j.z_{j,t}$

Explanation

1) Records the number of turbines turned on and off
2) Determines the flow through the pieces
3) Determines the initial flow through the pieces
4) The sum of the flow through the pieces equals the flow through the station
5) Determines the amount of power generated at each station

2.5 Reserve

There are two types of reserve which are modelled, tail water depressed, and spinning reserve. Tail water depressed is when a turbine is spun with air so that it can instantly run, however no water needs to be used, this has a cost associated with it which is shown in the objective function. Spinning reserve is when extra capacity on an already running turbine.

Indices

j = stations: Clyde, Roxburgh110, Roxburgh220

t = period: 1...336

Parameters

M_j = the maximum possible generation at station j

m_j = the maximum possible generation of a turbine at station j

N_j = the number of turbines at station j

RT_t = the target reserve which must be met

Decision variables

$\lambda_{j,t}$ = the number of turbines in tail water depressed at station j at time t

$\lambda^{+}_{j,t}$ = the number of turbines turned on for tail water depressed at station j at time t

$\lambda^{-}_{j,t}$ = the number of turbines turned off for tail water depressed at station j at time t

$s_{j,t}$ =the amount of spinning reserve offered at station j at time t

$p_{j,t}$ = the generation at station j at time t

r_t = the amount of reserve offered at time t

$d_{j,t}$ = the amount of tail water depressed reserve offered at station j at time t

Model CHERCO

1) $\lambda_{j,t} = \lambda_{j,t-1} + \lambda^{+}_{j,t} - \lambda^{-}_{j,t}$

2) $\lambda_{j,t} + z_{j,t} \le N_j$

3) $d_{j,t} \le m_j.\lambda_{j,t}$

4) $s_{j,t} + p_{j,t} \le m_j.z_{j,t}$

5) $s_{j,t} + p_{j,t} \le U_j$

6) $\sum\limits_{j \in STATIONS} s_{j,t} + \sum\limits_{j \in STATIONS} \lambda_{j,t} \ge R_t$

Explanation

1) Records the number of turbines turned on and off

2) The total number of turbines on must be less than the number of turbines

3) The amount of tail water depressed offered must be less than the capacity

4) The amount of spinning reserve offered must be less than the capacity

5) The total generation must be less than the maximum generation possible

6) The reserve target must be met

3 Results

This model was coded in AMPL/CPLEX. The model was run over several week long periods, using historical data in order to validate it. As the main objective is to maximise profit the most important thing is that the model generates when the price is high and doesn't when the price is low. Another big factor is the turbines should not be turning on and off too much, unless there is a huge variation in prices, due to the costs associated with turning turbines on.

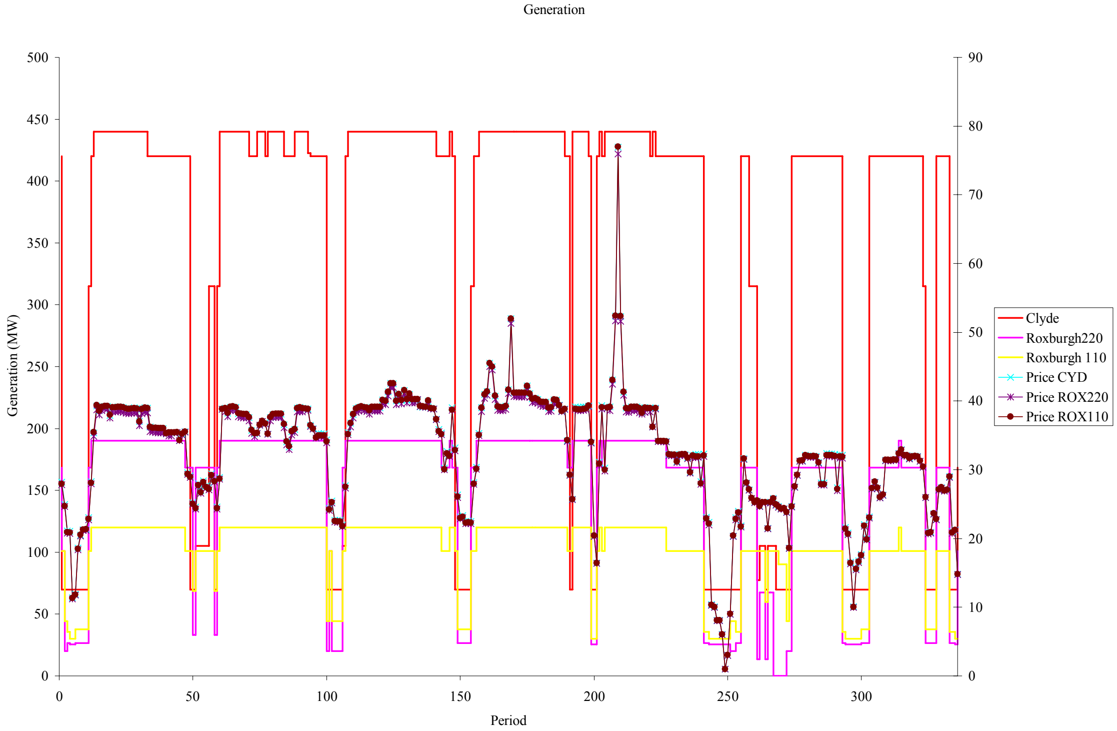


Figure 2: One week of generation

As can be seen in Figure 2 above the CHERCO model generates when the price is high and eases off when the price is low, it is also rare for turbines to turn on and then off again very soon afterwards. This shows that the model is behaving as it should. The generation can be seen more clearly when only a single day is shown, Figure 3 on the following page shows the generation for the Monday of this week. It is also worth noting that the high peaks of generation are where the model decides that it is worth sacrificing efficiency in order to generate more at high prices.

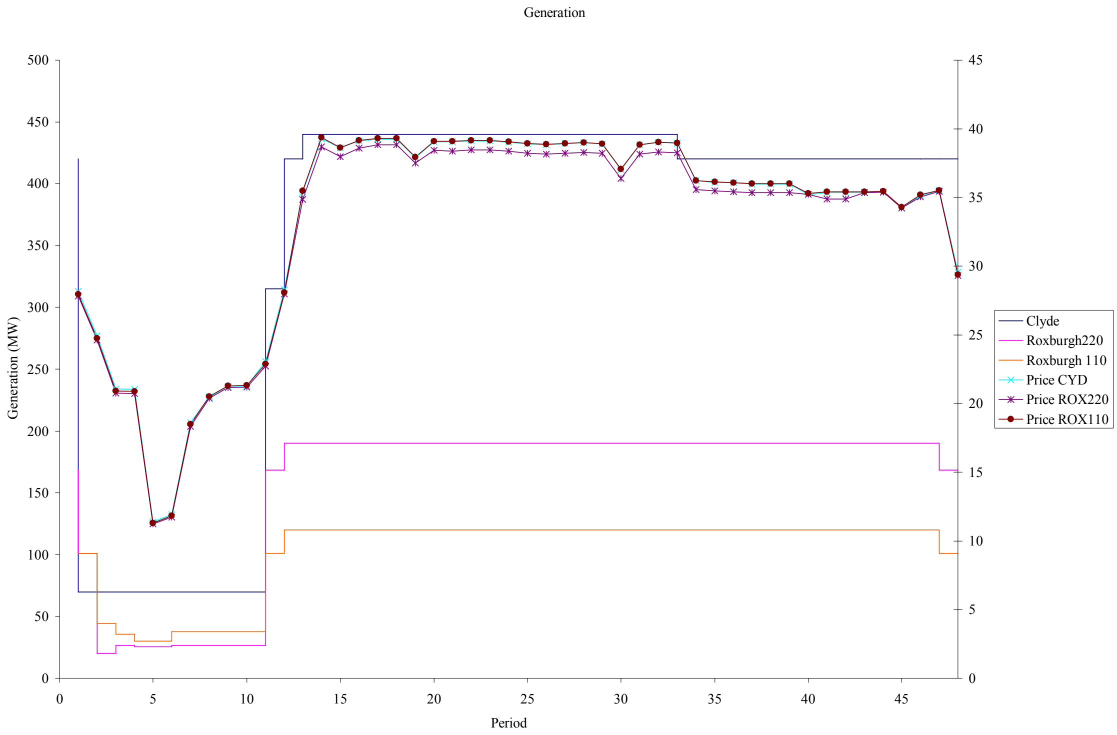


Figure 3: Monday generation

The other important features are that the Hawea release constraints work correctly. This can be seen in Figure 4 on the following page.

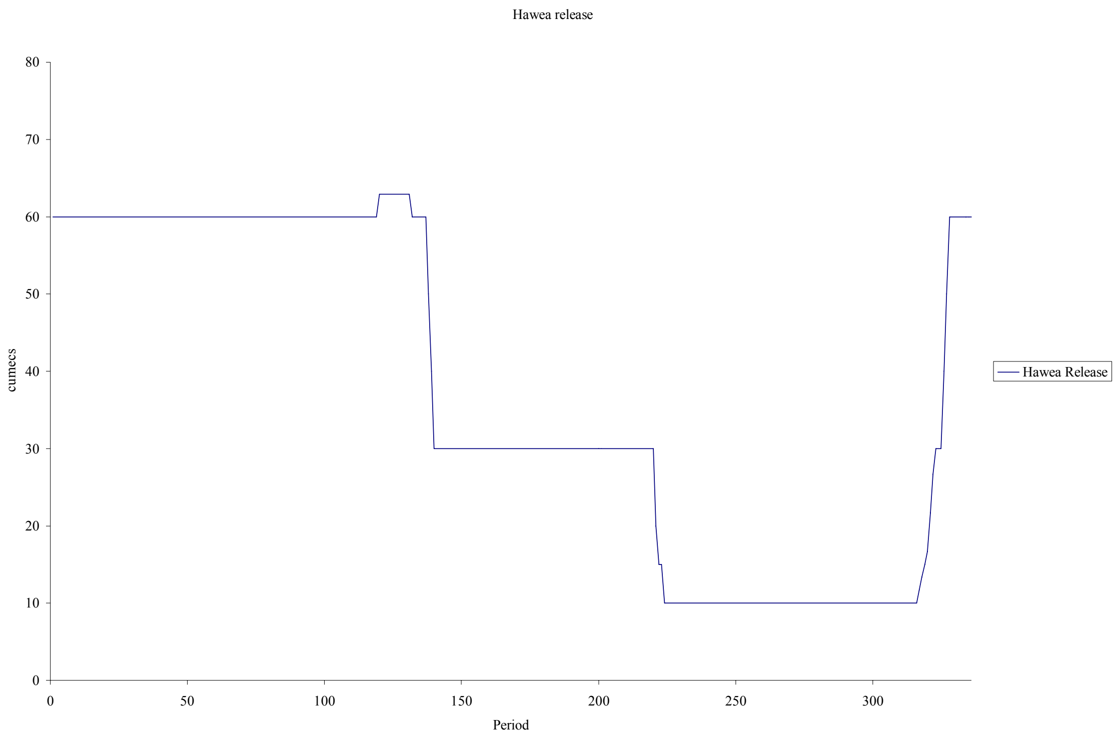


Figure 4: Hawea Release

The last main point of interest for the generator is how Hawea is run over the week. In Figure 5 below it was shown that it will run down the lake level then let it rise again, in this case the goal for Hawea is to return to its initial level.

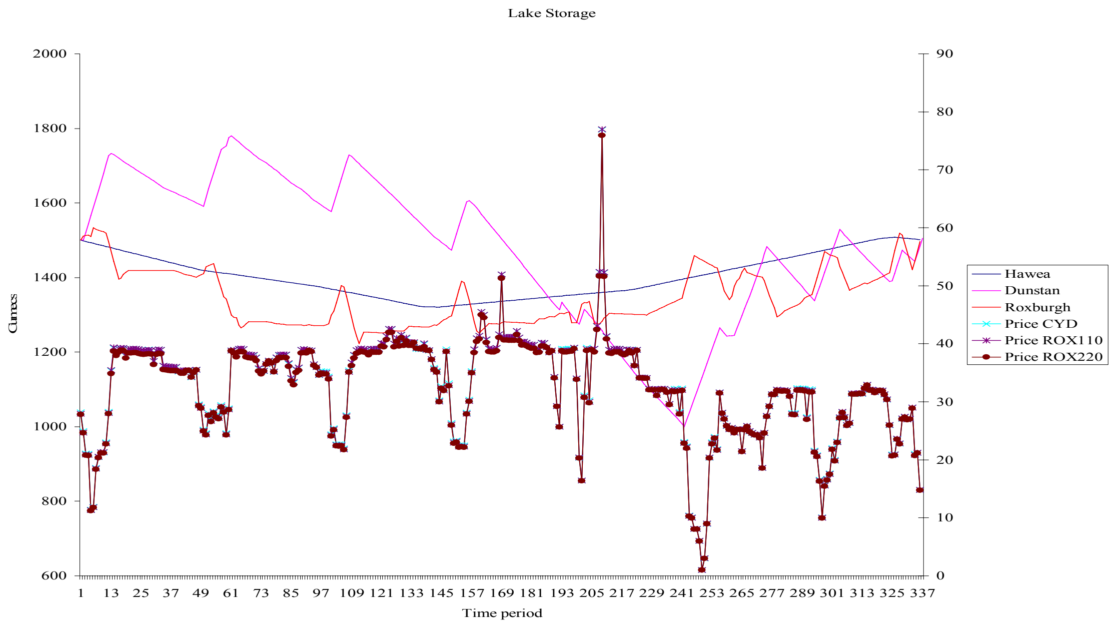


Figure 5: Storage over the week

4 Conclusions

The CHERCO model represents the whole Clutha river chain. It solves to optimise profit over a week given prices and inflows. The results so far appear to be good, it generates when price is high and doesn’t when the price is low. One interesting conclusion from this project so far is that it can be worth sacrificing efficiency in order to take full advantage of high prices.

Unit Crewing in the Airline Tour-of-Duty Planning Problem

Bassy Tam
Department of Engineering Science
University of Auckland
New Zealand
bassytam@gmail.com

Abstract

In crew scheduling, the aim of unit crewing is to keep groups of crew members from different crew ranks together, to operate a sequence of flights as a unit as much as possible. When such a group operates together for a sequence of flights, the sequence is called unit crewed. A unit crewed solution is considered to be more robust in the sense that aircraft departures are less likely to be delayed due to waiting for a crew member.

It is usual to solve the Tour-of-Duty (ToD) planning problem for each crew rank as a separate problem. To maximize unit crewing in the ToD solution, a set of ToDs for one rank in terms of minimal cost is constructed, and a weighted sum method is used to penalize any tour for the other rank that does not contain the same flight sequence.

We believe that a model that solves the two ToD planning problems simultaneously can improve the number of unit crewing sequences without increasing the total crew cost. We developed a bi-criteria model with an additional set of constraints to ensure that the maximum number of unit crewing connections can be chosen between the two crew ranks subject to a planned crew cost the airline accepts to pay.

1 Airline Crew Scheduling

Commercial airlines are required to solve many resource scheduling problems to ensure that aircraft and aircrew are available for each scheduled flight. In this paper we consider the *tour-of-duty* (ToD) planning problem: finding sequences of flights to partition the flight schedule into ToDs that can be flown by one crew member.

A ToD is an alternating sequence of *duty periods* and *layovers*, where duty periods comprise one or more flights, and may also include passengering flights. A *passengering flight* is one on which a crew member travels as a passenger in order to be positioned at a particular airport for a subsequent operating flight or to return to their home base.

The first duty period of a ToD must start at a *crew base*, and the last duty period must end at the same crew base. An airline might have several bases (cities) at which crew are domiciled. Each ToD will have an associated crew complement made up of a number of crew of some ranks.

Construction of a legal pairing is subject to a complex set of rules imposed by government and/or employment contracts. Those rules are very difficult to characterise mathematically in any other way than by enumeration. In addition, the cost of a pairing is usually a complex function of its components. Therefore, the ToD planning problem usually separates the problem of generating pairings from the problem of selecting the best subset of these pairings.

The cost minimization problem is then modelled under the assumption that the set of feasible pairings and their costs are explicitly available, and can be expressed as a *generalized set partitioning problem* (GSPP):

$$\begin{array}{lccc} \text{Minimize} & c^T x & & \\ \text{subject to} & \boldsymbol{A}x & = & e \\ & \boldsymbol{M}x & \begin{Bmatrix} \geq \\ = \\ \leq \end{Bmatrix} & b \\ & x & \in & \{0,1\}^n \end{array}$$

where $\boldsymbol{A}$ is a binary matrix and e is a vector of ones.

In the ToD planning model, each column or variable corresponds to one feasible ToD that can be flown by some crew member. The value of c_j, the cost of variable j, reflects the dollar cost of operating the j^{th} ToD. The decision variable x_j is equal to 1 if pairing j is included in the solution and 0 otherwise. The first set of constraints is referred to as *flight constraints*, and the second set contains the *crew base balancing constraints*.

Each *flight constraint* corresponds to a particular flight segment and ensures it is included in exactly one ToD. The elements of the $\boldsymbol{A}$ matrix can then be defined as:

$$a_{ij} = \begin{cases} 1, & \text{if the } j^{\text{th}} \text{ pairing includes } i^{\text{th}} \text{ flight in operation,} \\ 0, & \text{otherwise.} \end{cases}$$

However, if the j^{th} pairing includes the i^{th} flight as passengering flight, this will result in a column in which $a_{ij} = 0$.

The ToD planning model is usually augmented with additional constraints that permit restrictions to be imposed on the number of crew resources included from each crew base. This set of constraints is referred to as *crew base balancing constraints*. These crew base balancing constraints ensure the distribution of work over the set of crew bases matches the crew resources. They require that the number of crews contained in the chosen pairings which originate at a given crew base must be between specified lower and upper bounds. Each crew base balancing constraint represents a crew base restriction for the respective crew base. In this case, b_i is the maximal/minimal available resource, and m_{ij} is the resource attributed to the crew base balancing constraint i if pairing j is used.

One of the main difficulties with the ToD planning problem is that the number of feasible pairings is extremely large even for relatively small problems, so generating all possible ToDs for the optimization problem is often impossible. To some extent, this problem can be overcome by using a dynamic column generation technique. A *dynamic column generation* technique generates columns during the optimization process, i.e. generation of pairings and solving the generalized set partitioning problem is done iteratively during the optimization process.

To obtain an integer solution, a *branch and price* approach with special *constraint branching* strategy is used. The branch and price procedure is similar to the branch and bound technique, but dynamic column generation is used at each node of the branch and bound tree. *Follow-on branching*, which is a constraint branching strategy commonly used for this type of problem, forces or prohibits two flights to be operated as a subsequence in a pairing. On the one branches, all pairings that operate only one of the two flights are eliminated. On the zero branches, all pairings that operate both flights are eliminated. See Butchers et al (2001) for more details.

2 Operational Robustness

The ToD planning problem is solved well before the flight schedule becomes operational. In this planning stage, all flights are assumed to have departure times that are both fixed and known. This assumption is often proven wrong when the crew schedule is actually implemented. ToDs are usually less expensive if crews spend less time on ground between arrival and departure of two consecutive flights, hence the total working or operating hours are minimized.

Such crew schedules happen to become “de-optimized” in actual operation, as they are easily disrupted and chain impacts are usually found as a result. Thus, if the airline provides connection times between consecutive flights to both aircraft and crew members, which just satisfy the minimum time legally required, a late arriving flight will cause the following flight to depart late. However not only will the downstream flight that operates on the delayed aircraft depart late, but also the late arriving crew members who are changing aircraft will board late for their outgoing flights. After a few aircraft changes, many flights may be delayed by the initially minor delay.

Furthermore, disruptions may require the use of reserve crews to get back on schedule and originally scheduled crew might not be able to continue on their duty because of rule violations. As a result, substantial unplanned costs, such as overtime, fuel costs and compensations for parking and passengers with delayed or cancelled flights, can be incurred. Therefore airlines do not only require minimum cost solutions, but are also very interested in robust solutions.

A *robust ToD planning problem* is the problem of obtaining aircrew schedules in planning that are not necessarily optimal in terms of the crew cost in plan but that yield low crew cost in operations. Approaches to robust aircrew scheduling have been developed only recently, and have different measures of operational robustness.

Schaefer et al. (2001) solve a problem very similar to the original ToD planning model. However, they replace the objective coefficients c_j in the model with the expected cost of the j^{th} ToD. The observed effect of this approach is to produce ToDs with lower operational cost than those produced with planned cost.

Ehrgott and Ryan (2002) observe that a robust solution would have the property that if an upstream flight is likely to be delayed, crew should not be scheduled to change aircraft for a successive flight, which leaves after only minimal ground time. Thus, crews change aircraft between operating flight sectors less frequently in a robust ToD solution. Based on this observation, they develop an objective function to penalize ToDs which are not robust. They then try to minimize this objective while at the same time maintaining a cost effective solution.

Yen and Birge (2006) solve the robust ToD planning model with a similar robustness measure as described above (Ehrgott and Ryan) as a two-stage stochastic binary programming model with recourse by assuming the flight operation time is a random variable.

Shebalov and Klabjan (2006) solve the ToD planning problem based on the observation of a recovery procedure that uses crew swaps. In addition to the traditional objective of minimizing the ToD cost, they introduce a new objective of maximizing the number of opportunities for crew swapping. Thus, their model is a bi-criteria optimization model.

