Lecture Notes in Management and Industrial Engineering

Andreas T. Ernst · Simon Dunstall ·
Rodolfo García-Flores · Marthie Grobler ·
David Marlow *Editors*

Data and Decision Sciences in Action 2

Proceedings of the ASOR/DORS Conference 2018

Springer

Lecture Notes in Management and Industrial Engineering

This book series provides a means for the dissemination of current theoretical and applied research in the areas of Industrial Engineering and Engineering Management. The latest methodological and computational advances that can be widely applied by both researchers and practitioners to solve new and classical problems in industries and organizations contribute to a growing source of publications written for and by our readership.

The aim of this book series is to facilitate the dissemination of current research in the following topics:

- Strategy and Entrepreneurship
- Operations Research, Modelling and Simulation
- Logistics, Production and Information Systems
- Quality Management
- Product Management
- Sustainability and Ecoefficiency
- Industrial Marketing and Consumer Behavior
- Knowledge and Project Management
- Risk Management
- Service Systems
- Healthcare Management
- Human Factors and Ergonomics
- Emergencies and Disaster Management
- Education

More information about this series at http://www.springer.com/series/11786

Andreas T. Ernst · Simon Dunstall ·
Rodolfo García-Flores · Marthie Grobler ·
David Marlow
Editors

Data and Decision Sciences in Action 2

Proceedings of the ASOR/DORS Conference 2018

Springer

Editors
Andreas T. Ernst
School of Mathematics
Monash University
Clayton, VIC, Australia

Rodolfo García-Flores
CSIRO Data61
Docklands, VIC, Australia

David Marlow
Joint and Operations Analysis Division
Defence Science and Technology Group
Fishermans Bend, VIC, Australia

Simon Dunstall
CSIRO Data61
Docklands, VIC, Australia

Marthie Grobler
CSIRO Data61
Docklands, VIC, Australia

ISSN 2198-0772 ISSN 2198-0780 (electronic)
Lecture Notes in Management and Industrial Engineering
ISBN 978-3-030-60134-8 ISBN 978-3-030-60135-5 (eBook)
https://doi.org/10.1007/978-3-030-60135-5

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The Australian Society of Operations Research (ASOR) conference and Defence Operations Research Symposium (DORS) were held in Melbourne, Australia, from Tuesday 4 December to Thursday 6 December 2018. Both of these events are held annually, with DORS run by the Defence Science and Technology (DST) Group within the Australian Department of Defence.

The conference attracted approximately 250 local and international delegates, making it the largest joint ASOR/DORS conference ever. The conference included keynote and general presentations on 4 and 5 December, and a number of workshops on Thursday 6 December. The programme contained presentations across a vast array of theoretical and application areas. Application areas covered food and beverage supply chains, transport and logistics, health care, natural hazards, defence and more. The workshops featured the application of cutting-edge Operations research (OR) to real-world problems, such as real-options testing and last-mile logistics.

The ASOR conference and DORS have been combined on occasions in previous years. Given the relatively small Australian OR community across academia, industry and defence, combining the conferences provides significant benefits through greater networking and collaborative opportunities. On this occasion, the joint conference allowed DST to showcase its Strategic Research Investment into Modelling Complex Warfighting, an internally funded research programme that has strong collaborations with academia and industry.

The conference featured a brilliant line-up of invited local and international speakers: arguably the best assembled at an Australian OR conference other than IFORS 2011 in Melbourne. These speakers included Professor Paolo Toth from the University of Bologna and Professor John Bartholdi from the Georgia Institute of Technology. Their topics included optimization applications in transport and logistics, data analytics, cognitive science and uncertainty modelling.

The proceedings herein contain 23 full articles from the conference. Each article was initially reviewed by two external referees and has been further vetted by the respective session leaders and an editor. We trust that you enjoy these articles that showcase the best of Australia's current Operations Research.

Melbourne, Australia
March 2019

Andreas Ernst
Monash University (Chief Editor)

Simon Dunstall
CSIRO Data61

Rodolfo Garcia-Flores
CSIRO Data61

Marthie Grobler
CSIRO Data61

David Marlow
Defence Science and Technology Group

Introduction

The 2018 Australian Society of Operations Research (ASOR) conference and Defence Operations Research Symposium (DORS) showcased current and forward-thinking in the field of operational research. These proceedings present a selection of these high-quality articles, focusing on a variety of Operations Research (OR) techniques, including data analytics, decision analysis, mathematical programming, optimisation, scheduling, simulation and stochastic models. These techniques are applied to defence, transportation, risk analysis, project management, artificial intelligence, telecommunications and others. Within this collection of articles, we present OR in six main themes, focusing both on the improvement of decision support in commercial and defence environments.