All approaches only consider the ToD planning problem for one crew rank, problems might arise when airline has to operate different crew schedules for different crew ranks during the same period of time.

3 Unit Crewing

In some airlines, crew members may have different employment contracts, different scheduling rules and different pay schemes, even if they belong to the same crew type. So ToD planning problems have to be solved separately for each crew rank. This might cause a crew to split up so that some members may join other crew members to make up a crew on one flight while other members may join a second crew for another flight. This splitting of the crew is not robust from an operational point of view.

If an incoming flight arrives late, the crew members on that flight will arrive late which causes the crew members on board to be late for the downstream flight if the connection time between the two flights is tight. If members of the crew in the upstream flight split up and join other crew members to form new crews for downstream flights, then more than one outgoing flight will be affected.

In fact, this affects the technical crew more seriously, as there are fewer options for the recovery procedure for technical crew than for the flight attendants, since the number of reserve flight attendants is higher than the number of reserve pilots, and flight attendants are able to work on different types of aircraft, whereas pilots can only operate certain fleet type(s).

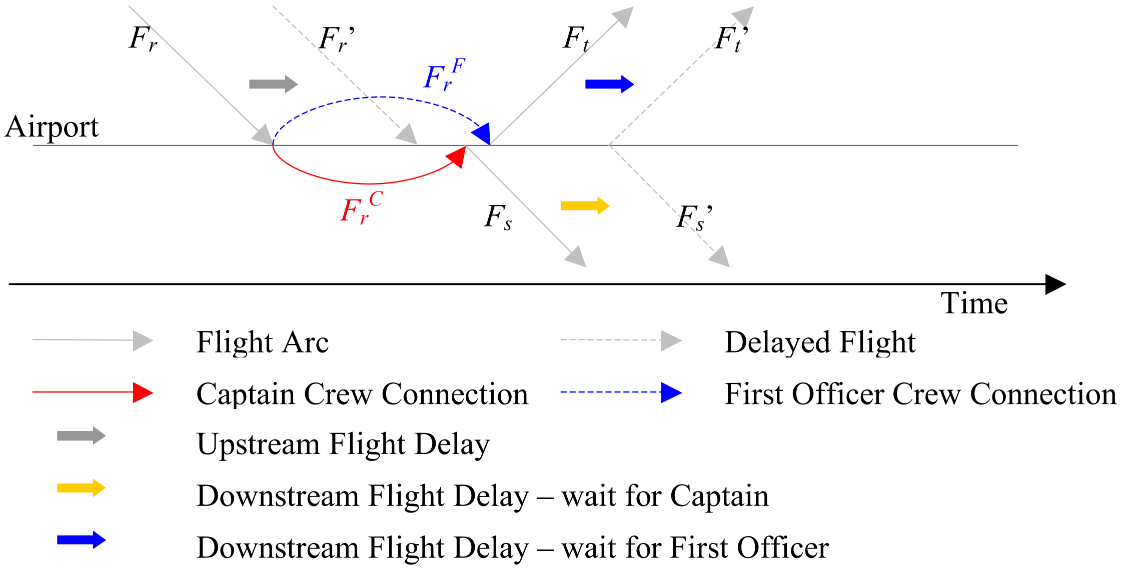


Figure 1: Non-Unit Crewed Schedule

Normally, there are two types of pilots (the Captain and the First Officer) to operate one aircraft in Air New Zealand domestic operations. Figure 1 shows a flight F_r that has one Captain ($F_r^{\,C}$) and one First Officer ($F_r^{\,F}$) scheduled on its operation, with flight F_s and flight F_t both having short connection time to F_r. $F_r^{\,C}$ was scheduled to join another First Officer to operate F_s and $F_r^{\,F}$ was scheduled to join another Captain to operate F_t. If F_r is late on its arrival, then $F_r^{\,C}$ and $F_r^{\,F}$ will be on board late for their next operating sectors, F_s and F_t, respectively, i.e. both flights are delayed on their departure.

Unit Crewing is an approach to constructing crew schedules so that members of a crew perform the same sequence of flights for as much of their duty periods as possible. A unit crewed schedule is considered to be more operationally robust in the sense that if members of a crew perform the sequence of flights for as much of their duty periods as possible, it is less likely to delay other flights due to waiting for a member of the crew from a disrupted upstream flight.

A *unit crewing connection* is a connection between two consecutive flights, F_r and F_s, where each member of the crew for F_r, operates F_s as a subsequence. Any connection between the start of a duty period and the first operating sector in the duty period is also considered to be a unit crewing connection.

If some crew members operate flights F_r and F_s as a subsequence and some crew members operate F_s as a starting sector of their duty period, delay might occur on F_s if F_r is disrupted, as F_s needs to wait for the crew members from F_r. So if each crew member operates the sector F_s as a starting sector in their duty period, then that is a unit crewing connection between the start of duty and F_s.

However, a connection between the last operating sector in the duty period and the end of that duty period is never considered as a unit crewing connection, as no downstream flight departure time is affected.

It is usual to obtain a crew schedule for one rank in terms of minimal cost (i.e. the traditional ToD planning model in GSPP formulation) and then select as many unit crewing connections as possible when solving the next rank. A main drawback for this method is that the solutions might not be Pareto optimal when we consider the total planned crew cost for both ranks and the number of unit crewing connections as objectives.

We believe that by solving both ranks simultaneously rather than in sequence it might be possible to lower the total crew cost with the same number of unit crewing connections.

This is because before we solve the unit crew ToD planning problem, a cost optimized crew schedule for one rank needs to be found. When we construct the cost minimal crew schedule for this rank, connections in the ToDs were selected by considering the cost and feasibility of the schedule for this rank only, without considering the optimality (or even feasibility) for the other ranks.

The only way to find ToD solutions for different crew groups that are unit crewed with each other is to solve the ToD planning problem for all ranks simultaneously, so that during the process of pairing generation, consecutive flight connections that are preferable for all ranks can be constructed.

4 Bi-Criteria Model

In the combination of two ToD planning problems, two sets of basic ToD constraints are involved in the problem, one for each crew rank. An additional set of constraints reflects the possible unit crewing connections between the two ToD planning problems.

Therefore, two sets of flight constraints, defined by $\boldsymbol{A_1}x_1 = \mathbf{e}$ and $\boldsymbol{A_2}x_2 = \mathbf{e}$, are included in the combined model. The columns of the matrix $\boldsymbol{A_1}$ correspond to possible pairings that can be flown by some crew member in the first rank (say the Captains), and the columns of the matrix $\boldsymbol{A_2}$ correspond to possible pairings that can be flown by some crew member in the other rank (say the First Officers). The vector of decision variables x_1 represents which pairing is to be included in the Captains' schedule, where x_{1j} is equal to 1 if the j^{th} possible pairing that can be flown by a Captain is included in the Captains' crew schedule and 0 otherwise. The vector of decision variables x_2 represents the First Officers' schedule, where x_{2j} is equal to 1 if the j^{th} possible pairing that can be flown by a First Officer is included in the First Officers' solution and 0 otherwise.

Two sets of crew base balancing constraints, defined by $\boldsymbol{M_1}x_1$ $\{\leq, =, \geq\}$ b_1 and $\boldsymbol{M_2}x_2$ $\{\leq, =, \geq\}$ b_2, representing the crew base restrictions for the Captain crew and First

Officer crew, respectively, are also included in the combined model. The Captains' maximal/minimal possible resources are represented by b_{1i}, and m_{1ij} is the resource assigned to the crew base corresponding to the Captains' constraint i if Captains' pairing j is used in the Captains' solution. The maximal/minimal possible resources that can be assigned to the First Officers are represented by b_{2i}, and therefore m_{2ij} is the resource assigned to the base corresponding to the First Officers' constraint i if pairing j that can be flown by some First Officer, is used in the First Officers' schedule.

The value of c_{1j} is the cost of operating the jth possible pairing in the Captain's set of ToDs. The value of c_{2j} is the cost of the jth possible pairing that can be flown by a First Officer.

To ensure that all crew ranks follow the same connections between flights as much as possible, a set of *unit crewing constraints* are added. The unit crewing constraints are derived from the ***A*** matrices for the two ToD planning problems. The combination of the two ToD planning problems with unit crewing constraints is formulated as:

$$
\begin{array}{lccccccc}
\text{Minimize} & c_1^T x_1 & +c_2^T x_2 & & & & & \\
\text{Minimize} & & & s_1 & +s_2 & & & \\
\text{subject to} & \mathbf{A_1} x_1 & & & & & = & e \\
 & \mathbf{M_1} x_1 & & & & & \left\{\begin{matrix}\geq\\=\\\leq\end{matrix}\right\} & b_1 \\
 & & \mathbf{A_2} x_2 & & & & = & e \\
 & & \mathbf{M_2} x_2 & & & & \left\{\begin{matrix}\geq\\=\\\leq\end{matrix}\right\} & b_2 \\
 & \mathbf{U_1} x_1 & -\mathbf{U_2} x_2 & -s_1 & +s_2 & & = & 0 \\
 & x_1, & x_2, & s_1, & s_2 & & \in & \{0,1\}^n
\end{array}
$$

where 0 is a vector of zeros, e is a vector of ones and $\boldsymbol{A_1}$, $\boldsymbol{A_2}$ and $\boldsymbol{U_1}$, $\boldsymbol{U_2}$ are matrices of zeros and ones.

Suppose the set of unit crewing connections **U** is defined by $\mathbf{U} = \mathbf{T} \cup \mathbf{S}$, where the set **T** represents all possible consecutive operating flight connections, and **S** is the set of first operating flights in any duty periods. Each element t in the set **T** can be defined as $t = (F_r, F_s)$, where F_r and F_s are flight segments that need to be covered. If a crew member operates flight F_s as a subsequence of operating flight F_r, then the connection is represented by (F_r, F_s). Elements in the set **S** can be defined as connections between the start of a duty period and the first operating sector in the duty period.

Each unit crewing constraint in the combined ToD planning problem corresponds to a particular element in the set **U**. The element u_{1ij} of the $\boldsymbol{U_1}$ matrix is equal to 1 if the jth pairing, which can be flown by a Captain, includes the ith connection of the set **U** and 0 otherwise. The same definition applies to the elements of the $\boldsymbol{U_2}$ matrix, but for the pairings that First Officers can fly. Therefore, the ith connection in the set **U** will be unit crewed if $u_{1ij}x_{1j} - u_{2ik}x_{2k}$ is equal to zero, that is, if the jth pairing is selected in the Captains' solution and the kth pairing is selected in the First Officers' solution, in which both of the pairings contain the ith flight connection in the set **U**.

Remember that a completely unit crewed set of ToDs is unlikely to be obtained. So a slack and surplus variable is needed for each unit crewing constraint, to allow any non-unit crewing connections. The surplus variable s_{1i} will be equal to 1, if the i^{th} consecutive flight connection for the set **U** is selected in the Captains' solution but no such connection is included in the First Officers' schedule and 0 otherwise. The slack variable s_{2i} will be equal to 1, if the i^{th} consecutive flight connection for the set **U** is included in the First Officers' solution but this connection is not selected in the Captains' schedule and 0 otherwise.

The second objective function, denoted by $s_1 + s_2$, is the sum of all the slack and surplus variables for the unit crewing constraints, is the minimum number of non-unit crewing connections within the two crew schedules that we are looking for.

5 Elastic Constraint Model

To obtain all Pareto optimal solutions for the combined ToD planning problem with unit crewing constraints, we use the elastic constraint method, this method allows not only all Pareto optimal solutions to be found, but is also flexible enough to allow management to select a preferred solution for implementation.

In the elastic constraint method, we solve the LP relaxation of the two cost minimizing ToD planning problems separately; this gives us an optimal planned total cost. This cost is then used to define the upper bound for the planned cost we accept to pay when we are minimizing the number of non-unit crewing connections.

To limit the size of the combined problem, unit crewing constraints are not pre-constructed. That is, constraints are not known a priori but are constructed during the optimization and branch and price processes.

The process starts with a problem with no unit crewing constraint. The LP relaxations of two cost minimizing ToD planning problems are solved separately. We then construct unit crewing constraints based on the sum of fractional values for the consecutive connections of the two problems. The cost of each ToD is used to construct the elastic cost constraint and the optimal planned crew cost for the LP relaxation is used to define the upper bound for the planned cost. This yields a restricted problem with the following formulation:

$$
\begin{array}{llllllll}
\text{Minimize} & & & s_1 & +s_2 & +ps_c & & \\
\text{subject to} & c_1^T x_1 & +c_2^T x_2 & & & -s_c & \leq & \varepsilon \\
 & \mathbf{A_1}x_1 & & & & & = & e \\
 & \mathbf{M_1}x_1 & & & & & \left\{\begin{matrix}\geq\\=\\\leq\end{matrix}\right\} & b_1 \\
 & & \mathbf{A_2}x_2 & & & & = & e \\
 & & \mathbf{M_2}x_2 & & & & \left\{\begin{matrix}\geq\\=\\\leq\end{matrix}\right\} & b_2 \\
 & \mathbf{U_1}x_1 & -\mathbf{U_2}x_2 & -s_1 & +s_2 & & = & 0 \\
 & x_1, & x_2, & s_1, & s_2 & & \in & \{0,1\}^n \\
 & & & & & s_c & \geq & 0
\end{array}
$$

The right hand value for the first constraint is the planned cost we desired to pay. The new surplus variable for this cost constraint, s_c, is introduced when the branch and price process begins, to reduce computational difficulties arising from this constraint (see Ehrgott and Ryan, 2002 for details). The cost coefficient for the surplus of the cost constraint is the penalty for violation of the cost constraint.

We only construct new constraints when there is no column returned from the column generator, i.e. when the current restricted master problem is optimal. To further reduce the size of the combined problem, we only select connections with sum of fractional values that is neither zero nor one in both problems to be part of the constraints in the model. That is, we only include connections as part of the unit crewing constraints once they are not unit crewed.

To obtain integer solutions, the original branch and price process is used, but the time required is long. In order to speed up the branch and price process, we have to make some adjustment on the branch and price process. See later section for details.

6 Dantzig-Wolfe Decomposition

We tried to use a Dantzig-Wolfe decomposition approach to solve the combined model as well. The advantage of using this approach is that the structure of the original problems remains unchanged. Thus no complexity is added to the already difficult problems. The two independent ToD planning problems are the sub-problems under the Dantzig-Wolfe decomposition approach. The set of unit crewing constraints is included in the master problem under the Dantzig-Wolfe decomposition approach.

The process starts with an empty master problem. When the i^{th} independent ToD planning problem is solved, this gives a matrix $\boldsymbol{X_i}$, which represents a set of basic feasible solutions to the i^{th} ToD planning problem. The j^{th} column of the matrix $\boldsymbol{X_i}$, x_i^j, is the j^{th} extreme point to the i^{th} ToD planning problem.

When all sub-problems are solved, unit crewing constraints are constructed based on the optimal solution of the sub-problems. The crew cost for each extreme point is used to define the elastic cost constraint, and the total optimal planned crew cost of the two ToD planning problems is used to define the upper bound of the cost. The j^{th} extreme point, x_i^j, in the set $\boldsymbol{X_i}$ leads to a column in the master problem in the form of $(c_i^T x_i^j, \boldsymbol{U_i} x_i^j, e_i)^T$, where e_i is a vector of two elements with the i^{th} element equal to 1 and 0 otherwise. The master problem is formulated as:

$$
\begin{array}{llllllll}
\text{Minimize} & & & s_1 & +s_2 & +ps_c & & \\
\text{subject to} & c_1^T \mathbf{X_1} \ddot{e}_1 & +c_2^T \mathbf{X_2} \ddot{e}_2 & & & -s_c & \leq & \varepsilon \\
 & \mathbf{U_1 X_1} \ddot{e}_1 & -\mathbf{U_2 X_2} \ddot{e}_2 & -s_1 & +s_2 & & = & 0 \\
 & e^T \ddot{e}_1 & & & & & = & 1 \\
 & & e^T \ddot{e}_2 & & & & = & 1 \\
 & \ddot{e}_1, & \ddot{e}_2 & & & & \geq & 0 \\
 & & & s_1, & s_2 & & \in & \{0,1\}^n \\
 & & & & & s_c & \geq & 0
\end{array}
$$

The element for the k^{th} unit crewing constraint in the j^{th} extreme point for the i^{th} ToD planning problem, $e_k^T \boldsymbol{U_i} x_i^j$, is simply the sum of fractional values for the k^{th} unit crewing connection in the solution x_i^j (where e_k is a vector of zeros with the k^{th} element equal to one). The j^{th} element for the vector $c_i^T \boldsymbol{X_i}$ is the crew cost for the j^{th} extreme point solution of the i^{th} ToD planning problem.

The vectors of decision variables (λ_1, λ_2) can be interpreted as the weight on each extreme point for any feasible solution. The two new constraints are the *convexity constraints*, to ensure any feasible solution in the master problem is a linear convex combination of the extreme points of the bounded polyhedral set.

After the master problem is constructed based on the extreme points found from the sub-problems, the master is solved to optimality in terms of minimal number of non-unit crewing connections, subject to the unit crewing constraints constructed from the optimal solution of all sub-problems. Extra unit crewing constraints are added based on the current solution of the master because the current solution of the master problem might contain non-unit crewing connections that are unit crewed in the optimal solution of the sub-problems.

The restricted master problem is solved by repeating the processes of solving the master and including new unit crewing constraints. The optimal solution of the restricted master problem is obtained once no new unit crewing constraint is found.

After the master problem is solved, the optimal dual variables of the restricted master problem, (π_c, π_u^T, π_1, π_2), will be passed back to the sub-problems in order to generate some better columns for the master. The scalar, π_c, is the dual variable associated with the cost constraint, and (π_1, π_2), are the dual variables associated with the convexity constraints for the two sub-problems. The vector π_u^T is the dual vector associated with the unit crewing constraints of the master problem.

To generated negative reduced cost column, the objective function of the first sub-problem will be re-formulated as:

$$\text{Minimize} \qquad (-\pi_c c_1^T - \pi_u^T \boldsymbol{U_1})x_1 - \pi_1$$

and the objective function of the second sub-problem will be re-formulated as:

$$\text{Minimize} \qquad (-\pi_c c_2^T + \pi_u^T \boldsymbol{U_2})x_2 - \pi_2$$

and since π_i is a constant, it is often ignored.

Before we pass the master optimal dual vector to the sub-problems, appropriate adjustments on the dual values are required. If the optimal basis of the master contains the k^{th} slack variable for the unit crewing constraint, $e_k^T s_2$, with a value of 0, this is the same as to have the k^{th} surplus variable, $e_k^T s_1$, with a value of 0, and vice versa.

If the k^{th} slack variable for the unit crewing constraint is in the basis, then the k^{th} element of π_u will be negative. This negative dual value tells the second sub-problem to generate solutions contain the k^{th} unit crewing connection if it is possible, but to avoid this connection in the first sub-problem. The same implication holds if the surplus variable is in the basis, but the first sub-problem will generate solutions containing this connection while this connection is avoided in the second sub-problem. So if the basis of the master contains a slack or surplus variable with a value of 0, we have to adjust the dual value for this slack/surplus corresponding constraint to be 0. This means we put no preference on this connection in both sub-problems.

The processes of solving the restricted master problem to optimality and generating columns for the master are repeated until no columns are return to the master from any of the sub-problems.

Note that it is not necessary for the slack and surplus variables for the unit crewing constraints in the master problem to be binary. If the sub-problems are integral, the slack and surplus variable for the unit crewing constraints in the master problem will be naturally integer. However, it is not wise to solve each sub-problem to integrality before including it as a column of the master by the consideration of solution time. So a good branch and bound strategy together with appropriate adjustment is needed to obtain integral solutions.

The value of $e_k^T \boldsymbol{U_i X_i} \lambda_i$ is the preference to the master problem on the k^{th} connection in the i^{th} sub-problem. If the value of $e_k^T \boldsymbol{U_i X_i} \lambda_i$ is close to 1, the master prefers to include the k^{th} connection in the i^{th} crew schedule, and the connection is not favoured if $e_k^T \boldsymbol{U_i X_i} \lambda_i$ is close to 0.

This makes up our branching strategy, as each element for the unit crewing constraint represents the sum of fractional values of the corresponding connection, and the value of $e_k^T\boldsymbol{U_iX_i}\lambda_i$ is the linear combination of the sum of fractional values on the k^{th} connection from extreme points generated from the i^{th} sub-problem. This means we select a branch for the follow-on branching strategy base on the value of $\boldsymbol{U_iX_i\lambda_{i.}}$ If $e_k^T\boldsymbol{U_iX_i}\lambda_i$ is close to 1, then a 1-branch should be imposed on the k^{th} connection in the i^{th} sub-problem, as this might yield a better solution. And a 0-branch on the k^{th} unit crewing connection if the value of $e_k^T\boldsymbol{U_iX_i}\lambda_i$ is close to 0.

The value of the slack and surplus variables for the unit crewing constraints in the master problem can be seen as the likelihood for the connection to be unit crewed. If the slack and surplus values for the k^{th} unit crewing constraint are both close to 0, then the k^{th} connection is more likely to be unit crewed. In other words, the preference to occupy this connection is the same for both crew members, that is both crew members are going to occupy this connection or they are not going to take this connection, and hence the same branch should be imposed in the sub-problems. If the slack or surplus value is close to 1, then this connection is less likely to be unit crewed and opposite branches should be imposed in the two sub-problems.

If a 1-branch is imposed on the k^{th} connection in the i^{th} ToD planning problem, then this sub-problem will always generate solutions containing the k^{th} connection. That is, it always generates solutions leading to a column in the master problem with a value of one at the k^{th} element of the unit crewing constraints, i.e. the value of $e_k^T\boldsymbol{U_i}x_i$ must be equal to 1, where x_i is any solution generated from the i^{th} sub-problem after a 1-branch is imposed on the k^{th} connection. To ensure the master selects this connection, we have to remove all columns from the master problem generated previously with the k^{th} element of $\boldsymbol{U_iX_i}$ not equal to 1. Same adjustment is performed if a 0-branch is imposed, but all columns with the k^{th} element of $\boldsymbol{U_iX_i}$ not equal to 0 are removed from the master.

7 Computational Difficulties

It is difficult to obtain an integral solution for the combined ToD planning problem with unit crewing constraints, as the sum of fractional values spread on many connections. This problem observed severity if we allow too many unit crewing constraints in the combined model or when we use the Dantzig-Wolfe method to solve the problem.

To overcome this problem, we have to limit the number of unit crewing constraints allowed in the problem and to impose a 1-branch on the unit crewed connection if it is operated on the same aircraft.

In the original branch and price process, we often impose 1-branch on a connection with sum of fractional values close to 1. But in our modified process, we also impose a 1-branch on connection with sum of fractional values equals to 1 if this connection is on the same aircraft and unit crewed. This is to prevent those originally unit crewed connection to become non-unit crewed.

Another advantages on impose 1-branch on those unit crewed connections are connections in the underlying flight network are limited and number of variables are reduced. The deduction on the flight network makes the column generation process faster and the deduction on number of variables makes the reduced cost calculation faster.

From our observations, if there is not enough 1-branches to solidify the flight network at the first LP relaxation, problem of spread over remained. Since the spread over of the sum of fractional values is caused by the unit crewing constraints and the more unit crewing constraints are included in the model, the worse of the problem. So we have to limit the number of unit crewing constraints in the model.

This means we have to choose what kind of connections should be formulated as unit crewing constraints when we have more non-unit crewed connections than number of constraints allowed.

This causes other problems. As if too few constraints are allowed in the model, good quality unit crewed schedules cannot be obtained. But while if too many constraints are allowed, problem of spread over remained.

8 Computational Results

Next we report the implementation results for the test problem based on a domestic flight schedule. The test problem is a 7 day flight schedule consisting of 442 flights. This schedule services seven cities (Auckland, Christchurch, Dunedin Hamilton, Rotorua, Queenstown and Wellington), with Auckland, Christchurch and Wellington as the crew bases.

With this moderate sized flight schedule, 253594 connections were generated from the flight network. This means we can include as many as 253594 unit crewing constraints into the combined problem.

Following table gives a brief summary on the performance from different methods. The second column of the table shows the total crew cost for the two ranks, the third column shows the percentage increase on total crew cost in compare to the LP optimal. The fourth column shows the number of non-unit crewed connections and the fifth column shows the number of unit crewed connections. The last column shows the time required to obtain the solution.

	Cost	**% Increase**	$\bar{u}$	u	**Time**
Separate	743035	0.2024%	239	690	20 mins
Combined	742061	0.0711%	1	809	2 hours
Dantzig-Wolfe	743825	0.3090%	1	812	8 hours

Only 10 minutes are required to solve the ToD planning problems for each of the two ranks separately. We solved the same problem with a limit of 100 unit crewing constraints and right hand size values of LP optimal and a penalty of 0.01 on each dollar cost violation under the combined model and Dantzig-Wolfe method. With no significant increase in crew cost, we obtained a set of schedules which are nearly perfect unit crewed. However, we need a lot longer time when we use the Dantzig-Wolfe method.

9 References

Anbil, R.; Tanga R.; and Johnson E. L. 1992, “A Global Approach to Crew-Pairing Optimization”, *IBM Systems Journal*, Vol. 31, No. 1, pp. 71 – 78.

Butchers, E. R.; Day, P. R.; Goldie, A. P.; Miller, S.; Meyer, J. A.; Ryan, D. M.; Scott, A. C.; and Wallace C. A. 2001, “Optimized Crew Scheduling at Air Zealand,” *Interfaces*, Vol. 31, No. 1 (January – February), pp. 30 – 56.

Dantzig, G.; and Wolfe, P. 1960, “Decomposition Principle for Linear Programs,” *Operations Research*, Vol. 8, pp. 101 – 111.

Ehrgott, M. and Ryan D.M. 2002, “Constructing Robust Crew Schedules with Bicriteria Optimization,” *Journal of Multi-Criteria Decision Analysis*, Vol. 11, pp. 139 – 150.

Schaefer, A. J.; Johnson E. L.; Kelywegt A. J.; and Nemhauser G. L. 2001, “Airline Crew Scheduling under Uncertainty,” Technical Report TLI-01-01, Georgia Institute of Technology.

Shebalov S.; and Klabjan D. 2006, “Robust Airline Crew Pairing: Move-up Crews”, *Transportation Science*, Vol. 40, pp. 300 – 312.

Yen, J. W. and Birge J. R. 2006, “A Stochastic Programming Approach to the Airline Crew Scheduling Problem”, *Transportation Science*, Vol. 40, pp. 3 – 14.

Optimization of Well Placement and Flow

Lei Zhang
Department of Engineering Science
The University of Auckland
New Zealand
lzha174@ec.auckland.ac.nz

Abstract

An optimization tool is developed to assist water companies to choose optimal well locations in a confined aquifer in order to maximize the expected profit of pumping and selling water subject to uncertainty in the conductivity of the porous media over a confined aquifer. This tool simulates the groundwater flow to measure the change in head at specified head observation points due to pumping. This information is then used in the optimization model to determine the well locations.

An expected value model is solved first to obtain an expected value solution. If the solution is not acceptable, a more expensive two-stage stochastic model is solved to find a solution that is better hedged against the uncertainty in the conductivity.

For the case study considered, solving the two-stage stochastic model improves the maximum profit by a significant amount compared to solving the expected value problem.

1 Introduction

Groundwater is defined as freshwater contained in aquifers below ground. Groundwater is accumulated by rainfall sinking into the soil over a long period of time. When rain water reaches the earth's surface it keeps going down until it penetrates through the soil to be stored in aquifers [1]. An aquifer which is confined between two impermeable aquitards is defined as a confined aquifer. A well that goes through a confining layer is known as an artesian well [2]. Because of its abundance and purity, it is an important resource for providing fresh water in many locations. In this optimization tool, a set of candidate well locations must be given and optimal well locations are chosen within the set. Head observation points are sampled over the confined aquifer to monitor the overall water table dropping over the operational field.

In a confined aquifer, the hydraulic head h is a linear function of the pumping rate vector $\mathbf{q}$,

$$h(\mathbf{q}) = h_i^0 + \sum_{j=1}^{n} a_{ij} q_j ,$$

where h_i^0 is the initial head at a head observation point i without pumping at candidate well location j and q_j is the new applied pumping rate at candidate well location j. The response coefficient a_{ij} is equal to the head drop at an observation point i due to a unit flow at candidate well location j [3]. The simulation software COMSOL is used to determine the response coefficients [4].

Stochastic programming can be used to make optimal decisions while taking into consideration uncertainty as to how the future might unfold. Two-stage stochastic programming is a widely applied model for problems where a decision (the first-stage decision) must be made before a random event is realized and another decision (the second stage decision) will be made after the random event is realized. The fundamental idea behind the two-stage stochastic model is the concept of recourse. In the context of the problem studied here, the recourse action is a pumping policy for the active wells (the first-stage decision) after the conductivity of the porous media is realized by drilling wells. By searching for an optimal location decision and a collection of pumping policies in response to conductivity scenarios, it aims to maximize the expected return less the expected pumping and construction costs subject to the following design criteria:

- The construction cost is constrained by a financial budget.
- The number of wells is constrained by limited construction resources.
- Both horizontal and vertical wells are possible with advanced directional drilling. Vertical wells are cheaper but horizontal wells are more efficient.
- Over pumping must be controlled to avoid excessive water table drop.
- Some locations are not considered viable for construction of water wells due to environmental concerns or technical difficulties. A set of candidate locations are given.

Because stochastic programs can be computationally expensive to solve, simpler approaches are sometimes opted for. A widely used approach is to replace all the data with their expected values and solve a smaller deterministic problem. This is called the expected value problem (*EV*). The expected profit obtained from implementing the *EV* solution in every conductivity scenario is *EEV*. The expected profit obtained from the two-stage stochastic model is denoted *RP*. The maximum profit in each conductivity scenario can be solved individually and the expected profit is denoted *WS*. In stochastic programming, for a maximum problem, a basic inequality states that $EEV \le RP \le WS$ [5]. Therefore, the expected value of the *EV* solution *EEV* is a lower bound on the stochastic solution *RP* and the *WS* solution is an upper bound on the stochastic solution. In practice, the *EV* problem should be solved first and the solution for optimal well locations is computed. Furthermore, the *EEV* solution and the *WS* solution are computed. If EEV is close to WS, it means that the uncertainty is not significant in the model. While the stochastic model might increase the computational cost and time significantly, the *EV* solution should be accepted as the business plan. On the other hand, if EEV is far below WS, the stochastic model is recommended to make a better decision on well locations in order to compute the improved solution *RP*.

2 The Expected Value Problem

Before formulating the model, the conductivity over a confined aquifer is dealt with by creating a set of conductivity scenarios *W*. See Appendix A for more details. Once the conductivity scenarios are generated, the response coefficients at head observation points due to pumping at candidate well locations are simulated. The expected coefficients $\overline{a}_{ij}$ are computed as

$$\overline{a}_{ij} = \sum_{w \in W} p_w a_{wij} \ ,$$

where p_w is the probability of each conductivity scenario occurring and a_{wij} is the response coefficient at head observation point *i* due to pumping at candidate well location *j* in each scenario $w \in W$.

The deterministic expected value problem is formulated as follows:

Indices:

i: Head observation point

j: Candidate well location

Parameters:

h_i^0	=	Initial head at location *i*
α_i	=	Percentage maximum head drop at location *i*
Q^l	=	Lower bound on total pumping rates
Q^u	=	Upper bound on total pumping rates
q_j^l	=	Lower bound on pumping rate at location *j*
q_j^u	=	Upper bound on pumping rate at location *j*
N^l	=	Lower bound on the number of active wells
N^u	=	Upper bound on the number of active wells
f_j	=	Construction cost at location *j*
c_j	=	Pumping cost per unit flow at location *j*
$\overline{a}_{ij}$	=	Response coefficients in the expected scenario
b	=	Financial budget
t	=	Total operating time in seconds every year
r	=	Conversion factor between the future return and the net present value
sp	=	Water selling price per cubic meter

Decision variables:

$\overline{x}_j$	=	Candidate location decision variable at location *j*
$\overline{h}_i$	=	Head at location *i* in the expected scenario
$\overline{q}_j$	=	flow at location *j* in the expected scenario

Objective:

$$\text{Max } z = -\sum_{j \in J} f_j \overline{x}_j + \sum_{j \in J} (sp - c_j)\overline{q}_j rt \tag{2.1}$$

Constraints:

$$\sum_{j \in J} f_j \overline{x}_j \le b \qquad \forall j \in J \tag{2.2}$$

$$N^l \le \sum_{j \in J} \overline{x}_j \le N^u \qquad \forall j \in J \tag{2.3}$$

$$q_j^l \overline{x}_j \le \overline{q}_j \le q_j^u \overline{x}_j \qquad \forall j \in J \tag{2.4}$$

$$Q^l \le \sum_{j \in J} \overline{q}_j \le Q^u \qquad \forall j \in J \tag{2.5}$$

$$\bar{h}_i = h_i^0 + \sum_{j\in J} \bar{a}_{ij}\bar{q}_j \qquad \forall i\in I, \forall j\in J \tag{2.6}$$

$$\bar{h}_i \geq (1-\alpha_i)h_i^0 \qquad \forall i\in I \tag{2.7}$$

$$\bar{x}_j \in \{0,1\},\ \bar{q}_j \geq 0,\ \bar{h}_i \geq 0 \qquad \forall i\in I, \forall j\in J \tag{2.8}$$

The objective function (2.1) is to maximize the revenue over the time horizon less the pumping and construction cost. The well location decision variable $\bar{x}_j$ is constrained by the financial budget (2.2) and the construction resource (2.3). The pumping rate variable $\bar{q}_j$ is constrained by the allowable range of pumping rates (2.4) and the range of the total pumping rates (2.5). The head variable $\bar{h}_i$ is dependent on the pumping rate variable $\bar{q}_j$ (2.6) and subject to the maximum head drop (2.7) at head observation point *i*. Constraints (2.8) impose the integrality requirements on the well location decision variables, and the non-negativity requirements on the head variables and pumping rate variables.

Once the expected value solution $\bar{x}$ is calculated, we are in a position to decide whether to accept this solution or not. The decision is made based on the difference between *EEV* and *WS*. The *EEV* solution is the expected profit obtained by implementing the solution $\bar{x}$ and solving for the optimal pumping policy in every conductivity scenario. The *WS* solution is the expected profit obtained by solving for the optimal well locations and the pumping policy in each conductivity scenario.

3 The Two-Stage Stochastic Model

In this section, the two-stage stochastic model is formulated. As the stochastic model requires more data and computational time, it is only recommended when the expected value solution $\bar{x}$ does not produce a satisfactory result, i.e. *EEV* greatly differs from *WS* .

In the context of pumping water, the first-stage decision is to determine which well locations to use in each scenario, i.e. active wells. The second-stage decision is to determine the pumping policy for the active wells in each scenario.

The complete formulation is as follows:

Indices:

i: Head observation point

j: Candidate well location

w: Conductivity scenario

Parameters:

h_i^0 = Initial head at location *i*

α_i = Percentage maximum head drop at location *i*

Q^l = Lower bound on total pumping rates

Q^u = Upper bound on total pumping rates

q_j^l = Lower bound on the pumping rate at location *j*

q_j^u = Upper bound on the pumping rate at location *j*

N^l = Lower bound on the number of active wells

N^u = Upper bound on the number of active wells

f_j = Construction cost at location j

c_j = Pumping cost per unit flow at location j

a_{wij} = Response coefficients in the scenario w

b = Financial budget

t = Total operating time in seconds every year

r = Conversion factor between the future return and the net present value

sp = Water selling price per cubic meter

Decision variables:

x_j = Candidate well decision variable at location j

h_{wi} = Head at location i in the scenario w

q_{wj} = Pumping rate at location j in the scenario w

Objective:

$$\text{Max } z = -\sum_{j\in J} f_j x_j + \sum_{w\in W} p(w)\{\sum_{j\in J}(sp - c_j) q_{wj} rt\} \tag{3.1}$$

Constraints:

$$\sum_{j\in J} f_j x_j \le b \qquad \forall j\in J \tag{3.2}$$

$$N^l \le \sum_{j\in J} x_j \le N^u \qquad \forall j\in J \tag{3.3}$$

$$q^l_j x_j \ge q_{wj} \ge q^u_j x_j \qquad \forall j\in J, \forall w\in W \tag{3.4}$$

$$Q^l \le \sum_{j\in J} q_{wj} \le Q^u \qquad \forall j\in J, \forall w\in W \tag{3.5}$$

$$h_{wi} = h^0_i + \sum_{j\in J} a_{wij} q_{wj} \qquad \forall j\in J, \forall w\in W, \forall i\in I \tag{3.6}$$

$$h_{wi} \ge (1-\alpha_i)\times h^0_i \qquad \forall w\in W, \forall i\in I \tag{3.7}$$

$$x_j \in \{0,1\},\ q_{wj} \ge 0,\ h_{wi} \le 0 \qquad \forall j\in J, \forall w\in W, \forall i\in I \tag{3.8}$$

The objective is to maximize the expected profit *RP* (3.1). The well location decision variable x_j is constrained by the financial budget (3.2) and construction resources (3.3). The pumping rate variable q_{wj} is constrained by the allowable range of pumping rates (3.4) and total pumping rates (3.5). The head variable h_{wi} is dependent on the pumping rate variable q_{wj} (3.6) and constrained by the maximum head drop (3.7). Constraints (3.8) impose the integrality requirements on the well location decision variables, and the non-negativity requirements on the head variables and pumping rate variables.

4 A Case Study

In this section, we use a sample problem to test both the expected value model and the two-stage stochastic model.

4.1 Problem

A water company recently secured a grant on a piece of land to construct water wells. Water will be supplied to local households and commercial users at a price of \$1.45 per cubic meter. The pumping cost of one cubic meter is \$0.28. A set of candidate well locations and associated pump types permitted to construct water wells are given. Some locations are not considered viable for construction of water wells due to environmental concerns or technical difficulties. A financial budget of \$1,500,000 has been allocated to this project. The company's resource consent limits construction to at most six wells. A horizontal well costs \$300,000 and a vertical well costs \$100,000.

The water company would like to use the models created in this project to find the optimal construction locations in order to maximize the expected profit over 10 years. The profit will be converted to a net present value. The interest rate is 8%.

4.2 Conductivity scenarios

Ten equally likely conductivity scenarios with equal probability are created using a conductivity estimation algorithm. See Appendix A for more information.

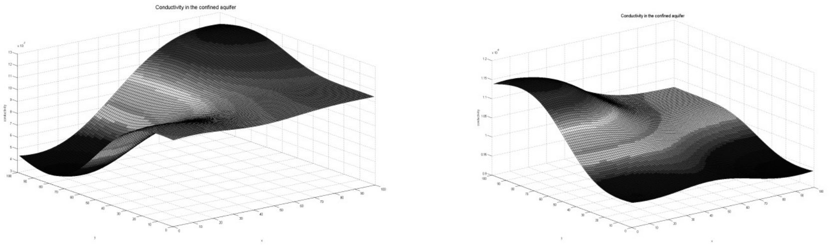


Figure 1: Two conductivity scenarios

4.3 Values of parameters

The minimum and maximum pumping rates of a well are provided by the construction company. The minimum pumping rate is $100m^3/day$ and the maximum pumping rate is $6000m^3/day$. The operating time is 90% of a full day. Converting to SI units, $q_j^{\,l} = 0.00126m^3/s$ and $q_j^{\,u} = 0.07m^3/s$.

The initial hydraulic head over this confined aquifer is $500m$. The maximum head drop at observation points is $100m$, i.e $\alpha_i = 0.2 \;\; \forall i \in I$.

The values for the minimum and maximum total stress rates are not available therefore the lower and upper bounds on total stress rates are dropped for this problem.

The selling price sp is \$1.45. The pumping cost c is \$0.28. The construction cost f of a horizontal well is \$300,000. The construction cost f of a vertical well is \$100,000. The maximum number of active wells N^u is 6. The financial budget b is \$1,500,000. The operating time per day is 90% of 24 hours.

4.4 Expected value problem

The expected value problem was first solved to obtain the *EV* solution $\bar{x}$ which is shown below in Figure 3. The *WS* solution is \$114,155,000 and the *EEV* solution is \$109,747,000. The gap between the WS solution and the EEV solution is \$4,408,000. This result suggests the two-stage stochastic model might produce a better solution.

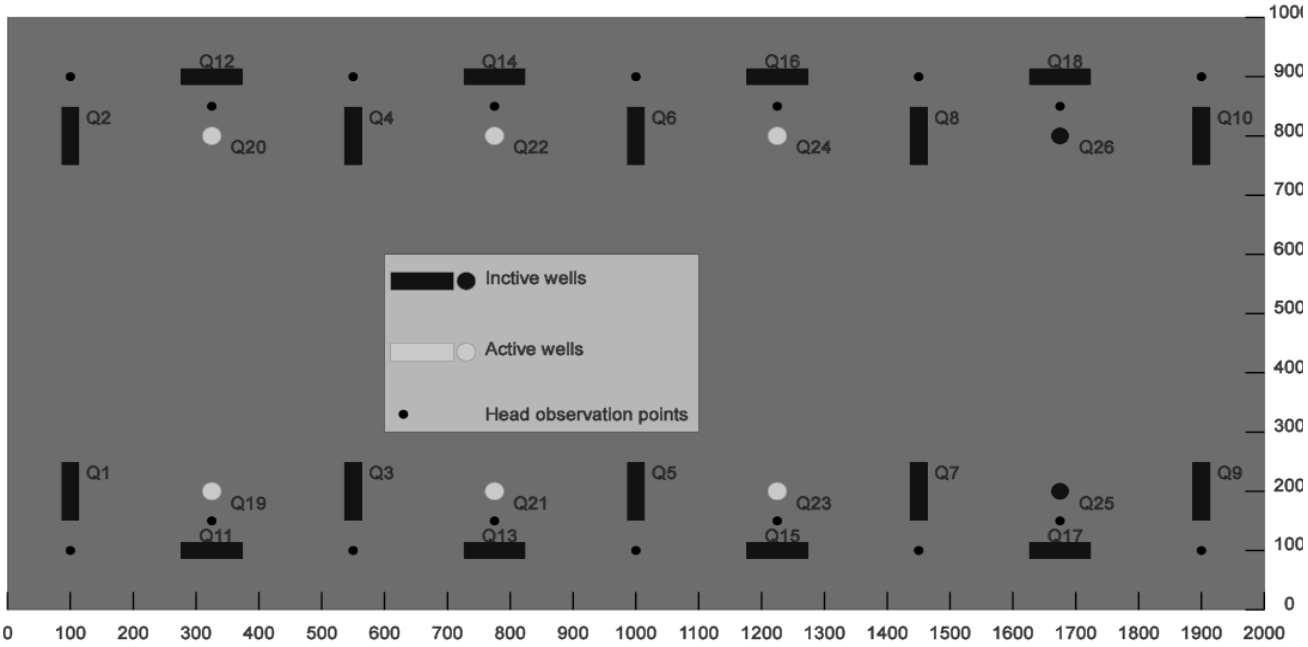


Figure 2: Optimal well locations obtained from the expected value problem

4.5 Two-stage stochastic model

The two-stage stochastic model was implemented with the aim of finding a better solution. The optimal well locations obtained from this model are shown in Figure 4. The *RP* solution is $113,435,000. Compared to the *EEV* solution, this solution increases the expected profit by $3,718,000. This result suggests that the two-stage stochastic model generates a better solution by considering every conductivity scenario. The optimal well locations are better hedged against the uncertainty in the conductivity of the porous media.