First, the food industry is crucial to ensure Australia's economic prosperity, as the country exports more than half of its production. Increasing the efficiency of food and beverage supply chains is not only of great economic importance but also impacts the social and environmental landscape by reducing food losses and diminishing the amount of resources used for production. These efficiency gains are translated into increased food availability for human consumption, lower pollution and environmental impact, reduction of greenhouse emissions and a more rational use of scarce commodities. Achieving these goals is not of exclusive interest to Australia, and the 2018 ASOR/DORS featured a dedicated **food and beverage** stream in collaboration with the *Wine and Food Supply Chain Council*, a research organisation composed of supply chain professionals committed to improving international supply chains for wine and food, and the University of Newcastle's *Food and Beverage Supply Chain Optimisation Industrial Transformation Training Centre.* The problems tackled include studying the impact of Vendor-Managed Inventory on a decentralized supply chain using a dynamic lot-sizing model (Evazabadian et al.), analysing the location of potential plants to process broccoli that would have otherwise become losses (Garcia-Flores et al.), the analysis of lot-sizing problems with shelf life constraints for perishable products (Chen et al.) and vessel route planning for import/export of seasonal products (Accorsi et al.). These articles show the breadth of interests in optimization for food and beverage supply chains.

Freight and logistics is an essential component of the national economy and the timeliness of product and service delivery. Delayed delivery can result in significant losses to both the sender and the recipient, and therefore advancements in technology, integration and globalization is a clear game changer for supply chain management. The efficiency of Australia's transport and logistics industry can thus be regarded as vital to the nation's productivity and wellbeing, with a direct impact on many other sectors. As such, the 2018 ASOR/DORS featured a dedicated **transport and logistics** stream to showcase some of the open issues and challenges faced in terms of vehicle routing. The problems addressed in this stream include studying a branch-and-price framework for the maximum cover and patrol routing (Chircop et al.), investigating linear complexity algorithms for visually appealing routes in the vehicle routing problem (Kilby et al.), comparing selection by marginal analysis and neural nets to prioritize autonomous supply (Sherman et al.) and planning capacity alignment for a coal chain (Eskandarzadeh et al.). This stream also considers methodologies and platforms in the context of situational awareness and industrial operations modelling (Baumgartner et al.). These articles discuss the alignment and planning abilities needed within a complex environment to uphold the assured quality and quantity within the sector.

The 2018 ASOR/DORS further featured a range of practical Operations Research applications in different scenarios across different sectors. Within this proceedings, these applications are grouped according to non-specific operational scenarios (largely represented in the ASOR streams and workshops) and combat scenarios (largely represented in the DORS streams). In the stream on **case studies and novel applications in non-specific operational scenarios**, we present research focusing particularly on decision-making under uncertainty. In the case of dynamic relocation of aerial firefighting resources to reduce expected wildfire damage, advanced analytics are applied to minimize expected fire damage (Davey et al.). In the context of operating room scheduling based on patient priority, a solution is put forward for master surgical schedule surgical case assignment using integer linear programming and heuristic approaches (Mashkani et al.). Finally, considering non-critical operational scenarios, mixed integer programming is applied to analyse statistical performances in the Australian Football League SuperCoach competition (Edwards). These research articles all present novel applications of scientific methods to advance the management and administration of an array of non-specific operational scenarios.

Defence Operations Research covers a vast array of topics and applications. In the Australian context, this includes providing OR support and advice to all phases of the Capability Life Cycle, such as force structure and design, capability acquisition, capability improvement and sustainability. It also involves the use of techniques such as simulation, optimization, experimentation and wargaming to test the capability of the current and future force. Additionally, Australia's Defence Science and Technology (DST) Group is seeking to both develop and exploit the latest technologies and thinking in operations research through its Strategic Research Investment (SRI) into Modelling Complex Warfighting (MCW). The expected outcome is a revolution in the way that operations research is conducted and applied to future force design and analysis. These proceedings incorporate contributions from all of these areas. In 2018 ASOR/DORS, the stream on **defence decision support analysis** focusses on the application of the appropriate Operations Research techniques to address the extant defence problem. It includes a detailed review of problems considering strategic risk management and its applications in practice (Filinkov et al.), as well as the ability of the security-based global petroleum simulations to accurately model conflict scenarios (Calbert). This stream also puts forward a systems approach to analysing organizational-level adaptability in a critical review of the Australian Army Lessons Network (Omarova et al.). Other articles use the stochastic multi-criteria decision analysis of combat simulation data for selecting the best land combat vehicle option (Cao et al.) and the understanding of the impacts of wheels and tracks on operational outcomes for armoured fighting vehicles in the Australian context (Tomecko et al.). These papers provide a good cross-section of the range of techniques that are required to get insights into this complex area. They including approaches using simulation, optimization and business management, often used in combination.