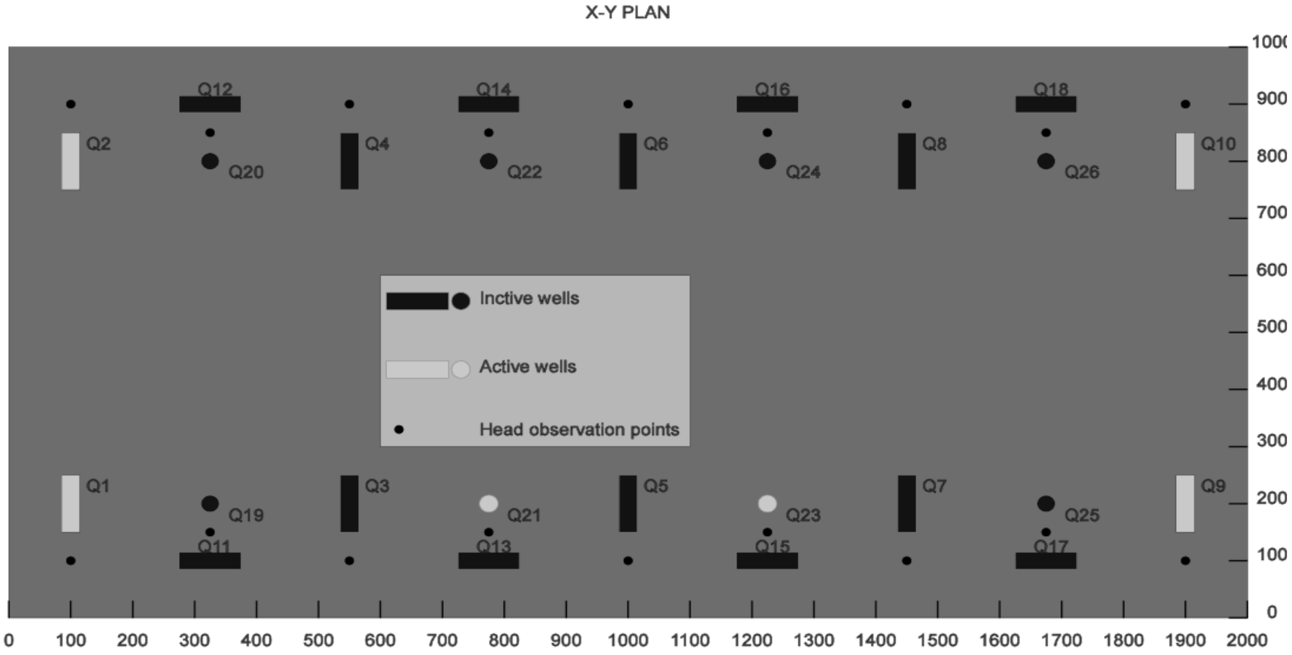


Figure 2: Optimal well locations obtained from the stochastic model

5 Conclusions

In conclusion, an optimization tool is developed to choose optimal well locations in a confined aquifer in order to maximize the expected profit subject to uncertainty in the conductivity over a confined aquifer. The first method is to take the expected conductivity scenario and solve for the optimal well locations. If the *EEV* solution associated with these optimal well locations greatly differs from the *WS* solution, the two-stage stochastic model is implemented with the desire of producing a better solution; if the *EEV* solution is close to the *WS* solution, these optimal well locations are accepted. A case study showed that the uncertainty in the conductivity over a confined aquifer would have a significant impact on the choice of optimal well locations.

Appendix A Generating Conductivity Scenarios

A conductivity scenario is generated as the follows:

- Four points, p_k, *k*=1,2,3,4, are chosen over the confined aquifer

$$x_1 = 100 \quad x_2 = 100 \quad x_3 = 900 \quad x_4 = 900$$
$$y_1 = 800 \quad y_2 = 900 \quad y_3 = 100 \quad y_4 = 900$$
$$z_1 = z_2 = z_3 = z_4 = 50$$

- In each scenario *w*, the conductivity c_{wk} at each point *k* is randomly generated from its corresponding triangular distribution. A triangular distribution is defined by three parameters a, b and c. The values for a, b and c for each point is given in the following table. The unit is m/s.

Location	a	b	c
p1	0.0008	0.24	0.08
p2	0.000005	0.02	0.5
p3	0.000001	1	0.00002
p4	0.00001	1	0.358

The conductivity value at any other point is computed using the following algorithm

$$d_1 = (x - x_1)^2 + (y - y_1)^2$$
$$d_2 = (x - x_2)^2 + (y - y_2)^2$$
$$d_3 = (x - x_3)^2 + (y - y_3)^2$$
$$d_4 = (x - x_4)^2 + (y - y_4)^2$$

$$r_1 = \frac{1/d_1}{T}$$

$$r_2 = \frac{1/d_2}{T}$$

$$r_3 = \frac{1/d_3}{T}$$

$$r_4 = \frac{1/d_4}{T}$$

$$c_{m,x,y} = r_1 c_{w1} + r_2 c_{w2} + r_3 c_{w3} + r_4 c_{w4}$$

References

1. Environment Canterbury (2007) “Groundwater”. Retrieved 2007 from http://www.ecan.govt.nz/Our+Environment/Water/Groundwater/
2. The Groundwater Foundation (2007) “The Watercycle”. Retrieved 2007 from http://www.groundwater.org/kc/gwwatercycle.html
3. David P.A., Mulligan A.E. (2000) Optimal management of flow in groundwater systems. Academic Press
4 “Comsol user manual”. Retrieved 2007 from http://www.comsol.ch/
5. Birge.J.R, Louveaux.F (1997) Introduction to stochastic programming. Springer

Game Theory Applications in Operations Management

Gérard Cachon
The Wharton School
University of Pennsylvania
Phildelphia, PA USA

Abstract

This talk will provide an overview of how game theory is currently being applied to problems in operations management, including coordination within supply chains, competition across horizontal competitors and revenue management with strategic consumers. Successes and limitations of the methodology will be discussed, leading to suggestions for future research directions.

Gérard Cachon is Fred R. Sullivan Professor of Operations and Information Management, The Wharton School, University of Pennsylvania, where he has been since 2000. He has won numerous teaching awards at Penn and his former institution Duke University. He is coauthor with C. Terwiesch of *Matching Supply with Demand: An Introduction to Operations Management*, McGraw-Hill, 2005. He is visiting the University of Auckland Department of Engineering Science until June 2008.

Optimization of Product Placement in a Retail Environment

Garrett van Ryzin
Columbia University
New York NY USA
(joint work with Wei Ke, Columbia University)

Abstract

We consider the impact of physical locations on the sales of a product assortment. We propose some heuristics to solve the placement optimization problem that combine the subjective rules of human planners with optimization modeling. Extensions of the technique and on-going work on placement optimization are discussed.

Siren Live – Software for Realtime Optimised Ambulance Redeployment

Jeff Meyer
The Optima Corporation
Auckland, New Zealand
j.meyer@TheOptimaCorporation.com

Andrew Mason
Dept of Engineering Science
University of Auckland
New Zealand
a.mason@auckland.ac.nz

Abstract

In 2001, the ambulance service in Melbourne approached the University of Auckland seeking software that could simulate their ambulance operations. By starting from an early research tool developed by Andrew Mason and Shane Henderson, this software was duly delivered as part of a collaborative effort between the University and The Optima Corporation. Thanks to significant ongoing development and marketing efforts by Optima, this Siren Predict software is now being used (or being implemented) in Australia, Britain, Canada and Denmark.

Siren Predict is a simulation tool that enables different scenarios to be modelled and explored for planning purposes. As a simulation, Siren Predict contains sufficient decision making capabilities to mimic typical operational procedures such as deciding which vehicles to send to a call or what hospital to transport a patient to.

Siren Live is the first of a suite of tools being developed by Optima that will deliver more advanced real-time decision making capabilites to the providers of Emgergency Medical Services. Siren Live is an integrated system that receives real time updates of ambulance position and status, and uses this information to analyse the current capability to quickly respond to potential future calls. When vehicle coverage is poor in an area, Siren Live recommends that a *redeployment* operation be conducted in which idle vehicles are moved from one waiting location to another to improve overall coverage. Siren Live uses an integer programming model to generate these redeployments.

In this presentation, we will present a demo of the Siren Live system and discuss our experiences in implementing this operations research technology in Canada and Australia.

An iterative approach to integrated aircraft routing, crew pairing and flight re-timing

Oliver Weide, David Ryan and Matthias Ehrgott
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 planning problems aircraft routing, crew pairing and flight re-timing. Aircraft and crew must be allocated to flights in a schedule in a cost minimal way. Additionally, the departure times of some flights are allowed to vary within some time window.

Although these problems are not independent, they are usually formulated as independent mathematical optimization models and solved sequentially. This approach might lead to suboptimal allocation of aircraft and crew.

We solve the aircraft routing and crew pairing problem with an iterative approach, alternately solving aircraft routing and crew pairing. This approach generates a series of low cost solutions that are also robust to disruptions that may occur during operations. The solutions show large improvements in terms of cost and robustness compared to the sequential solution. They also show the trade-off between cost and robustness enabling the choice of a preferred solution for operation.

We present an approach to allow flexibility for the departure times of the flights within the iterative approach. We demonstrate on real world data sets how effectively time windows further improve the robustness of the solutions.

In airline scheduling a sequence of planning problems must be solved (see e.g. Klabjan (2005)): First, marketing decisions in the schedule design problem determine the schedule of flights the airline operates. Each flight is specified by origin, destination, departure date, departure time, and duration. 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. 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. In a similar way to the aircraft routing problem, the crew pairing problem allocates crew to flights in a minimal cost way. A set of generic crew pairings is constructed subject to many rules so that each flight is covered exactly once. The last of the planning problems is crew rostering. Based on the constructed crew pairings, a line of work is assigned to each individual crew member.

We consider three of the above problems: schedule design, aircraft routing, and crew pairing. Instead of constructing a schedule from scratch we only solve a re-timing problem: An original schedule is given and only small changes of the departure times of some flights are allowed. The solution of our approach consists of a set of re-timed flights together with aircraft routings and crew pairings covering all flights of the given schedule.

Traditionally, the three problems are solved sequentially although the problems are interdependent (see Barnhart and Cohn (2004)). During operations delays occur frequently due to late passengers or bad weather, for example. If an incoming flight is late, the next flight the aircraft is operating will be late because turn around times are usually very short for efficiency reasons. If the crew change aircraft after the delayed flight and only minimal ground time is available, the next flight the crew is operating will also be late. Such a propagation of delay can cause serious disruptions to the operation of the flight schedule. The total delay over all flights caused by the initial disruption can be reduced if the crew follow the aircraft as much as possible and change aircraft only when ground time between flights is much longer than the minimum. In this sense, in an operationally more robust solution the crew pairings depend on the given aircraft routings. As a first question it is natural to ask if it is possible to improve the robustness of the crew pairing solution by permitting changes to the aircraft routing solution to encourage the aircraft to follow the crew. Additionally, departure times of the flights are determined a priori in the schedule design problem. Allowing some flexibility of departure times increases the number of possible connections between flights for crew and aircraft. Hence, crew may not be required to change aircraft as many times as for a fixed schedule. If it is still necessary for crew to change aircraft some additional buffer time may be allocated for the crew when they change aircraft in order to compensate for delays. The second question we try to answer is therefore if flexibility in departure time of some flights can further improve the robustness of the aircraft routing and crew pairing solutions.

We investigate the two questions above and show first that it is indeed possible to reduce the cost of the crew pairing solution and simultaneously increase its robustness by considering crew and aircraft together. Additionally, we show that only small changes of departure times can increase the robustness of the solutions even further.

We formulate the schedule re-timing problem, aircraft routing problem, and the crew pairing problem in one integrated model (similarly to Mercier and Soumis (2007)). This model yields one optimal solution for the three problems where the objective function is a weighted sum of cost and some value attached to robustness. However, the model is very hard to solve. Integration increases the complexity of the problems which are already NP-hard individually. Decomposition methods are proposed in the literature to solve the integrated problem but long computation times are needed to solve the model to optimality.

Instead of solving the integrated model, we propose to solve the original problems iteratively. We first consider a fixed schedule. We start with a minimal cost crew pairing solution without taking aircraft routings into account. In each iteration we solve the individual aircraft routing problem first, taking into account the current crew pairing solution. Then, given the aircraft routing solution we resolve the crew pairing problem. We only use the objective functions in both problems to pass information from the problem solved previously to generate more and more robust solutions. Hence the constraints are unaltered and the complexity of the two problems is not increased. We stop the process when the level of robustness can not be improved any further. This procedure generates a series of feasible solutions for the integrated model with varying cost and robustness measure. Subsequently, for each solution we allow an increasing amount of flexibility for the departure times of some flights to improve robustness. We generate improved re-timed solutions with a varying number of re-timed flights. This two step approach allows us to separately evaluate the benefits of the iterative approach and the additional departure time flexibility.

The airline is not required to associate a cost with robustness or re-timings but can study the trade-off between cost, robustness and schedule changes and then choose a solution they prefer to operate.

Applying this approach to various domestic airline schedules we obtain low cost solutions that are highly robust and only require a small amount of departure time changes. The solutions obey all rules imposed by the airline and are ready to be implemented in practice.

References

Barnhart, C., and A.M. Cohn. 2004. "Airline Schedule Planning: Accomplishments and Opportunities." *Manufacturing & and Sevice Operations Management* 6 (1): 3–22.

Klabjan, D. 2005. "Large-scale Models in the Airline Industry." In *Column Generation*, edited by G. Desaulniers, J. Desrosiers, and M.M. Solomon, 163–196. Kluwer Scientific Publishers.

Mercier, A., and F. Soumis. 2007. "An integrated aircraft routing, crew scheduling and flight retiming model." *Computers & Operations Research* 34:2251–2265.

The Train Driver Recovery Problem - a Set Partitioning Based Model and Solution Method

Natalia J. Rezanova
Informatics and Mathematical Modelling
Technical University of Denmark, Denmark
e-mail: njr@imm.dtu.dk

David M. Ryan
The Department of Engineering Science
The University of Auckland, New Zealand
e-mail: d.ryan@auckland.ac.nz

Extended Abstract to 42nd Annual ORSNZ Conference, 2007

Every railway operator experiences disruptions during the daily operations due to external influences or internal failures. A Danish railway operator DSB S-tog A/S is no exception. DSB S-tog A/S operates on an urban train network with at least 6 trains per hour in each direction departing from every station of the network and up to 30 trains per hour in each direction departing from the Copenhagen central station. Minor train delays on the network are recovered by re-establishing the original plan using the slack time built into the timetable or delaying other trains. Major disruptions in the train schedule are recovered by re-routing or cancelling trains. A train is re-routed if it is turned back before reaching the end terminal station or driven through some stations without stopping. A cancellation is applied either to a single train task or to a whole train line, resulting in cancellations of all train tasks of a particular line for a certain period of time.

Disruptions in the train timetable affect the train driver schedule. When a train is delayed, re-routed or cancelled, a driver might be late for the next scheduled train task of the duty. If the driver is not available in due time for a train departure, the train task is assigned to another driver. If there is no available driver to cover the train task, the train is delayed or cancelled, causing a propagation of disruptions in the schedule. At the present time the re-scheduling process of disrupted train driver duties is conducted manually. During major disruptions a large number of driver duties has to be re-scheduled and a recovery solution can be difficult to assess.

An optimization method presented here is developed for operational recovery of the train driver schedule in a cooperation with DSB S-tog A/S. The train driver re-scheduling has received a very limited attention by Operations Research practitioners. The crew re-scheduling problem for train driver duties disrupted due to the maintenance work on train tracks is solved by [1] for the largest passenger railway operator in The Netherlands. An integer programming approach to a simultaneous train timetable and crew roster recovery problem, tested on the New Zealands Wellington Metro line, is presented in [4].

For a particular disruption we identify a *disruption neighbourhood*, which is a part of the driver schedule characterized by a set of train tasks and a set of train drivers. The initial disruption neighbourhood is identified by a set of drivers, who's duties contain train tasks which are known to be disrupted within a certain *recovery period*, a time period within which a recovery solution is aimed to be found. A train task is disrupted if it is delayed, cancelled, re-routed or uncovered, i.e. assigned to an absent driver. All train tasks belonging to the initial set of drivers within the recovery period are included into the initial disruption neighbourhood. The Train Driver Recovery Problem (TDRP) aims at finding a set of feasible train driver recovery duties for drivers within the disruption neighbourhood with minimum modification from the original train driver schedule, such that all train tasks within the recovery period are covered and the driver duties outside the recovery period and duties of drivers not included in the disruption neighbourhood are unchanged.

The TDRP is formulated as a set partitioning problem, where variables represent recovery duties of train drivers. The set of generalized upper-bound *train driver constraints* ensure that each train driver is assigned to exactly one recovery duty in the schedule. The *train task constraints* have a set partitioning structure and ensure that each train task in the recovery schedule is covered exactly once. It is observed in [3] that the linear programming relaxation of the set partitioning formulation of the crew rostering problem, which has a similar structure to the TDRP, possesses strong integer properties due to the existence of the generalized upper-bound crew constraints, which contribute to the perfect structure of the submatrix, corresponding to each crew member. This observation implies that the linear programming relaxation of the Train Driver Recovery Problem (TDRP-LP) also possesses strong integer properties.

The solution method for solving the Train Driver Recovery Problem is based on solving the TDRP-LP and finding an integer solution with a constraint branching strategy. Since the cost of the recovery is not determined by a physical cost of the driver schedule (the drivers are already paid to be at work), but rather by the fictitious cost which expresses how attractive each recovery duty is, the *optimality* of the solution is not as important as the *feasibility* of the solution. The TDRP-LP is solved with a column generation method based on a limited subsequence strategy, where recovery duties with negative reduced costs are generated by limiting the number of tasks (subsequences) a driver can perform after finishing any task in the duty. Starting with a small number of subsequences, it is gradually increased, allowing to consider less attractive subsequent tasks for recovery duties. When a feasible solution to the TDRP-LP is found, we consider the problem solved. If the initial number of drivers is not enough to cover all train tasks in the initial disruption neighbourhood, the disruption neighbourhood is *expanded* in two possible ways: either the number of drivers in the disruption neighbourhood is increased by adding available stand-by drivers or the recovery period is extended, including more train tasks from the involved drivers' duties. If the problem remains infeasible due to uncovered train tasks when there are no more available drivers to add to the disruption neighbourhood, the decision support system sends an infeasibility message to the dispatcher specifying which train tasks are uncovered and hence have to be delayed or cancelled.

If the solution to the TDRP-LP is fractional, a constraint branching strategy similar to the one described in [2] is applied in order to find an integer solution. Since every train driver submatrix in the set partitioning formulation of the problem is perfect, the fractions occur in the TDRP-LP only across train drivers' blocks of columns. It is therefore sensible on 1-branches of the Branch & Bound tree to force one driver r to cover a train task s, which also appears in another driver's optimal recovery duty while forbidding other drivers to include s in their recovery duties. On the 0-branch we forbid the driver r to cover the train task s. A depth-first search on 1-branches of the Branch & Bound tree is implemented and the branching procedure is terminated as soon as the first integer solution is found.

We generate test scenarios based on historical data from one day of DSB S-tog A/S operations in January 2007, when the operations were severely disrupted due to a broken switch near the central station. Recovery periods between 1 and 3 hours were considered. Test results show that for the test instances involving between 10 and 40 train drivers and between 20 and 50 train tasks the solutions are found within 1–3 seconds. For the test instances involving between 40 and 70 drivers and between 50 and 80 train tasks the solutions are found within 20–40 seconds. For the test instances involving between 70 and 90 drivers and between 80 and 120 train tasks the solutions are found within 1–2 minutes. The majority of all test instances (ca. 98%) produces integer solutions to the linear relaxation of the TDRP, confirming the strong integer property of the problem formulation.