Force design activities are ongoing within defence and an essential component of the development of the Defence White Paper in Australia every five years. Given that force design is the primary focus of the MCW SRI, the 2018 ASOR/DORS included a stream dedicated to **force design analysis**. From this stream, only one paper was included for publication in the proceedings and added to the section on data analytics in defence. This uses evolutionary algorithms (Hicks et al.) together with a simulation-based evaluation to determine promising force structure options. This has some similarity to the approaches used in the papers by Cao et al. and Tomecko et al. in the much narrower context of selecting appropriate army vehicle options.

The last 2018 ASOR/DORS stream on **novel applications for defence** encompasses all other defence-related submissions. It includes two articles considering machine learning approaches in the context of defence-related problems. The first of these uses a genetic programming framework to discover novel tactics and behaviours in air combat scenarios (Masek et al.), while the second develops a method for generating an “expanded basis set” for a random forests model (Keevers). This section concludes with an article that tests and proposes various approaches for analysing intra-run data to explain causal events and support the analysis of closed-loop combat simulations (Grieger et al.).

Traditionally, OR has focused strongly on deterministic optimization while using discrete event simulation to examine the effects of uncertainty and randomness in complex systems. The articles presented at the 2018 ASOR/DORS conference demonstrate the ongoing trend of combining both optimization and stochastic modelling. While this is mathematically and computationally challenging, these more complex models are becoming more tractable with advances in both computing and mathematical methodology.

Contents

Part I Food and Beverage

Part II Transport and Logistics

Part III Case Studies and Novel Applications in Non-specific Operational Scenarios

Part IV Defence Decision Support Analysis

Part V Other Novel Applications and Data Analytics in Defence

Part I
Food and Beverage

Chapter 1
A Dynamic Lot Sizing Model Under Vendor Managed Inventory (VMI)

Farshid Evazabadian, Regina Berretta, and Mojtaba Heydar

Abstract This paper explores the impact of Vendor managed inventory (VMI) on a decentralized supply chain by proposing two Mixed-integer linear programming (MILP) models for a dynamic lot sizing problem. The models describe the decision-making for lot sizing before and after the implementation of VMI. The proposed models highlight that VMI takes advantage of centralized decision-making, and can reduce the cost of the lot sizing by better synchronization of the decisions. Numerical results, provided to compare the efficiency of VMI with the traditional decentralized lot sizing, indicate significant cost reduction under VMI. A set of experiments is also designed to determine the impact on retailers' cost (ordering and holding) as well as vendors cost (setup and holding) on the gap.

Keywords Vendor managed inventory · Retailer managed inventory · Lot sizing · Mixed-integer linear program

1.1 Introduction

Vendor managed inventory (VMI) is a partnership based on the information shared between the retailer and the vendor (the manufacturer or the supplier) [3]. In VMI, the management of the retailer's inventory is transferred from the retailer to the vendor, after setting shelf space requirements or service level by the retailer [18]. Therefore, the retailer's role changes from managing the inventory to renting the inventory

F. Evazabadian (✉) · R. Berretta
School of Electrical Engineering and Computing, The University of Newcastle, Callaghan, NSW 2308, Australia
e-mail: Farshid.Evazabadian@uon.edu.au

R. Berretta
e-mail: Regina.Berretta@newcastle.edu.au

M. Heydar
Discipline of Mathematics, School of Natural Sciences, University of Tasmania, Hobart, TAS 7001, Australia
e-mail: Mojtaba.Heydar@utas.edu.au

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_1

space [14, 18]. Both retailer(s) and the vendor benefit from VMI implementation. Vendors benefit from VMI by having direct access to the demand information and consequently having more accurate forecasts [12]. Retailers benefit from VMI by increasing their service level and product availability [5, 12], Yao et al. [22].

According to Marquès et al. [13], VMI has three main agreements: partnering agreement, logistical agreement and production and dispatch agreement. In *partnering agreement*, retailers and vendors decide how to collaborate with each other. In fact, they agree on issues such as information that should be shared, the periodicity of transferring information and the timescale of forecasting [13]. In *logistical agreement*, parties discuss issues related to transportation and delivery of products, such as determining the minimum delivery quantity and transport schedule [7, 13]. In *production and dispatch agreement*, the parties make decisions about production planning and shipment scheduling [13]. Production planning contains different stages, including aggregate planning, detailed planning and lot sizing [9]. Lot sizing is considered as an operational level decision that aims to identify the quantity and time of production that minimizes the sum of production, setup and holding costs [2, 9]. The early developments in lot sizing originate from the Economic order quantity (EOQ), proposed by Harris [8], and the Wagner-Whitin algorithm by Wagner and Whitin [6, 20].

In this study, we show how lot sizing can be modelled under VMI. A review related to the lot sizing under VMI is presented in the next section. In Section Mathetmatical Modelling, two mathematical models for the lot sizing problem are introduced, where only one works under VMI. In Section Numerical Experiments, two sets of numerical experiments are presented. In the first, the models are tested with a set of generated instances to show the range of cost reduction under VMI. Next, our study shows the impact of some of the parameters (retailer setup cost, retailer holding cost, vendor setup cost and vendor holding cost) on the VMI efficiency. Finally, the conclusion and directions for future research are presented in Section Conclusion.

1.2 Literature Review

Lot sizing models under VMI can be divided into static and dynamic models. By static models, we mean single-period and continuous timescale models, while dynamic models are multi-period models. In static models, it is assumed that the parameters of the problem, especially demand, are constant and do not change over the planning horizon [11]. However, these parameters can change over time in a dynamic model. Static models can also be divided into deterministic and stochastic models. According to these two factors: uncertainty of parameters (stochastic-deterministic) and variation of parameters over time (static-dynamic), four possible categories of studies can be defined as shown in Table 1.1.

The deterministic-static studies related to lot sizing under VMI first introduced by Yao et al. [22]. They developed two models based on EOQ to compare the total

Table 1.1 Classification of studies for lot sizing under VMI

	Static	Dynamic
Deterministic	Deterministic-Static	Deterministic-Dynamic
Stochastic	Stochastic-Static	Stochastic-Dynamic

cost of a supply chain under VMI and Retailer management inventory (RMI). Van der Vlist et al. [19] extended Yao et al. [21] work by considering shipment cost for the problem. Pasandideh et al. [16] presented an EOQ model with the backlog and discussed how the backlog can increase the profit of the supply chain. Sadeghi et al. [17] presented a model for a multi-vendor, multi-retailer and single warehouse, including a limit for the number of orders and inventory space.

Deterministic-static problems focus on the retailer-vendor competition. Yugang et al. [26] study a supply chain in which the product has two prices: wholesale price and end price, the former is determined by the vendor and the latter set by the retailer. Yu and Huang [25] considered a condition in the supply chain where there are two games between retailers and vendor. The first game is between retailers to gain market share and the second game is between vendor and retailers to achieve profit. They formulated the problem as two Nash games: vertical (between retailers) and horizontal (between vendor and retailers). Yu et al. [24] developed a model to identify which retailers (among several retailers) should be chosen for a VMI partnership to maximize the profit of supply chain.

Stochastic-static studies consider the condition that the values of some of the parameters are not known. In the earliest research in this area, Mishra and Raghunathan [14] considered a supply chain with two vendors and one retailer in which vendors produce two different but substitutable products (each vendor produces one product). Yao et al. [21] study the condition which the vendor, instead of holding a significant amount of the product as inventory, the vendor can pay the retailers to convert the lost sale to backorder when stock-out occurs. In another research, Yao et al. [23] extended their study by considering both backorder and lost sale. Nia et al. [15] developed a Fuzzy programming (FP) model to identify the order size in a supply chain in which multi-products are produced.

In contrast with single-period models, dynamic lot sizing models under VMI have gained less attention. All the dynamic studies fall into the deterministic dynamic category (Table 1.1). In one study, Jaruphongsa et al. [10] developed an MILP model for the incapacitated lot sizing problem, in which for each period, there is a time window, during which the demand must be satisfied. Al-Ameri et al. [1] developed an MILP model for integrated routing and lot sizing problem in a supply chain with multiple manufacturers and retailers. Archetti et al. [4] proposed an MILP model for integrated routing and lot sizing problem in a supply chain, in which there is one vendor, producing one product for several retailers over a time horizon.