References

[1] D. Huisman, *A column generation approach for the rail crew re-scheduling problem*, European Journal of Operational Research, 180 (2007), pp 163–173.

[2] D. M. Ryan, *The solution of massive generalized set partitioning problems in aircrew rostering*, The Journal of the Operational Research Society, 43 (1992), pp 459–467.

[3] D. M. Ryan, J. C. Falkner, *On the integer properties of scheduling set partitioning models*, European Journal of Operational Research, 35 (1988), pp 442–456.

[4] C. G. Walker, J. N. Snowdon, D. M. Ryan *Simultaneous disruption recovery of a trian timetable and crew roster in real time*, Computers & Operations Research, 32 (2005), pp 2077–2094.

Rural Postmen, Rewards and Grid Networks

Mark R. Johnston and Stefanka Chukova
School of Mathematics, Statistics and Computer Science
Victoria University of Wellington, New Zealand

{Mark.Johnston,Stefanka.Chukova}@vuw.ac.nz

Abstract

Arc routing problems, such as the Chinese Postman Problem (CPP), the Rural Postman Problem (RPP) and their variants, are cousins of classical vehicle routing problems. For example, the RPP specifies a subset of arcs that are required to be visited by a single vehicle at minimum total cost. Although RPP is NP-hard, simple construction and improvement heuristics have proven successful in applications. The Rural Postman Problem with Rewards (RPP+R) has recently been proposed by Aráoz, Fernández, and Zoltan (2006), in which a subset of arcs carry a reward that is collected only if that arc is traversed by the vehicle, and the objective is to maximize the total reward collected subject to a constraint on the total tour cost. Rewards provide an explicit mechanism for modelling the tradeoff between difficulty (cost) of service and benefit (reward) from service. To date, only polyhedral results have been reported for the RPP+R. In this paper we restrict our study to the behaviour of heuristics on undirected grid networks. These are easy to visualize and are appropriate models for the roading networks in urban areas. We estimate and compare the distribution of performance of local-search based heuristics for the RPP on random undirected grid network problem instances of the CPP defined by a single edge-density parameter. Computational experience shows that local-search based heuristics for RPP behave well enough on CPP, so that it is justifiable that they are applied to other arc routing problems. Visualisation of grid networks has provided useful insights into the behaviour of local-search based heuristics.

1 Introduction, Problem Descriptions and Brief Literature Review

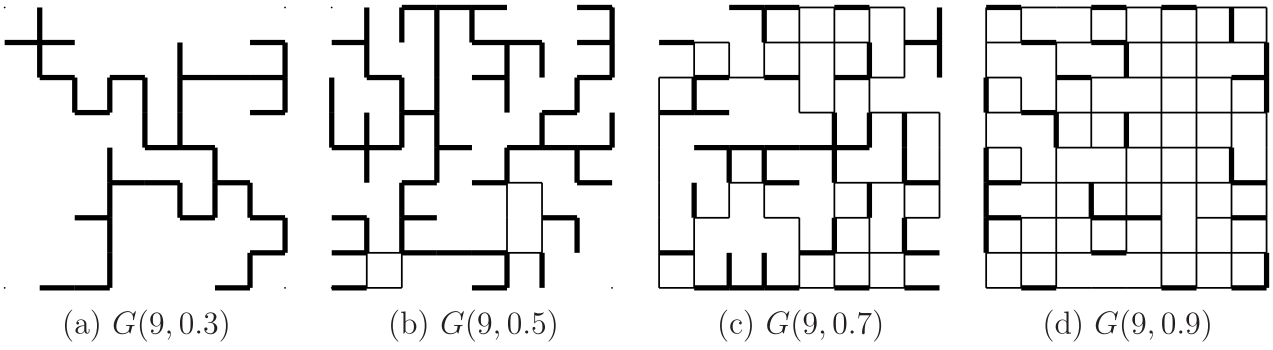
The field of vehicle routing and scheduling is a rich and diverse field, spanning the modelling of complex real world problems and related fine algorithmic details of their efficient optimal solution. The aspects that we find most attractive about this field of study are that the problems are easy to visualise in terms of physical situations (translated onto the computer screen) and model with graph theoretic solution structures.

This paper brings together four ideas: arc routing (the requirement to visit particular edges rather than particular nodes in a network), undirected grid networks (easy to visualise and also study as a whole class of problem instances), design of local-search based heuristics, and the incorporation of rewards.

Idea 1. Arc routing. Arc Routing problems (including Chinese Postman Problem and Rural Postman Problem and their variants) are well studied in the Operations Research literature. For a comprehensive surveys, see the two survey articles of Eiselt, Gendreau, and Laporte (1995a, 1995b), the research collection of Dror (2000), and the textbook of Evans and Minieka (1992). Consider a graph $G = (V, A)$ with node set V, arc set A, and a nonnegative cost matrix $C = (c_{ij})$ associated with A, where the arcs can be undirected, directed or both. The Chinese Postman Problem (CPP) is to determine a minimum-cost closed walk on all arcs of A. The Rural Postman Problem (RPP) only requires a subset $R \subseteq A$ of arcs to be traversed. The major complexity difference is that RPP (both undirected and directed) is NP-hard (Lenstra and Rinooy Kan (1976)), whereas CPP is efficiently solved by the method of Edmonds and Johnson (1973) in polynomial time. In an optimal CPP or RPP solution, no edge is traversed more than twice.

Idea 2. Undirected grid networks. Urban roading networks can be appropriately modelled with grid networks, see, e.g., Frizzell and Giffin (1995). Consider a regular lattice of $n \times n$ equally spaced points, joined with $n(n-1)$ horizontal and $n(n-1)$ vertical undirected edges. The mechanism for creating a random grid problem instance denoted as of type $G(n, p)$ is simply to delete one edge at a time while ensuring that the resulting set of *edges* remain connected, until only $p\,2n(n-1)$ edges remain. Since all values of $0 \leq p \leq 1$ are admissible, it is therefore possible for a node to have degree zero, but all nodes of degree ≥ 1 are connected. We also require that all edges are connected to a central *depot* node from which the tour nominally departs. Small examples corresponding to $n = 9$ and different values of p are given in Figure 1. Note that small values of p have tree-like structures, many nodes of degree zero, and most edges are repeated in the optimal CPP tour. In Section 3.1 we investigate the expected properties of $G(n, p)$ further.

Note that although various collections of test problem instances are available for the arc routing problems, e.g., Eglese and Li (1992) and the websites `http://www.uv.es/~belengue/carp/` and `http://www.iwr.uni-heidelberg.de/groups/comopt/people/theis/GRPLIB/index.html`, there is often insufficient information



(a) $G(9, 0.3)$ (b) $G(9, 0.5)$ (c) $G(9, 0.7)$ (d) $G(9, 0.9)$

Figure 1: Example grid networks, where bold indicates that the edge would be traversed twice in an optimal CPP tour.

available to be able to visualise the networks. By contrast, grid networks are very easy to visualise, can be varied by a single edge density parameter p, and each node has degree at most four.

Idea 3. Local search. Given a current solution $\boldsymbol{x}$, a local search method searches a *local neighbourhood* of solutions $\mathcal{N}(\boldsymbol{x})$ that are in some sense close to $\boldsymbol{x}$. Local search has been proven successful for vehicle routing problems and other combinatorial optimization problems, see, e.g., Aarts and Lenstra (1997) and Toth and Vigo (2002). The area of research remains very active, e.g., the recent advances in large-neighbourhood search of Ahuja et al. (2002) are finding numerous successful applications.

Idea 4. Rewards. Rewards provide an explicit mechanism for modelling the tradeoff between difficulty (cost) of service and benefit (reward) from service. The prototypical example goes as follows. A salesman collects a reward in every city j that he visits and travels between cities i and j at cost c_{ij}. The salesman must determine both which subset of cities to visit and also in which order to visit those cities, i.e., these are two interdependent decisions. In the Orienteering Problem (OP), the salesman wishes to maximize the reward value collected without exceeding a prescribed travel cost budget, visiting each city at most once. Tsiligirides (1984) appears to be the first to consider the combinatorial optimization problem formulation now known as the OP, although he called it a Generalised TSP. Golden, Levy, and Vohra (1987) were the first to coin the name *Orienteering Problem* and also showed that OP in NP-hard. The recent survey paper of Feillet, Dejax, and Gendreau (2005) calls the general class of these problems Profitable Tour Problems (PTPs).

Aráoz, Fernández, and Zoltan (2006) introduced the Privatized Rural Postman Problem in which each required edge in the Rural Postman Problem has an associated reward. The objective is, as in OP, to maximize the reward value collected without exceeding a prescribed travel cost budget. We prefer to call this the Rural Postman Problem with Rewards (RPP+R) to emphasise the rewards. To the best of our knowledge, only polyhedral results have been reported for the RPP+R, i.e., no algorithm or computational results have been reported.

Research questions. The goal of this research is to evaluate whether local-search can be effective for RPP+R. This paper addresses one small step in this project, i.e., to evaluate whether local search is sufficiently effective for RPP to conclude that it is likely to be worth pursuing for RPP+R. We restrict our analysis to undirected grid networks of the type described above.

Outline of this paper. The remainder of the paper is structured as follows. Section 2 designs a local-search based heuristic for RPP. Section 3 designs and conducts a computational experiment to establish some expected properties of undirected grid networks and to evaluate how these local-search based heuristics perform on the CPP. Finally, Section 4 offers some conclusions and recommendations for future research, including some thoughts on how to introduce rewards into the mix.

2 Design of Local Search Heuristics for the Rural Postman Problem

In order to specify a local search heuristic to solve the RPP, we must describe the data structure used to model a RPP tour, the local operators that can make small local changes to improve a given RPP tour, a construction heuristic to create an initial RPP tour, and some improvement logic for how to apply local operators to find a local-optima.

Data structure. We adopt the data structure of Hertz, Laporte, and Mittaz (2000) in their tabu search heuristic for the Capacitated Arc Routing Problem (CARP). Here a RPP tour is a sequence of oriented required edges. The shortest path between required edges is traversed to determine the tour's length. The representation of a rural postman tour in this way is very efficient, especially when the required edges are sparse. It also ensures the connectivity of the RPP tour.

Local operators. Figure 2 depicts the symbols used to define the local search operators for RPP in Figure 3. Figure 2(a) represents a required edge (thick), (b) represents two required edges (thick) joined by a shortest path (thin) between the given endpoints which consists of zero or more nonrequired or required (deadheaded) edges, (c) represents a sequence of one or more required edges joined by shortest paths, i.e., (c) is defined recursively as either (a) or (e), and finally in (d) the square box represents the required depot node and (d) is defined recursively as either the depot node alone or (f).

Figure 3 then *defines* the local operators applied in local-search based heuristics for the RPP. Here, TWO-EXCHANGE and THREE-EXCHANGE are used to improve the tour length. These edge exchanges are simply those of Lin and Kernighan (1973) for TSP modified to ensure that no required edge is broken. Insertion operators INSERTION and GENERALISED-INSERTION are used in tour construction. Here GENERALISED-INSERTION inserts an new required edge, together with a local reoptimisation of the tour. This is modified from Gendreau, Hertz, and Laporte (1992) which proved very effective for TSP.

Construction heuristics. Algorithm 1 describes a FARTHEST-INSERTION heuristic for RPP, based upon the TSP heuristic of Rosenkrantz, Stearns, and Lewis (1977). Here insertion uses the best INSERTION and GENERALISED-INSERTION operators. By changing 'argmax' to 'argmin' we obtain a CHEAPEST-INSERTION heuristic. Informal computational experience suggests that FARTHEST-INSERTION performs consistently slightly better than CHEAPEST-INSERTION and so we have not formally included CHEAPEST-INSERTION in our experiments in Section 3. Note also that we have not

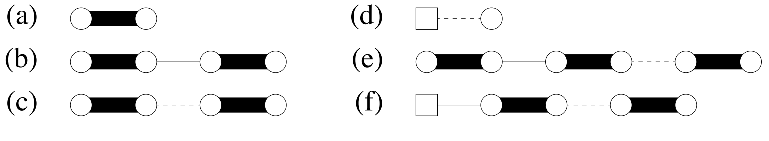
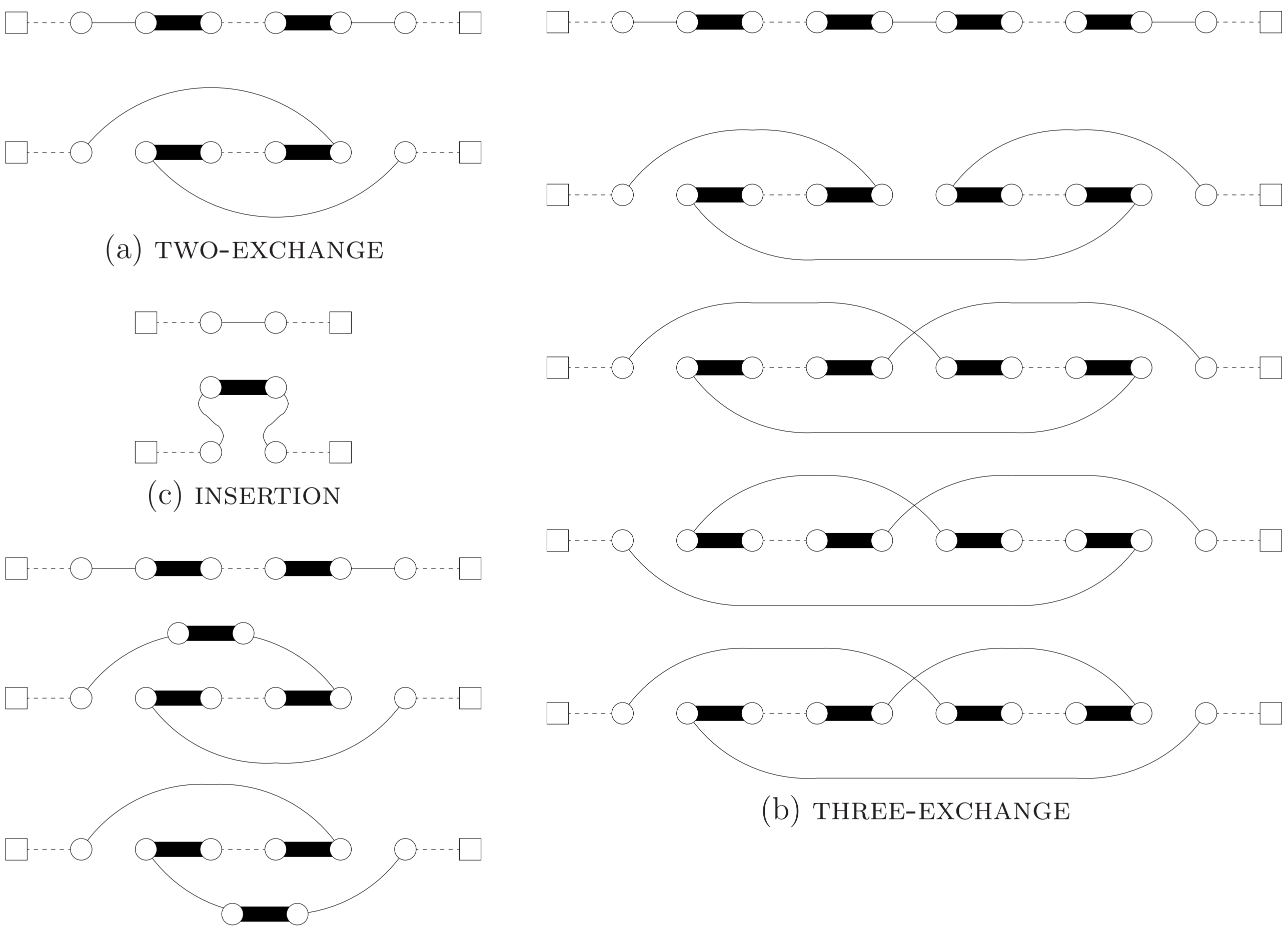


Figure 2: Definition of symbols



(a) TWO-EXCHANGE

(c) INSERTION

(d) GENERALISED-INSERTION

(b) THREE-EXCHANGE

Figure 3: Local search operators for the RPP.

```
Algorithm 1 function FARTHEST-INSERTION
  $\tau \leftarrow (depot, depot)$
  while (some required edges remain) do
      // Determine best way to insert each remaining required edge
      for each remaining required edge $e$ do
          $c(e) \leftarrow$ minimal insertion cost of $e$
      end
      // Farthest insertion
      Insert edge $e^* \leftarrow \text{argmax}\{c(e)\}$
  end
  return($\tau$)
end
```

implemented the construction heuristic of Frederickson (1979) (which is based on Christofides' heuristic for the TSP).

Improvement logic. The improvement logic used for RPP local-search is simply to apply the best-improving two-exchange or three-exchange until no further improvement is possible. It is important to note, but slightly more difficult to prove, that together two-exchange and three-exchange are sufficient to ensure that no edge is traversed more than twice in an RPP tour.

3 Computational Experiment on the Chinese Postman Problem

Edmonds and Johnson (1973) provide a polynomial time algorithm to solve CPP. The critical step is solving a minimum-weight matching problem on shortest path distances between *odd-degree* vertices of the network in order to determine which edges should be traversed twice in order to create an Euler tour; see Evans and Minieka (1992) for a clear explanation of the algorithm with examples.

3.1 Expected Properties of Grid Networks

Since the Edmonds and Johnson (1973) algorithm involves matching odd-degree vertices, we estimate the expected degrees of nodes in grid networks of different sizes and densities. $G(n, 1)$ has 4 nodes of degree 2 (the four corners), $(n-2)^2$ nodes of degree 4 (the internal nodes) and $4(n-2)$ nodes of degree 3 (the sides), adding to a total of n^2 nodes. Simulations provide additional results as shown in Table 1. As noted earlier, it is possible to have nodes of degree zero. Table 1 shows a trend in the number of nodes of degree 2 (these are in a sense redundant nodes in CPP). The interesting property is the expected number of odd-degree nodes in $G(n, p)$ which peaks around $p = 0.7$, as summarised in Table 2. This indicates that the difficulty of *solving* CPP on a grid network of fixed n is roughly the same for $0.6 \leq p \leq 0.9$.

Table 1: Average number of nodes of degree $(0, 1, 2, 3, 4)$ on grid types $G(n, p)$ for 1000 problem instances of each (simulation).

p	$n = 7$	$n = 9$	$n = 11$
0.1	`(40.1, 2.4, 6.1, 0.4, 0.0)`	`(66.0, 2.9,11.3, 0.8, 0.0)`	`(98.0, 3.3,18.4, 1.3, 0.0)`
0.2	`(30.9, 3.8,12.5, 1.7, 0.1)`	`(51.0, 5.0,22.2, 2.7, 0.1)`	`(76.1, 6.4,34.3, 4.1, 0.1)`
0.3	`(23.0, 5.7,16.8, 3.3, 0.2)`	`(37.0, 8.7,29.0, 5.9, 0.4)`	`(54.0,12.5,44.5, 9.4, 0.6)`
0.4	`(14.0, 8.9,19.6, 5.9, 0.6)`	`(22.2,14.1,33.4,10.3, 1.0)`	`(32.2,21.1,49.9,16.2, 1.6)`
0.5	`( 6.9,11.3,20.8, 8.7, 1.3)`	`(10.0,18.4,34.5,15.6, 2.5)`	`(13.7,27.3,51.2,24.7, 4.1)`
0.6	`( 2.9,10.4,20.4,12.6, 2.7)`	`( 3.9,16.0,32.7,22.8, 5.6)`	`( 5.2,22.7,47.6,36.0, 9.5)`
0.7	`( 1.1, 6.9,17.9,17.3, 5.8)`	`( 1.4,10.1,27.9,30.1,11.5)`	`( 1.9,14.0,40.1,46.1,18.9)`
0.8	`( 0.4, 3.7,14.4,20.6, 9.9)`	`( 0.5, 5.1,21.2,34.4,19.8)`	`( 0.5, 6.6,29.1,51.9,32.9)`
0.9	`( 0.0, 1.2, 9.0,22.2,16.6)`	`( 0.1, 1.5,12.1,34.9,32.4)`	`( 0.1, 1.9,15.9,50.2,52.9)`
1.0	`( 0.0, 0.0, 4.0,20.0,25.0)`	`( 0.0, 0.0, 4.0,28.0,49.0)`	`( 0.0, 0.0, 4.0,36.0,81.0)`
	49 nodes	81 nodes	121 nodes

Table 2: Summary of the expected number of odd/even degree nodes and the actual number of edges (in brackets).

p	$n = 7$	$n = 9$	$n = 11$
0.1	2.8/ 6.1 (8)	3.7/11.3 (14)	4.6/18.4 (22)
0.2	5.5/12.6 (17)	7.7/22.3 (29)	10.5/34.4 (44)
0.3	9.0/17.0 (25)	14.6/29.4 (43)	21.9/45.1 (66)
0.4	14.8/20.2 (34)	24.4/34.4 (58)	37.3/51.5 (88)
0.5	20.0/22.1 (42)	34.0/37.0 (72)	52.0/55.3 (110)
0.6	23.0/23.1 (50)	38.8/38.3 (86)	58.7/57.1 (132)
0.7	24.2/23.7 (59)	40.2/39.4 (101)	60.1/59.0 (154)
0.8	24.3/24.3 (67)	39.5/41.0 (115)	58.5/62.0 (176)
0.9	23.4/25.6 (76)	36.4/44.5 (130)	52.1/68.8 (198)
1	20.0/29.0 (84)	28.0/53.0 (144)	36.0/85.0 (220)
	49 nodes	81 nodes	121 nodes

3.2 Evaluation of Farthest Insertion and Local Search

For the main computational experiment, we consider only the Chinese Postman Problem on $G(n, p)$ with unit edge weights, i.e., all edges are required to be visited at least once. Note that in an optimal CPP solution, no edge is traversed more than twice, and hence the objective is to minimize the number of edges traversed a second time (called "deadheading"). The CPP is the *hardest* RPP on which to test heuristic methods that are designed for RPP, since CPP is the most dense RPP.

The critical step of the Edmonds and Johnson (1973) algorithm is solving a minimum-weight matching problem on shortest path distances between *odd-degree* vertices. We denote this as the OPTIMAL-MATCHING weight. Now consider a RPP tour found by FARTHEST-INSERTION or LOCAL-SEARCH. Subtracting the number of required edges from the total length travelled gives a weight directly comparable to the OPTIMAL-MATCHING weight. In other words, computational results report only to total deadheaded weight.

Experimental Results. Figure 4 presents the overall computational results conducted on undirected grid networks of size $n = 9$ with $0.5 \leq p \leq 0.9$. For values of $p < 0.5$ the networks are very close to trees and hence almost all edges are traversed twice. In Figure 4, the left, middle and right sets of five boxplots correspond to OPTIMAL-MATCHING, FARTHEST-INSERTION and LOCAL-SEARCH respectively. The OPTIMAL-MATCHING weight decreases with p. This trend also occurs with LOCAL-SEARCH but not with FARTHEST-INSERTION. For $p = 0.5$, LOCAL-SEARCH does not improve upon FARTHEST-INSERTION, but OPTIMAL-MATCHING is significantly better (due to judicious use of cycles in the almost acyclic network). For $p = 0.6$, LOCAL-SEARCH slightly improves FARTHEST-INSERTION, and performs similarly to OPTIMAL-MATCHING. For $p = 0.7$, there is good improvement of LOCAL-SEARCH over FARTHEST-INSERTION and comparable to OPTIMAL-MATCHING. For $p \in \{0.8, 0.9\}$, there is substantial improvement of LOCAL-SEARCH over FARTHEST-INSERTION but falling well behind OPTIMAL-MATCHING.

In conclusion, as p increases, LOCAL-SEARCH makes significant improvements upon FARTHEST-INSERTION. Also, LOCAL-SEARCH performs comparably to OPTIMAL-

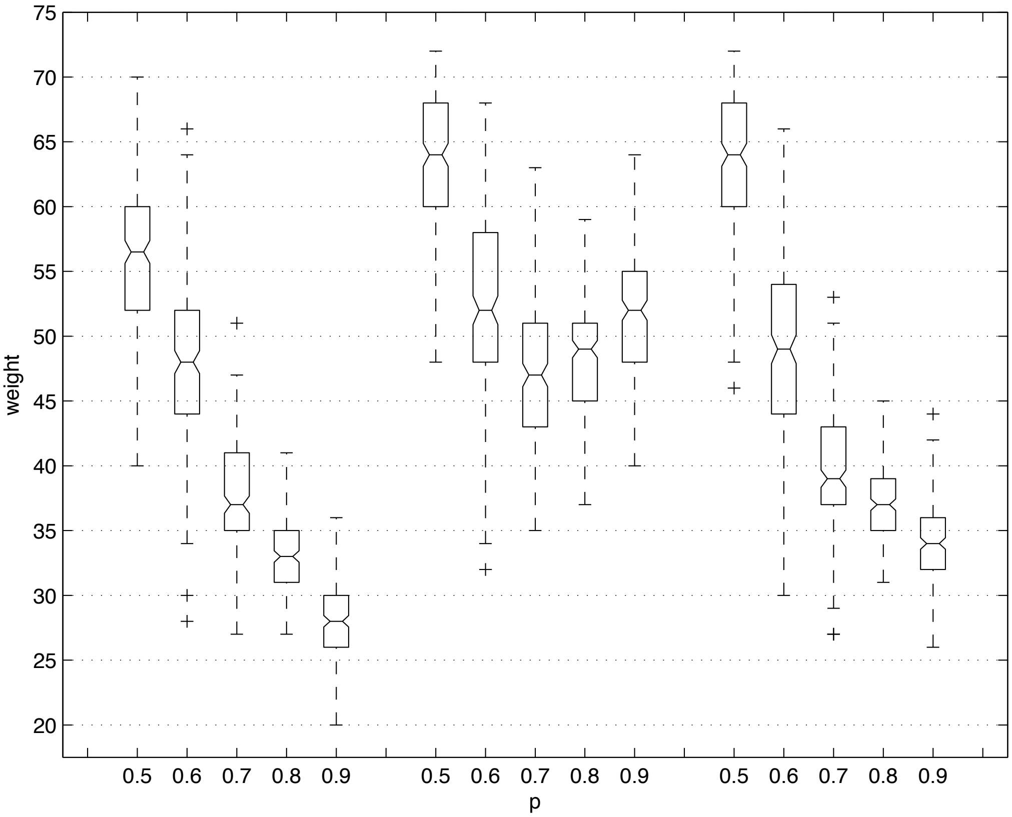


Figure 4: Boxplots summarising the distribution of performance on $G(9, p)$, i.e., a 9×9 grid, for different values of p, over 200 problem instances in each case. The left block of five boxplots show the distribution of the OPTIMAL-MATCHING weight, the middle block show FARTHEST-INSERTION, and the right block show LOCAL-SEARCH.

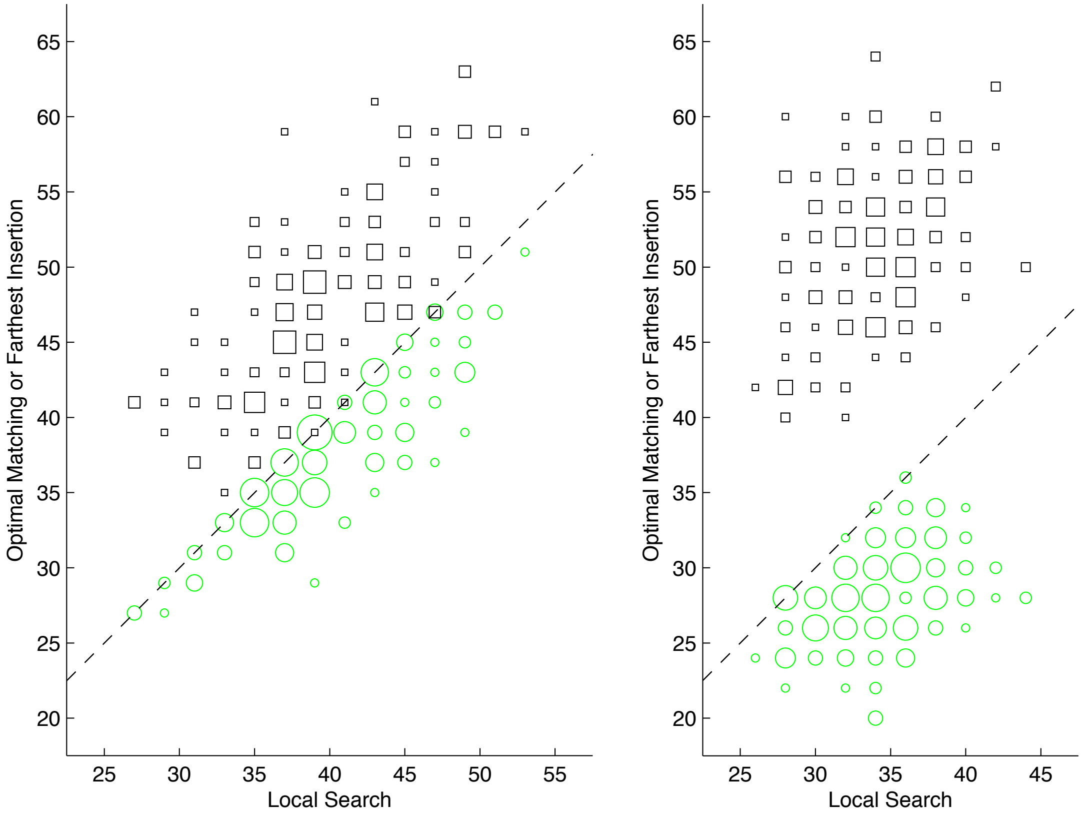
MATCHING when the grid network is not too sparse and not too dense.

Figure 5 looks more closely at the results for $p = 0.7$ and $p = 0.9$. Note that the plot scales are the same. Here $p = 0.7$ is more spread out than $p = 0.9$, i.e, the range of performance for all methods are larger. Also, for $p = 0.9$ the region of circles is very distinct from the region of squares, indicating that LOCAL-SEARCH is much closer in performance to OPTIMAL-MATCHING than FARTHEST-INSERTION, and often performing well. For $p = 0.7$ we see that LOCAL-SEARCH performs consistently well compared to OPTIMAL-MATCHING, while still improving upon FARTHEST-INSERTION (which occasionally does unexpectedly well).

A few observations on verification and insights from visualisation of solutions is warranted. Visualisation (and animation) has proven valuable in checking for traversal more than twice, checking the end cases of edge exchanges, and in verifying that no edge is traversed more than twice.

4 Conclusions and Recommendations for Future Research

Computational experience shows that local search based heuristics for RPP behave well enough on CPP problem instances that are not too sparse or too dense. Therefore, it is justifiable that they are applied to other arc routing problems. Visualisation of grid networks has provided useful insights into the behaviour of local-search



(a) $G(9, 0.7)$

(b) $G(9, 0.9)$

Figure 5: Scatter plots (area of symbol represents the frequency) of performance of OPTIMAL-MATCHING vs LOCAL-SEARCH (circles) and FARTHEST-INSERTION vs LOCAL-SEARCH (squares) on 200 problem instances of (a) $G(9, 0.7)$ and (b) $G(9, 0.9)$.

based heuristics.

Future work will focus on the design of local search heuristics for the RPP+R, beginning by considering only unit rewards. We will design SWAP operators together with local re-optimisation of the tour, similar to GENERALISED-INSERTION. The design of new improvement logic will be fundamental, including how to take advantage of reward clusters. New benchmark problem sets will be required to test and compare the new methods. Additional research questions centre on how to usefully include infeasible tour segments during local search. A major challenge is the visualisation of the tradeoff between reward and distance for different densities of edges and rewards and how these visualisations assist in refining the improvement logic. Integrating rewards into CARP and incorporating stochastic elements into the RRP are two challenging directions for further study. We consider that the RPP+R is a computationally challenging combinatorial optimisation problem worthy of further research.

References

Aarts, E., and J.K. Lenstra, eds. 1997. *Local Search in Combinatorial Optimization.* Wiley.

Ahuja, R.K., Ö. Ergun, J.B. Orlin, and A.P. Punnen. 2002. "A survey of very large-scale neighborhood search techniques." *Discrete Appl. Math.* 123:75–102.

Aráoz, J., E. Fernández, and C. Zoltan. 2006. "Privatized rural postman problems." *Computers and Operations Research* 33:3432–3449.

Dror, M., ed. 2000. *Arc Routing: Theory, Solutions and Applications.* Springer.

Edmonds, J., and E. Johnson. 1973. "Matching, Euler Tours and the Chinese Postman." *Mathematical Programming* 5:88–124.

Eglese, R.W., and L.Y.O. Li. 1992. "Efficient routeing for winter gritting." *Journal of the Operational Research Society* 43:1031–1034.

Eiselt, H.A., M. Gendreau, and G. Laporte. 1995a. "Arc routing problems, part 1: the chinese postman problem." *Operations Research* 43:231–242.

———. 1995b. "Arc routing problems, part 2: the rural postman problem." *Operations Research* 43:399–414.

Evans, J.R., and E. Minieka. 1992. *Optimization Algorithms for Networks and Graphs.* New York: Marcel Dekker.

Feillet, D., P. Dejax, and M. Gendreau. 2005. "Traveling salesman problems with profits: an overview." *Transportation Science* 39:188–205.

Frederickson, N. 1979. "Approximation algorithms for some routing problems." *J. Assoc. Comput. Mach.* 26:538–554.

Frizzell, P.W., and J.W. Giffin. 1995. "The split delivery vehicle scheduling problem with time windows and grid network distances." *Computers and Operations Research* 22 (6): 655–667.

Gendreau, M., A. Hertz, and G. Laporte. 1992. "New insertion and postoptimization procedures for the traveling salesman problem." *Operations Research* 40:1086–1094.

Golden, B.L., L. Levy, and R. Vohra. 1987. "The orienteering problem." *Naval Research Logistics* 34:307–318.

Hertz, A., G. Laporte, and M. Mittaz. 2000. "A tabu search heuristic for the capacitated arc routing problem." *Operations Research* 48:129–135.

Lenstra, J.K., and A.H.G. Rinooy Kan. 1976. "On general routing problems." *Networks* 6:273–280.

Lin, W., and B.W. Kernighan. 1973. "An effective heuristic algorithm for the traveling salesman problem." *Operations Research* 21:498–516.

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.

Toth, P., and D. Vigo, eds. 2002. *The Vehicle Routing Problem.* Philadelphia, PA: SIAM.

Tsiligirides, T. 1984. "Heuristic methods applied to orienteering." *Journal of the Operations Research Society* 35 (9): 797–809.

Existence of Nash-Cournot equilibria over transmission networks

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

Abstract

In this work, we are concerned with characterizing a set of conditions to ensure the existence of an uncongested Cournot equilibrium over an electricity network. Since the deregulation of electricity markets began, many models have been created which consider the electricity market as a game. However, in Cournot models with transmission it is possible that a pure-strategy Cournot equilibrium does not exist. To address this issue, models have been created which limit the rationality of the players or change the sequencing of the game (Yao, Oren, and Adler 2006). These models are able to guarantee the existence of a pure-strategy equilibrium, but, due to their assumptions, could be considered to be less realistic representations of the underlying game.

The properties of the electricity market that can affect the existence of pure-strategy equilibria are the capacities on the transmission lines. Limited line capacities can lead to some parts of the network becoming isolated (all lines into the sub-network are at capacity); without the threat of competition, generators located at those nodes can act in a less competitive fashion by withholding electricity, this will lead to higher prices (it can be shown that the social welfare will be worse in this situation).

In this, we build on the work of Borenstein *et al.* (Borenstein, Bushnell, and Stoft 2000), who considered the effect of the line capacity in a two-node network, to deal with more network configurations. The Cournot model employed consists of strategic generators and linear fringes over a pool market. We formulate this model as a game, wherein the strategic generators are Cournot players (their decision is the quantity of electricity that they wish to generate), and they have full-rationality. To find the Nash equilibrium, we formulate this problem as a bi-level game, whereby the market dispatch problem is embedded within each generator's profit maximization problem. The Nash-Cournot equilibium, can be described as a solution to an equilibrium problem with equilibrium constraints (EPEC) (Ralph and Smeers 2006). However, the EPEC may have many solutions that are not equilibria for the Cournot game, this problem is therefore difficult to solve.

For situations in which the transmission network is radial, we derive conditions on the line capacities which are necessary and sufficient to ensure that the unconstrained Cournot equilibrium exists. These conditions form a convex polyhedral set, which

means they can be easily be embedded as constraints into an optimization problem, ensuring competitive play, while optimizing the use of resources.

We extend this work to examine the impact of two other network properties on this set. The first of these is the effect that losses have on the existence of the uncongested equilibrium; we show that under the assumption of quadratic losses, the presence of a large loss coefficient does not necessarily preclude the existence of an uncongested equilibrium. The second line property is the reactance; this governs the flow of electricity around a loop. Here, we show that when the transmission network contains a loop, we are no longer guaranteed that the conditions ensuring the existence of the uncongested Cournot equilibrium are convex.

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 (2): 294–325.

Ralph, Daniel, and Yves Smeers. 2006, October. "EPECs as models for electricity markets." *Power Systems Conference and Exposition.*

Yao, J., S.S. Oren, and I. Adler. 2006, June. "Modeling and Computing Two-settlement Oligopolistic Equilibrium in a Congested Electricity Network." Energy policy and economics, University of California Energy Institute.

Dispatching repairmen

Colm Hartigan
Department of Engineering Science
University of Auckland

Abstract

Fault-response companies receive calls from customers with service problems, and must respond to these faults by sending a repairman to the fault within a certain time frame. The problem facing a dispatcher in a fault response company is to determine which fault each repairman should repair next. To make the best decision, the dispatcher must consider a plan for the day or week ahead, determining good routes for each repairman.

The problem is modelled as a multiple travelling salesman problem with time windows. The problem is solved initially using estimated repair times. The problem is then resolved when new information arrives. Either a new fault must be repaired, or a fault repair time is found to be different from the estimate.

It was found that the model performed well where the locations of all the faults were known at the start of the day, with the repair times estimated. Problems where new faults arrive during the day can lead to the repairmen doubling back to repair faults in areas they have just visited.

Airline Revenue Management and Game over Itinerary

Abstract

In this article we model a monopolistic market where two airlines have differentiated products and they compete on market shares at the same time as they optimize revenue by taking an appropriate inventory control policy. We show that over a set of possible itineraries there exists only one possible combination of itineraries that both airlines are optimized given the other airline pricing and network strategy and it is Nash equilibrium over network of flights.

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

Solving the Biobjective Integer Minimum Cost Flow Problem

Andrea Raith and Matthias Ehrgott
Department of Engineering Science
University of Auckland
New Zealand
a.raith@auckland.ac.nz and m.ehrgott@auckland.ac.nz

Abstract

Minimum cost network flow problems have practical applications in various areas such as distribution systems planning, medical diagnosis, transportation, or capacity planning, to mention just a few. As with most real-world problems, there is more than one objective to be considered and the objectives are usually conflicting. We solve the biobjective integer minimum cost flow (BIMCF) problem, which is NP-hard and intractable. There currently is no algorithm in the literature to compute the (complete) set of efficient solutions for BIMCF problems. We propose to solve BIMCF with a two phase algorithm. In phase 1 supported extreme solutions are computed with a parametric network simplex algorithm. In phase 2, we transform the BIMCF problem into several single-objective weighted sum problems and solve them with a k best flow algorithm to obtain the remaining solutions. The proposed algorithm is tested on random networks.

1 Introduction

Single-objective integer minimum cost flow problems have received a lot of attention in the literature as they have various applications (e.g. Ahuja, Magnati, and Orlin (1993)). In most real-world optimisation problems, there is usually more than one objective that has to be taken into account, thus leading to multiobjective integer minimum cost flow problems (MIMCF). We restrict our considerations to the biobjective case (BIMCF). The problem of finding all efficient solutions of BIMCF is intractable, Ruhe (1988) presents an example problem with exponentially many solutions. BIMCF is an $\mathcal{NP}$-hard problem, as the biobjective shortest path problem, a special case of BIMCF, was shown to be $\mathcal{NP}$-hard by Serafini (1986).

We propose to solve the BIMCF problem using a two phase approach. In the first phase, extreme supported efficient solutions (those that define extreme points of the convex hull of feasible solution vectors in objective space) are computed with a simplex-based algorithm. Other efficient solutions are computed in the second phase using a ranking algorithm on restricted areas of the objective space. We test our algorithm on different problem instances generated with the well known network generator NETGEN.

2 Biobjective Integer Minimum Cost Flow Problem

Let $G = (N, A)$ be a *directed network* with a set of nodes $N = \{1, \ldots, n\}$ and a set of arcs $A \subseteq N \times N$ with $a = (i, j) \in A$ and $|A| = m$. Two non-negative costs $c_a = (c_a^1, c_a^2) \in \mathbb{N} \times \mathbb{N}$ are associated with each arc $a \in A$. Furthermore, there is a non-negative integer lower bound capacity l_a and an upper bound capacity u_a with $l_a \leqq u_a$ on every arc a. An integer numerical value b_i, the balance, is associated with each node. The value $b_i > 0$, $b_i < 0$, or $b_i = 0$ indicates that, at node i, there exists a *supply* of flow, a *demand* of flow, or neither of the two (i is then called *transshipment* node), respectively. The BIMCF problem is defined by the following mathematical programme:

$$
\begin{aligned}
\min \quad & z(x) = \begin{cases} z_1(x) = \sum_{a \in A} c_a^1 x_a \\ z_2(x) = \sum_{a \in A} c_a^2 x_a \end{cases} && (1)\\
\text{s.t.} \quad & \sum_{\{a : a=(i,j) \in A\}} x_a - \sum_{\{a : a=(j,i) \in A\}} x_a = b_i \quad \forall i \in N && (2)\\
& l_a \leqq x_a \leqq u_a \quad \text{for all } a \in A && (3)\\
& x_a \text{ integer} \quad \text{for all } a \in A. && (4)
\end{aligned}
$$

Here x is the vector of flow on the arcs, constraint (2) represents flow conservation at the different nodes, and we assume that $\sum_{i \in N} b_i = 0$. The *feasible set* X is described by constraints (2) – (4). Its image under the objective function is $Z := z(X)$. Without loss of generality we assume $l_a = 0$ in the following.