As discussed, the majority of the studies related to lot sizing under VMI falls in the static category and there are a few studies in the dynamic area. Static models, due to considering the parameters constant, cannot be applicable for the real cases. Hence, in this study, we present a dynamic lot sizing model under VMI and show

how VMI can be different from a traditional supply chain. Moreover, we show the most affecting factors that differentiate VMI from a traditional supply chain.

1.3 Mathematical Modelling

We consider a supply chain including one manufacturer (vendor) and multiple retailers ($j = 1, 2, \ldots, J$). There are multi-products ($i = 1, 2, \ldots, I$) which each retailer has a specific demand over a time horizon ($t = 1, 2, \ldots, T$). It is assumed that retailers are independent in terms of decision-making, and there is no competition between them for achieving a higher profit. In other words, the demand d_{ijt} for the product i of retailer j in period t is independent of the other retailers' demand. It should be noted that this work considers products with infinite shelf life which will not perish over the horizon. The retailer's costs include an ordering fixed cost sr_{ijt} that does not depend on the quantity of the order, a unit ordering cost o_{ijt}, a holding cost hr_{ijt} for each unit of product and a backorder cost br_{ijt} for each unit of product, if stock-out occurs. It is also considered an ordering lead time lr_{ij}, which is the time, measured in number of periods, between placing of an order and receiving the ordered products. The aim of each retailer is to minimize its total cost, including ordering, holding and backorder cost.

The vendor's costs are setup and production costs for each unit denoted by s_{it} and c_{it}, respectively, a holding cost for each unit (h_{it}), and if stock-out occurs, a backorder cost bs_{it} is incurred for each unit. In regards to capacity for production, it is considered that v_{it} is the necessary time to produce one unit of product i in period t, f_{it} is the setup time required to start production, and production lead time l_i indicates the time from initiations of production until the products become available. Similarly, to the retailers, the vendor aims to minimize its total cost, which includes setup, production and holding costs considering available time (cap_t) as a major limitation. It is assumed that vendor and retailers both have their own warehouse for holding the products.

In order to respond to the demand, two scenarios are considered. In the first scenario which is called RMI, retailers decide how to satisfy the demand. For this purpose, they set some orders and send them to the vendor and based on those orders the vendor plans the production. In the second scenario which is VMI, retailers share the information of demand with the vendor and transfer the decision-making to the vendor. It is assumed that at the beginning of the first period, depends on the scenario, retailers either share the demand information (VMI case) or send their orders over the time horizon to the vendor (RMI case).

In the RMI scenario, each party deals with its own optimization problem. Therefore, in the RMI scenario two mathematical models are required, one for each party (retailers and vendor). In RMI, first retailers make decision and afterwards vendor. However, in the VMI scenario, retailers share the demand information and all the decisions are made by the vendor. Hence, in the second scenario, there is only one mathematical model and all the decisions are made simultaneously in it.

First, we describe the mathematical models for the RMI scenario. In RMI there is one model for retailer j (Model R$_{\mathrm{j}}$) and one for the vendor (Model V). The notation used is as follows:

Index	
i :	index for product $i = 1, ..., I$.
j:	index for the retailer $j = 1, ..., J$
t:	index for time period $t = 1, \ldots, T$.
Parameters:	
sr_{ijt}:	fixed cost of ordering product i for the retailer j in period t
o_{ijt}:	unit ordering cost of product i for the retailer j in period t
hr_{ijt}:	unit holding cost of product i for retailer j at the end of period t
d_{ijt}:	demand of product i for retailer j in period t
lr_{ij}:	ordering lead time of product i for retailer j
br_{ijt}:	backorder cost incurred for each unit of product i when stock-out for retailer j occurs in period t
s_{it}:	setup cost incurred if product i is produced in period t
c_{it}:	unit production cost of product i in period t
h_{it}:	unit holding cost for the product i in vendor warehouse at the period t
v_{it} :	necessary time to produce one unit of product i in period t
f_{it}:	setup time required to produce product i in period t
l_i:	the production lead time of product i
bs_{it}:	the backlogging cost which is incurred for each unit of product when the stock-out for vendor occurs
M:	an upper bound for production
cap_t:	total production time available in period t