In the remainder of this paper we use the following orders on $\mathbb{R}^2$:

$$
\begin{aligned}
y^1 \leqq y^2 \quad &\Leftrightarrow \quad y_k^1 \leqq y_k^2 \; k = 1,2 \quad \text{and}\\
y^1 \leq y^2 \quad &\Leftrightarrow \quad y_k^1 \leqq y_k^2 \; k = 1,2; \; y^1 \neq y^2.
\end{aligned}
$$

Definition 1 *A feasible solution $\hat{x} \in X$ is called* efficient *if there does not exist any $x' \in X$ with $(z_1(x'), z_2(x')) \leq (z_1(\hat{x}), z_2(\hat{x}))$. The image $z(\hat{x}) = (z_1(\hat{x}), z_2(\hat{x}))$ of $\hat{x}$ is called* non-dominated. *Let X_E denote the* set of all efficient solutions *and and let Z_N denote the* set of all non-dominated points. *We distinguish two different types of efficient solutions.*

Supported *efficient solutions are those efficient solutions that can be obtained as optimal solutions to a (single objective) weighted sum problem*

$$\min_{x \in X} c_\lambda(x) = \lambda^1 z_1(x) + \lambda^2 z_2(x) \tag{5}$$

for some $\lambda^1 > 0, \lambda^2 > 0$. The set of all supported efficient solutions is denoted by X_{SE}, its non-dominated image Z_{SN}. The supported non-dominated points lie on the boundary of the convex hull conv(Z) of the feasible set in objective space.

Supported efficient solutions which define an extreme point of conv(Z) are called extreme *supported efficient solutions.*

The remaining efficient solutions in $X_{NE} := X_E \backslash X_{SE}$ are called non-supported *efficient solutions. They cannot be obtained as solutions of a weighted sum problem as their image lies in the interior of conv(Z). The set of non-supported non-dominated points is denoted by Z_{NN}.*

The two objective functions z_1 and z_2 do generally not attain their individual optima for the same values of $\hat{x}$. We assume in the following that there exists no $\hat{x}$ such that $\hat{x} \in \text{argmin}\{z_1\}$ and $\hat{x} \in \text{argmin}\{z_2\}$ for a problem of the form (1) - (4).

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 $x \in X \backslash X_E$ are either dominated or equivalent to at least one $x \in X_E$.*

The presented solution approach computes a complete set X_E. Another notion of optimality that is used in the context of biobjective optimisation is *lexicographic minimisation.*

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

3 Literature

An excellent and very recent review on multiobjective minimum cost flow problems is given by Hamacher, Pedersen, and Ruzika (2007). We will therefore only briefly mention relevant literature. To our knowledge, there is no published work on the MIMCF, so the following is restricted to BIMCF. All exact solution approaches to find a (complete) set of efficient solutions for BIMCF, i.e. supported and non-supported solutions, consist of two phases, also known as the two phase method. In the first phase a complete set of supported efficient solutions, or at least the extreme ones, is computed. In the second phase all remaining solutions are computed.

In case all capacities, supplies, and demands are integer, which we assume in this paper, any approach to solve the biobjective continuous minimum cost flow problem can be used in phase 1 of the BIMCF to find a complete set of extreme supported solutions, e.g. Lee and Pulat (1991), Pulat, Huarng, and Lee (1992), Sedeño-Noda and González-Martín (2000), Sedeño-Noda and González-Martín (2003). For the continuous problem it is sufficient to generate all extreme supported solutions. The algorithms presented by Lee and Pulat (1991), Pulat, Huarng, and Lee (1992) may generate some non-extreme supported solutions, whereas the algorithms by Sedeño-Noda and González-Martín (2000) Sedeño-Noda and González-Martín (2003) generate extreme supported solutions only.

Lee and Pulat (1991) remark that their procedure can be extended to generate all integer efficient solutions with image on the edges of *conv*(Z), i.e. all supported solutions. Every efficient solution found by their algorithm corresponds to a basic tree and two solutions are called *adjacent* if the two corresponding trees differ in only two arcs. Whenever the flow changes by δ when moving from one efficient solution to an adjacent one, they propose to increase the flow stepwise by $1, 2, \ldots, \delta - 1$ to obtain all *intermediate* solutions and claim to obtain all supported solutions this way. This is incorrect, as not all non-extreme supported solutions can be obtained as intermediate solutions of two adjacent basic efficient solutions, an example is given by Eusébio and Figueira (2006).

Several papers are dedicated to the computation of non-supported efficient solutions, assuming all non-dominated extreme points are known. Lee and Pulat (1993) perform an explicit search of the solution space, by using intermediate solutions between adjacent basic solutions (which is not sufficient, see remark above) and modifying upper and lower bounds of arcs. They assume non-degeneracy of the problem. Huarng, Pulat, and A. (1992) extend this algorithm to allow degeneracy in the problems.

Sedeño-Noda and González-Martín (2001) argue that these two papers are incorrect and present an approach that is based on the basic tree structure of solutions. Having found a complete set of extreme supported solutions in phase 1, the algorithm by Sedeño-Noda and González-Martín (2001) moves from one efficient solution to adjacent solutions, in order to identify efficient ones among them. Przybylski, Gandibleux, and Ehrgott (2006) give an example of a network where one efficient solution is not adjacent to any of the other efficient solutions, hence showing that Sedeño-Noda and González-Martín (2001) can not generate a complete efficient set.

Figueira (2002) present an approach where ε-constraint problems are repeatedly solved via branch-and-bound to obtain non-supported solutions.

In her master's thesis, do Castelo Batista Gouveia (2002) uses a k best flow algorithm to enumerate all solutions of biobjective network flow problems, including of course all efficient solutions. She analyses the number of feasible flows, of efficient solutions, and of non-dominated points in the problem.

In the following, we summarise two recent technical reports that were not included in Hamacher, Pedersen, and Ruzika (2007). Eusébio and Figueira (2006) give examples of networks, where for a supported extreme and supported non-extreme non-dominated point, basic and non-basic efficient solutions exist. It is known that there is always a basic solution for every extreme non-dominated point, but the authors show that there may be other non-basic efficient solutions that lead to the same point, so that it may be impossible to obtain *all* efficient solutions when using a simplex-based method. The authors also give a network in which supported solutions exist that can not be obtained as intermediate solutions between two extreme supported solutions.

Eusébio and Figueira (2007) illustrate and prove that supported solutions are indeed connected via chains of zero-cost cycles in the incremental graph constructed from basic feasible solutions corresponding to extreme supported solutions. They use this relationship to obtain all supported solutions to a BIMCF problem.
The same result can be obtained by considering a weighted sum objective (5) for which two neighbouring extreme supported solutions are optimal. The suppported points on the edge of $conv(Z)$ connecting the two extreme non-dominated points can be obtained by applying the k best flow algorithm by Hamacher (1995), which is also based on cycles in the incremental graph. We explain how to apply the k best flow algorithm in Section 4.2.

4 A Two Phase Algorithm to Solve BIMCF

We solve the BIMCF problem with the two phase method. A formulation of the two phase method for general multiobjective combinatorial optimisation problems can be found in Ulungu and Teghem (1995).

The two phase method is based on computing supported and non-supported non-dominated points separately. In phase 1 extreme supported efficient solutions are computed, possibly taking advantage of their property of being obtainable as

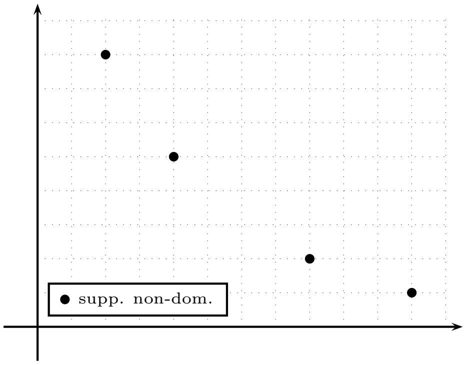


Figure 1: Supported non-dominated points.

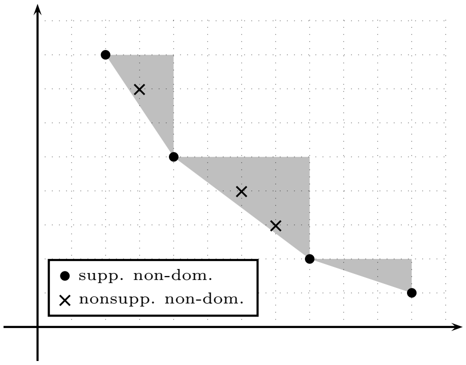


Figure 2: All non-dominated points.

solutions to the weighted sum problem (5), for an illustration see Figure 1. In phase 2 the remaining supported and non-supported efficient solutions are computed with an enumerative approach, as there is no theoretical characterisation for their efficient calculation. The search space in phase 2 can be restricted to triangles given by two consecutive supported non-dominated points as indicated in Figure 2. It is expected that the search space in phase 2 is highly restricted due to information obtained in phase 1 so that the associated problems can be solved quickly.

4.1 Phase 1 – Parametric Simplex

In phase 1 of the two phase method, we compute a complete set of extreme supported solutions of the problem. As mentioned above, any solution method to solve the biobjective continuous minimum cost flow problem can be used here. We use a parametric simplex method by Sedeño-Noda and González-Martín (2000). Initially, one of the two lexicographically optimal solutions, e.g. the $lex(1,2)$-best solution, is obtained with a single-objective network simplex algorithm with accordingly modified objective. From the initial solution, a network simplex algorithm is employed, always choosing a basis entering arcs with the least ratio of improvement of z_2 and worsening of z_1. If there is more than one arc with minimal ratio, one of them is chosen as entering arc, and the others are saved in a list of candidates. After the arc entered the basis, the reduced costs of the remaining arcs in the candidate list are reevaluated. As long as there are remaining candidate arcs in the list violating optimality for the second objective, one of them is introduced into the basis, then the reduced costs of the remaining arcs are reevaluated until there is no more candidate arc in the list or all remaining candidate arcs are optimal with respect to the second objective. Now, an extreme supported solution is obtained and a new list of candidate arcs with minimal ratio is computed. The procedure generates extreme supported solutions moving in a left-to-right fashion. The parametric simplex algorithm finishes when no candidate arcs to enter the basis can be found, i.e. the $lex(2,1)$-best solution is obtained.

4.2 Phase 2 – Ranking k Best Flows

In phase 2, we compute the remaining solutions (supported non-extreme and non-supported solutions). The objective vectors of those solutions can only be situated in the triangle defined by two consecutive extreme supported non-dominated points as indicated in Figure 2. Let $z^1, \ldots, z^s$, where $z^i = (z_1(x^i), z_2(x^i))$ and z^i are sorted by increasing z_1, be the extreme supported points obtained in phase 1. For each pair of neighbouring extreme supported points z^i and z^{i+1}, we define weighting factors

$$\lambda^1 = z_1(x^{i+1}) - z_1(x^i) \quad \text{and} \quad \lambda^2 = z_2(x^i) - z_2(x^{i+1}). \tag{6}$$

Using λ_1 and λ_2 in (5), we obtain a single-objective flow problem which has optimal solutions z^i, z^{i+1} (of course all supported solutions between z^i and z^{i+1} are optimal as well). Applying a k best flow algorithm by Hamacher (1995) to problem (5), we can generate feasible network flows in order of their cost. The k best flow algorithm applied to the single-objective problem with objective $c_\lambda(x) = \lambda^1 z_1(x) + \lambda^2 z_2(x)$ will generate the following solutions:

$$x^1, x^2, x^3, \ldots, x^k \quad \text{with} \quad c_\lambda(x^1) \leqq c_\lambda(x^2) \leqq c_\lambda(x^3) \leqq \ldots \leqq c_\lambda(x^k).$$

When applying the k best flow algorithm in phase 2, flows are computed in order of their cost and those flows that are within the current triangle and not dominated by any other flow in the triangle are saved. Instead of specifying k a priori, the process continues until it can be guaranteed that all non-dominated points (and their efficient flows) have been found.

We call $z^i_N = (z_1(x^{i+1}), z_2(x^i))$ the *local nadir point* of the current triangle. The "worst" solution we are interested in, is the one that is one unit of cost better than z^i_N in each objective. Its weighted objective value is an upper bound to the weighted sum of the two costs of any efficient feasible flow in the current triangle. Thus, initially, we enumerate k best flows x while

$$\lambda^1 z_1(x) + \lambda^2 z_2(x) \leqq u_\lambda \text{ with } u_\lambda = \lambda^1(z_1(x^{i+1}) - 1) + \lambda^2(z_2(x^i) - 1). \tag{7}$$

Whenever an efficient solution with cost vector within the triangle is found, it is saved and the upper bound can be improved, as the new point dominates parts of the triangle. For a detailed description of how the upper bound is updated, please refer to Przybylski, Gandibleux, and Ehrgott (2008) or Raith and Ehrgott (2007).

5 Numerical Results

We investigate the performance of our solution method with networks generated by NETGEN (Klingman, Napier, and Stutz 1974), which is slightly modified to include a second objective function. We vary parameters as in Table 1. We generate 30 instances for each set of parameters. We generate problems N01-N12 with varying sum of supply ($\sum_{i \in N: b_i > 0} b_i$) and problems F01-F12 with fixed total sum of supply, as we observe that increasing the sum of supply with the network size significantly complicates the problem. All NETGEN instances are listed in Table 1.

5.1 Numerical Results

All numerical tests are performed on a Linux (Ubuntu 7.04) computer with 2.80GHz Intel Pentium D processor and 1GB RAM. Run-time is measured with a precision of 0.01 seconds. We make the following observations:

When fixing the number of nodes n in a network but increasing the number of arcs m the number of efficient solutions increases, this is illustrated by instances N01-N12 and F01-F12.

Table 1: Test Instances: NETGEN

Name	n	m	*sources*	*sinks*	$\sum_{i\in N:b_i>0} b_i$	*transhipment sources*	*transsshipment sinks*
N01 / F01	20	60	9	7	90 / 100	4	3
N02 / F02	20	80	9	7	90 / 100	4	3
N03 / F03	20	100	9	7	90 / 100	4	3
N04 / F04	40	120	18	14	180 / 100	9	7
N05 / F05	40	160	18	14	180 / 100	9	7
N06 / F06	40	200	18	14	180 / 100	9	7
N07 / F07	60	180	27	21	270 / 100	14	10
N08 / F08	60	240	27	21	270 / 100	14	10
N09 / F09	60	300	27	21	270 / 100	14	10
N10 / F10	80	240	35	38	350 / 100	17	14
N11 / F11	80	320	35	38	350 / 100	17	14
N12 / F12	80	400	35	38	350 / 100	17	14

Table 2: Results for problems N01 – N12, F01 – F12, and G01 – G12

	$\lvert Z_N\rvert$			$\lvert Z_{SN}\rvert/\lvert Z_{NN}\rvert$	time		
Name	average	min	max	average	average	min	max
N01	168.13	15	392	0.28	0.40	0.01	1.45
N02	271.13	66	852	0.22	0.76	0.09	3.17
N03	375.43	126	702	0.18	1.40	0.27	3.78
N04	455.10	137	879	0.15	7.09	1.67	26.36
N05	660.63	252	1801	0.14	11.84	3.16	36.95
N06	948.30	266	2280	0.12	22.58	5.05	74.91
N07	867.80	410	1399	0.11	42.21	11.48	94.32
N08	1510.37	531	2834	0.09	90.88	27.11	245.20
N09	1553.47	808	2448	0.09	112.62	32.77	238.82
N10	1138.77	552	1901	0.10	125.42	46.44	372.95
N11	2036.20	989	4109	0.08	289.05	69.97	559.34
N12	2480.70	1287	3921	0.07	397.94	138.38	813.76
F01	181.13	24	491	0.27	0.52	0.04	2.81
F02	260.53	15	685	0.24	0.99	0.02	4.58
F03	353.77	158	788	0.20	1.54	0.28	6.41
F04	213.87	65	380	0.20	2.44	0.58	5.58
F05	354.10	144	701	0.15	5.19	1.86	11.77
F06	478.87	176	714	0.13	9.20	2.53	33.65
F07	203.97	48	410	0.16	7.17	0.87	22.40
F08	343.23	165	860	0.14	13.48	5.31	41.27
F09	454.17	230	950	0.12	21.35	8.18	47.9
F10	146.43	72	277	0.18	8.80	2.75	17.27
F11	277.90	131	680	0.15	19.64	8.38	54.04
F12	414.50	234	693	0.12	34.03	12.57	66.47

For all presented instances, we can observe that the more efficient solutions there are in a problem, the longer the run-time of the algorithm. Also, despite the instances being fairly small, they have a lot of solutions.

For problem type F10, the number of efficient solutions is lower, on average, than that of problems F01, F04, and F07 although they all have the same ratio n/m. This happens, because the value of $\sum_{i\in N:b_i>0} b_i$ is fixed, in problem F10 there are only 100 units of flow shipped through the network consisting of 80 nodes.

The sum of supply significantly increases the number of efficient solutions, which can be seen by comparing the results for problems F01-F12 with the corresponding results of problems N01-N12. It is, however, more realistic to increase $\sum_{i\in N:b_i>0} b_i$ while increasing the network size.

$|Z_{SN}|/|Z_{NN}|$, the ratio of supported and non-supported non-dominated points, is decreasing when the total number of solutions is increasing for NETGEN instances, on average. In most NETGEN instances, less than 20% of all solutions are supported. Thus, the majority of solutions is non-supported. In Figures 3 and 4, the non-dominated points of one instance of each of the classes F01, N01 are shown. This

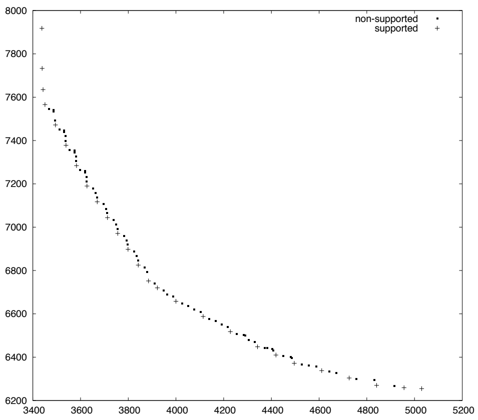


Figure 3: All non-dominated points of one instance of class F01.

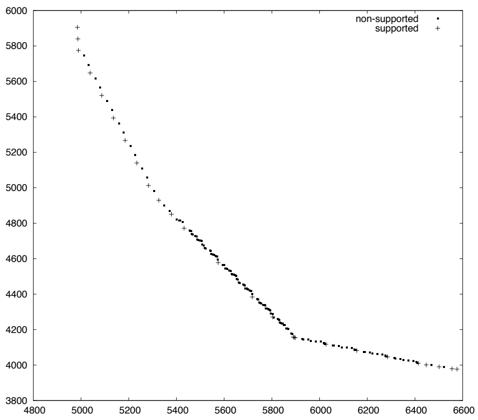


Figure 4: All non-dominated points of one instance of class N01.

illustrates that most non-supported points are in fact very close to the boundary of $conv(Z)$. The given figures are just two examples, but we observe a similar behaviour in most of the problem instances. By obtaining only the supported solutions of a problem, a fairly good approximation of the set of efficient solutions can be obtained.

6 Conclusion

The presented two phase algorithm works well to solve BIMCF problem, but the problems solved within reasonable run-time are fairly small. It is therefore worth investigating how to increase the performance of the presented algorithm to make it possible to solve bigger problems. Future research could address the extension of the the two phase algorithm for BIMCF to the MIMCF problem with more than two objectives. This can be done along the lines of Przybylski, Gandibleux, and Ehrgott (2007), where a two phase method for multiobjective integer programming is presented together with an example of the application to the assignment problem with three objectives.

References

Ahuja, R.K., T.L. Magnati, and J.B. Orlin. 1993. *Network Flows: theory, algorithms, and applications.* Prentice Hall.

do Castelo Batista Gouveia, M. 2002. "The bi-criteria network flow problem: A study of an algorithm for searching efficient solutions (in Portuguese)." Master's thesis, Faculdade de Economia da Universidade de Coimbra.

Eusébio, A., and J.R. Figueira. 2006. "Why is it difficult to find non-dominated solutions in bi-criteria network flow problems when using simplex based methods?" Technical Report, Instituto Superior Técnico, Portugal. submitted.

———. 2007. "A note on the computation of supported non-dominated solutions for bi-criteria network flow problems." Technical Report Working Paper N3/2007, Instituto Superior Técnico, Portugal.

Figueira, J. 2002. "On the integer bi-criteria network flow problem: A branch-

and-bound approach." Technical Report, Université Paris-Dauphine. Cahier du LAMSADE no 191.

Hamacher, H.W. 1995. "A note on K best network flows." *Annals of Operations Research* 57:65–72.

Hamacher, H.W., C.R. Pedersen, and S. Ruzika. 2007. "Multiple Objective Minimum Cost Flow Problems: A Review." *European Journal of Operational Research* 176:1404–1422.

Huarng, F., P. Pulat, and Ravindran A. 1992. "An algorithm for bicriteria integer network flow problem." *Proceedings of the 10th International Conference on Multiple Criteria Decision Making, Taipei, Taiwan*, Volume 3. 305–318.

Klingman, D., A. Napier, and J. Stutz. 1974. "NETGEN: A Program for Generating Large Scale Assignment, Transportation, and Minimum Cost Flow Problems." *Management Science* 20:814–821.

Lee, H., and P.S. Pulat. 1991. "Bicriteria network flow problems." *European Journal of Operational Research.*

———. 1993. "Bicriteria network flow problems: Integer case." *European Journal of Operational Research* 66 (1): 148–157.

Przybylski, A., X. Gandibleux, and M. Ehrgott. 2006. "The biobjective integer minimum cost flow problem–incorrectness of Sedeño-Noda and González-Martín's algorithm." *Computers & Operations Research* 33 (5): 1459–1463.

———. 2007. "A Two Phase Method for Multi-objective Integer Programming and its Application to the Assignment Problem with Three Objectives." Technical Report, LINA, Université de Nantes.

———. 2008. "Two-phase algorithms for the bi-objective assignment problem." *European Journal of Operational Research* 185 (2): 509–533.

Pulat, P.S., F. Huarng, and H Lee. 1992. "Efficient solutions for the bicriteria network flow problem." *European Journal of Operational Research* 19 (7): 649–655.

Raith, A., and M. Ehrgott. 2007. "A comparison of solution strategies for biobjective shortest path problems." Technical Report 648, Department of Engineering Science, The University of Auckland. `http://www.esc.auckland.ac.nz/research/tech/tech.html`.

Ruhe, G. 1988. "Complexity results for multicriteria and parametric network flows using a pathological graph of Zadeh." *Zeitschrift für Operations Research* 32:59–27.

Sedeño-Noda, A., and C. González-Martín. 2000. "The Biobjective Minimum Cost Flow Problem." *European Journal of Operational Research* 124:591–600.

———. 2001. "An algorithm for the biobjective integer minimum cost flow problem." *Computers & Operations Research* 28 (2): 139–156.

———. 2003. "Am alternative method to solve the biobjective minimum cost flow problem." *Asia-Pacific Journal of Operational Research* 20:241–260.

Serafini, P. 1986. “Some Considerations about Computational Complexity for Multi-Objective Combinatorial Problems.” In *Recent advances and historical development of vector optimization*, edited by J. Jahn and W. Krabs, Volume 294 of *Lecture Notes in Economics and Mathematical Systems*, 222–232. Berlin: Springer Verlag.

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.

LGE 2007

John Paynter

Abstract

The triennial local government elections (LGE) are arguably the largest logistics exercise in the country. 73 Territorial Authorites (TA) run elections for their mayor, Councillors in 283 wards, members in 144 community boards and subdivisions, members for 18 Regional Councils in 63 constituencies, 21 District Health Boards and 20 Licensing Trusts (in 42 wards). Additionally there were 10 special issues such as Maori Representation Polls, Fluoride Referenda that are not in the Electoral Enrolment Centre lists. In all there are 466 versions of the base stock sent to electors. In order to cut costs non-resident ratepayers are sent ballot papers based on the same base stock but with the issues in which they are not entitled to vote voided. These subsidiary rolls increase the number of combination to be produced to 1371 (but some may not be needed as no non-Resident Ratepayer is present for that instance - the final figure was 1130). In all an estimated 2.95 million individual voting packs will be sent out at the end of September. With tight deadlines, the need to produce and pack the ballot papers in different locations being able to predict combinations, stock numbers and timing is crucial. In addition there is a need for special votes to be considered. Both ordinary and special votes must reconcile to the rolls in the scrutiny process and positioning and counting information be sent to the different scanning providers.

An Expected Net Present Value Approach to Optimal Harvesting

R. Suri
Institue of Information and Mathematical Sciences
Massey University
New Zealand
r.k.suri@massey.ac.nz

Abstract

In this study, a bio-economic model of a fishery under continuous harvesting is investigated. The population is assumed to consist of a single species of fish, and the growth rate is assumed to be density-dependent only. The optimal harvesting strategy is assumed to maximize the expected net present value of total profit earned by the harvester. The aim is to numerically analyze the sensitivity of the present value of total profit with respect to different combinations of the catchability and cost parameters. The effect of high and low variance components in the price and the growth rate dynamics is studied and the conclusions about the deterministic and the stochastic cases are drawn.

1 Introduction

Harvesting of fish is an ecological as well as an economic issue. Considerable research has gone into optimal harvesting policies when the resource stock follows deterministic growth (Clark 1990). However, varying environmental and biological conditions cause random fluctuations which make the population growth stochastic. In many models, the associated cost structure assumes the price to be either fixed or a prescribed function which can be oversimplified. These issues led to research in the area of stochastic growth and price dynamics.

The optimal rate of extraction, when the resource stock follows stochastic growth and the output price is either fixed or a prescribed function, has been studied by a large body of literature (Reed 1974; Gleit 1978; Pindyck 1984). The effect of random fluctuations in price and population growth has also been investigated (Hanson and Ryan 1998). But, in all the aforementioned papers, there is no emphasis on the conservation of fish population.

In order to overcome this problem and to prevent the fish from becoming extinct, we introduce a population barrier and the fish stock is not allowed to fall below that level throughout harvesting.

2 Model formulation

The deterministic Schaefer model (Schaefer 1957) is extended to its stochastic version. In the stochastic case, the system experiences continuous disturbance in the population growth and the price. The growth dynamics of the resource population is assumed to follow an Itô-stochastic differential equation

$$dx(\tau) = \left[\left\{rx(\tau)\left(1 - \frac{x(\tau)}{K}\right)\right\} - qE(\tau)x(\tau)\right]d\tau + \sigma_1(\tau)dW_1(\tau) \tag{1}$$

where $x(\tau)$ is the biomass of the fish population at time τ, $r > 0$ is the *intrinsic growth rate*, $K > 0$ is the *environmental carrying capacity*, $E(\tau)$ is the *fishing effort*, q is the *catchability coefficient*, $\sigma_1(\tau)$ is the diffusion component representing the random growth effects, and $dW_1(\tau)$ denotes standard Wiener increments. The effort is constrained as $0 \leq E(\tau) \leq E_{\max}$ for all τ, where $E_{\max}$ is a fixed constant.

The price dynamics is also given by an Itô-stochastic differential equation

$$dp(\tau) = \mu(\tau)\,d\tau + \sigma_2(\tau)\,dW_2(\tau) \tag{2}$$

where $p(\tau)$ is the price per unit harvest, $\mu(\tau)$ is the drift in price, $dW_2(\tau)$ denotes standard Wiener increments, and $\sigma_2(\tau)$ is the diffusion component representing the magnitude of random price effects. We assume the cost of harvest to be of quadratic form, given by $c_1E(\tau)+\frac{c_2}{2}E(\tau)^2$, where c_1 and c_2 are positive constants. Considering the optimal harvesting strategy to be the one which maximizes the expected net present value of total profit, the problem can be formulated as an optimal control problem with the control variable being $E(\tau)$ and the value function given by

$$J^*(x(0),0) = \max_{E(t)} \mathcal{E}\left[\int_0^T e^{-\delta t}\left(p(\tau)qx(\tau) - c_1 - \frac{c_2}{2}E(\tau)\right)E(\tau)d\tau\right] \tag{3}$$

The optimal harvesting policy is determined using the stochastic dynamic programming technique which yields Hamilton-Jacobi-Bellman partial differential equation for the discounted flow of profit as follows:

$$\begin{aligned} -\frac{\partial J^*}{\partial t} &= \max_{E(t)}\Big[\left(p(\tau)qx(\tau) - c_1 - \frac{c_2}{2}E(\tau)\right)E(\tau) - \delta J^* \\ &\quad + \mu(\tau)\frac{\partial J^*}{\partial p} + \left\{rx(\tau)\left(1 - \frac{x(\tau)}{K}\right) - qE(\tau)x(\tau)\right\}\frac{\partial J^*}{\partial x} \\ &\quad + \frac{\sigma_1(\tau)^2}{2}\frac{\partial^2 J^*}{\partial x^2} + \frac{\sigma_2(\tau)^2}{2}\frac{\partial^2 J^*}{\partial p^2}\Big] \end{aligned} \tag{4}$$

3 Conclusions and discussion

Due to the non-linearity and complex nature of the Hamilton-Jacobi-Bellman equation, the Crank-Nicolson finite-difference method is employed to obtain numerical solution. The simulations show that the stochastic model behaves like the deterministic model in the mean sense. The sensitivity of the net present value of total profit with respect to the catchability and the cost parameters is analyzed numerically. For low catchability it is noted that the quadratic term in the cost function has a significant effect on the discounted total profit whereas the effect of the linear term is negligible. For high catchability, the discounted total profit is seen to be more or less constant with the optimal effort at maximum level.

The world has witnessed many fisheries collapsing due to over-exploitation. We try to mitigate this problem by maintaining the population above the minimum viable level throughout the harvesting period. The assumption of a cost function which is quadratic in fishing effort allows us to derive an analytical expression for the optimal effort; the resulting solution is different from the bang-bang solution which is usually obtained in the case of a cost function which is linear in effort. With the bang-bang solution there exists an *optimal* population threshold above which it is optimal to harvest at full capacity. If the initial population level is below this threshold then the optimal harvest rate is zero; the population is allowed to grow until it reaches the optimal threshold, and then it is harvested using maximum effort. In other words, the optimal effort switches between zero and its maximum value. Whereas, a cost quadratic in effort allows the optimal effort to assume values other than zero and maximum.

In this work, the drift component μ in the price dynamics is assumed to be zero. Another possibility is to incorporate a drift dependent upon the harvest or the population available for harvest at each time stage. Moreover, the two standard Wiener processes dW_1 and dW_2 governing the random changes in the population growth and the price dynamics are assumed to be uncorrelated. This analysis can be extended to include correlation between dW_1 and dW_2.

References

Clark, C.W. 1990. “Mathematical bioeconomics - the optimal management of renewable resources.” *Wiley series, New York.*

Gleit, A. 1978. “Optimal harvesting in continuous time with stochastic growth.” *Mathematical Biosciences* 41:111–123.

Hanson, F.B., and D. Ryan. 1998. “Optimal harvesting with both population and price dynamics.” *Mathematical Biosciences* 148:129–146.

Pindyck, R. 1984. “Uncertainty in the theory of renewable resource markets. The review of economic studies,.” *Proceedings of the American Mathematical Society* 51:289–303.

Reed, W.J. 1974. “A stochastic model for the economic management of a renewable animal resource.” *Mathematical Biosciences* 22:313–337.

Schaefer, M.B. 1957. “Some considerations of population dynamics and economics in relation to the management of marine fisheries.” *Journal of the Fisheries Research Board of Canada* 14:669–681.

Multiple objective linear programming in beam intensity optimization of IMRT

Lizhen Shao and Matthias Ehrgott
Department of Engineering Science
The University of Auckland, New Zealand
l.shao@auckland.ac.nz; m.ehrgott@auckland.ac.nz

Abstract

We formulate the beam intensity optimization problem of IMRT as a multiobjective linear programme (MOLP). The constraints of the MOLP involve the dose deposition matrix. It is calculated by mathematical models of the physical behaviour of radiation as it travels through the body. The results are always imprecise due to the nonuniform composition of the patient body. Thus, solving the MOLP exactly may give an unwarranted impression of precision, but the result of the optimization can of course not be more precise than the input data. Therefore, it is perfectly acceptable to solve the MOLP approximately to within a small fraction of a Gy (Gray, the unit of measure for radiation dose) for clinical purposes. We propose two objective space methods for approximately solving the MOLP. They are an approximation version of Benson's algorithm and an approximate dual variant of Benson's algorithm, which solve the primal MOLP and dual MOLP, respectively. We proved that they are guaranteed to find -nondominated sets. Application of the two methods to the beam intensity optimization problem for clinical cases shows that the dual approximation method always shows a computational advantage in our experiments.

Area restricted forest harvesting with adjacency branches

Alastair McNaughton
Department of Mathematics
University of Auckland
Private Bag 92019
New Zealand
a.mcnaughton@auckland.ac.nz

David Ryan
Department of Engineering Science
University of Auckland
Private Bag 92019
New Zealand
d.ryan@auckland.ac.nz

Abstract

Forest harvesting applications with restrictions on the permitted area of contiguous clearfell are usually modelled by using numerous adjacency constraints. We replace these adjacency constraints with a technique involving adjacency branches. Each branch is based on an actual clearfell area violation. These violations are found with respect to nuclear sets and associated time intervals. The adjacency branches which eliminate these violations are constructed with respect to the same nuclear sets. The talk will include results from numerical trials indicating recent progress with this model.

Key words: forest harvesting, linear programming, adjacency constraints, area restriction model, optimization.

1 Problem review

When timber is felled in a large commercial forest it is generally required that the size of the openings be restricted. The purpose for this regulation is so as to maintain environmental standards, preserving soil, shelter and drainage. In particular we wish to avoid erosion, sheet-runoff, silting of streams and the associated eye sores.

It is customary to model forest harvesting applications as mixed integer programmes. The objective is the present net worth of the forest, which is to be maximized subject to various strategic and site-specific constraints, as explained by McNaughton et al [9]. The most troublesome of these constraints are the so-called adjacency constraints. These are used to enforce the regulation concerning the maximum area of opening. We use the term *clearfell* to describe a contiguous area of the forest which has been recently harvested. Once such a clearing has been made, none of the surrounding forest may be cut for a certain specified length of time, *the greenup period, T*, which is typically between 5 and 10 years. The term *area restricted model, ARM,* is used by Murray [14] to describe this situation.

This paper is part of a continuing research project and should be read as a sequel to related papers at ORSNZ 2001 [10], 2002 [11], 2003 [12] and 2004 [13].

The aim of this research project is to develop an appropriate model and an effective solution algorithm which solves the forest harvesting problem with area restricted clearfell in an exact sense. The key idea is to replace the customary plethora of complicated adjacency constraints by a system of a much smaller number of adjacency branches.

2 Literature review

The bibliography contains a number of relevant papers. Particular attention is drawn to the several recent ARM formulations by Crowe et al [2], Goycoolea et al [3], Gunn and Richards [4], Murray et al [16] and Vielma et al [17]. These very recent papers present exact solutions to ARM formulations, but the performance attained has limitations particularly with respect to the number of time periods in the planning horizon. The maximum number of time periods considered in these papers is no more than 7. In addition there are limitations with respect to the forest size. Only 2 of these models appear capable of dealing with forests of over 1000 units.

3 Model outline

The key concept is that of a *nuclear set.* This consists 2 parts. First there are 1 or more contiguous units which form the *nucleus*, the sum of whose area is less than or equal to the maximum clearfell area. The second part, the *perimeter*, consists of those units adjacent to the nucleus each of which individually has sufficient area to produce (along with the nucleus) a clearfell violation. A *clearfell violation* then consists of a nuclear set along with a time interval, say $[a, b]$, (with $b - a \leq T$) such that all the units in the nucleus are felled within the interval $[a, b]$ and in addition at least 1 of the perimeter units has also been felled in the appropriate time interval $[b - T + 1, a + T - 1]$.

The model incorporates both column generation and constraint generation, although the latter is essentially a component of the adjacency branching. An initial relaxed LP contains only road plan constraints, non-declining yield constraints and yearly area constraints. The initial variables consist just of elementary columns each representing the harvesting of just 1 unit in a given year. Column generation then develops numerous composite harvesting plans leading to an improved objective value. The resulting solution is then searched for clearfell violations. The way this is done is that each nuclear set is checked sequentially to detect any violations of the sort noted above. If an adjacency violation is detected an adjacency branch is implemented to remove this violation. The detail of these branches is given below. The RLP is then re-optimized and more column generation follows.

Once no clearfell violations remain the solution is scanned to find any fractional decision variables. These are then removed by constraint branching (also called *integer branching*) in the manner developed by Ryan [9]. Once no adjacency violations and no fractional decision variables remain, the associated integer solution is recorded. The branch and bound tree continues to be searched until either an acceptable objective value is obtained, or the root node is found.

4 Modifications to the model

The most significant modifications to the model since the 2004 paper [13] are as follows. The definition of a nuclear set has been modified. The list of nuclear sets has roughly doubled in size, but the previous complicated method for determining the minimum such list has now been removed. The need for any adjacency constraints has now been removed. This has allowed the solution of much bigger applications. There are now a greater number of branching steps, but each branch is much simpler and the solution algorithm moves through these branches much smoother.

5 Adjacency branches

This is the major innovation, and will be described in detail. Suppose a nuclear set has n units in its nucleus. Suppose we find an adjacency violation with a nucleus felled in time interval $[a, b]$ and at least 1 of the perimeter units felled in $[b - T + 1, a + T - 1]$. For the 1-branch we require all the nucleus felled in time interval $[a, b]$ and none of the perimeter units felled in $[b - T + 1, a + T - 1]$. For the 0-branch we require at most $n - 1$ of the nucleus units felled in time interval $[a, b]$ and there are no restrictions of the perimeter units. Adjacency branches are prioitized with respect to the net present worth of the nucleus.

6 Numerical results from trials

The following table records results from a simulated trial involving 400 units.

horizon (time periods)	green-up (time periods)	objective (million $US)	upper bound	time (seconds)	adjacency branches
5	1	6.12	6.13	3	23
	2	6.06	6.13	8	45
	3	6.02	6.08	7	32
	4	5.76	5.95	3	33
	5	5.58	5.70	7	45
10	1	10.85	10.86	6	47
	2	10.78	10.80	13	64
	3	10.72	10.73	13	68
	4	10.60	10.70	17	80
	5	10.49	10.59	19	94
15	1	14.27	14.42	44	88
	2	14.23	14.37	53	101
	3	14.12	14.13	33	102
	4	14.00	14.11	32	124
	5	13.85	13.99	52	153
20	1	16.71	16.72	52	75
	2	16.66	16.67	68	136
	3	16.59	16.67	98	153
	4	16.46	16.50	74	165
	5	16.05	16.25	73	166

Table 1: Trials involving a poorly regulated forest of 400 units.

Successful trials have been completed with up to 1600 units. Various forest types have been considered including over-mature, poorly regulated and well regulated.

7 Conclusions and on-going research

The model appears to represent an advance especially with respect to the number of time periods being used. It is significant that multiple period green up is possible. This results in planning at a much more precise level than is usually obtained in tactical forest planning. It is encouraging that large applications of 1600 units are possible, but it is hoped to extend the capacity further. The model has been simplified, partly to make it easier to explain, but it would be good now to consider incorporation of more constraints. In particular, road planning constraints could be easily incorporated.

References

[1] Crowe, K., J. Nelson and M. Boyland. 2003. *Solving the area-restricted harvest-scheduling model using the branch and bound algorithm*, Can. J. For. Res. 33, pp 1804-1814.

[2] Goycoolea, M., A.T. Murray, F. Barahona, R. Epstein and A. Weintraub. 2005. *Harvest scheduling subject to maximum area restrictions: exploring exact approaches*, Oper. Res. 53(3), pp 490-500.

[3] Gunn, E.A. and E.W. Richards. 2005. *Solving the adjacency problem with stand-centered constraints*, Can. J. For. Res. 35, pp 832-842.

[4] Hoganson, H.M. and J.G. Borges. 2000. *Impacts of the Time Horizon for Adjacency Constraints in harvest scheduling*, Forest Science 46(2), pp 176-187.

[5] McDill, M.E. and J. Braze. 2000. *Comparing adjacency constraint formulations for randomly generated forest planning problems with four age-class distributions*, Forest Science 46(3), pp 423-436.

[6] McDill, M.E., S.A. Rebain and J. Braze. 2002. *Harvest Scheduling with Area-Based Adjacency Constraints*, Forest Science 48, pp 631 - 642.

[7] McNaughton, A.J., M. Rönnqvist and D.M. Ryan. 2000. *A Model which Integrates Strategic and Tactical Aspects of Forest Harvesting.* In System Modelling and Optimization, Methods, Theory and Applications, Edited by M.J.D. Powell and S. Scholtes, Kluwer Academic Publishers Boston, pp 189-208.

[8] McNaughton, A.J., G.D. Page and D.M. Ryan. 2001. *Adjacency Constraints in Forest Harvesting*, proceedings of the ORSNZ, 2001, pp 9-15.

[9] McNaughton, A.J. 2002. *Optimisation of Forest Harvesting Subject to Area Restrictions on Clearfell*, proceedings of the ORSNZ, 2002, pp 307-313.

[10] McNaughton, A.J. 2003. *Adjacency constraints and adjacency branches*, proceedings of the ORSNZ, 2003, pp .

[11] McNaughton, A.J. 2004. *Recent Progress on the Area Restriction Problem of Forest Harvesting*, proceedings of the ORSNZ, 2004, pp .

[12] Murray, A. 1999. *Spatial Restrictions in Forest Scheduling*, Forest Science 45(1), pp 45-52.

[13] Murray, A.T. and A. Weintraub. 2002. *Scale and Unit Specification Influences in Harvest Scheduling with Maximum Area Restrictions*, Forest Science 48, pp 779-789.

[14] Murray, A.T., M. Goycoolea and A. Weintraub. 2004. *Incorporating average and maximum area restrictions in harvest scheduling models*, Can. J. For. Res. 34, pp 456-464.

[15] Vielma, J.P., A.T. Murray, D. Ryan and A. Weintraub. 2003. *Improved Solution Techniques for Multiperiod Area-based Harvest Scheduling Problems*, Systems Analysis in Forest Resources: Proceedings of the 2003 Symposium, pp 285-290.