Decision variables:

$\mathrm{YR}_{ijt} = \begin{cases} 1 \\ 0 \end{cases}$	If retailer j orders a batch of product i
	Otherwise
XR_{ijt}:	The order quantity of product i for retailer j in period t.
IR^{+}_{ijt}:	Overstock inventory of product i for retailer j in period t
IR^{-}_{ijt}	Under-stock inventory of product i for retailer j that backlogged at the end of period t
$Y_{it} = \begin{cases} 1 \\ 0 \end{cases}$	If the vendor produces product i in period t
	Otherwise
X_{it}:	Production quantity of product i in period t
I^{+}_{it}:	Overstock inventory of product i for the vendor at the end of period t

(continued)

(continued)

I_{it}^{-}	Under-stock inventory of product i backlogged for the vendor at the end of period t

In the following, Models R_j (for the retailer j) and V (for the vendor) are presented.

Model R_j*(for Retailer* $j : 1, \ldots, J$*)*:

$$\min \sum_i \sum_t sr_{ijt} YR_{ijt} + \sum_i \sum_t hr_{ijt} IR_{ijt}^{+} + \sum_i \sum_t br_{ijt} IR_{ijt}^{-} + \sum_i \sum_t o_{ijt} XR_{ijt} \tag{1.1}$$

s.t

$$IR_{ijt}^{+} - IR_{ijt}^{-} = IR_{ijt-1}^{+} - IR_{ijt-1}^{-} + XR_{ij,t-lr_{ij}} - d_{ijt}, i = 1, ..., I; t = 1, ..., T, \tag{1.2}$$

$$XR_{ijt} \le \left(\sum_{t'} d_{ij,t'} \right) YR_{ijt}, \quad i = 1, ..., I; t = 1, ..., T, \tag{1.3}$$

$$IR_{ijt}, XR_{ijt} \ge 0, i = 1, ..., I; t = 1, ...T, \tag{1.4}$$

$$YR_{ijt} \in \{0, 1\}, \quad i = 1, ..., I; t = 1, ...T. \tag{1.5}$$

In model R_j, the objective function (1.1) aims to minimize the total cost of retailer j including ordering, holding and backorder costs. Equation (1.2) is the inventory balance for the retailer j. Constraint (1.3) are logical constraints that relate the decision variable X_{it} to the binary decision variable Y_{it} meaning that whenever an order quantity is greater than zero, variable YRit must take on value 1. Constraints (1.4) and (1.5) show the domains of the variables.

Model V*(for vendor)*:

$$\min \sum_t \sum_i s_{it} Y_{it} + \sum_t \sum_i h_{it} I_{it}^{+} + \sum_t \sum_i bs_{it} I_{it}^{-} + \sum_t \sum_i c_{it} X_{it} \tag{1.6}$$

s.t

$$I_{it}^{+} - I_{it}^{-} = I_{it-1}^{+} - I_{it-1}^{-} + X_{i,t-l_i} - \sum_j XR_{ijt}, \quad i = 1, ..., I; t = 1, ..., T, \tag{1.7}$$

$$X_{it} \le MY_{it}, \; i = 1, ..., I; t = 1, ..., T, \tag{1.8}$$

$$\sum_i (f_{it} Y_{it} + v_{it} X_{it}) \le cap_t, \quad t = 1, ...T, \tag{1.9}$$

$$I_{it}, X_{it} \geq 0, \ \ i = 1, ..., I; t = 1, ...T, \tag{1.10}$$

$$Y_{it} \in \{0, 1\} \ i = 1, ..., I; t = 1, ...T. \tag{1.11}$$

In model V, the objective function (1.6) aims to minimize the sum of holding, backorder, production and setup costs for the vendor. The Eq. (1.7) is the inventory balance for the vendor. Constraint (1.8) guarantee that the solution have set up when it has production. Constraint (1.9) represent the capacity limitation for production. Constraints (1.10) and (1.11) show the domain of variables.

It should be noted that the decision variables XR_{ijt} in Model R are defined as an input parameter for Model V. Formulations (1.12)–(1.21) represents the mathematical model for VMI.

Objective function (1.12) minimizes the summation of retailers and vendor costs. The constraints (1.13)–(1.16) are similar to constraints (1.2)–(1.5) except the former is considered for just one retailer (retailer j), but the latter is considered for all the retailers simultaneously. Finally, constraints (1.17)–(1.21) are similar to the constraints (1.7)–(1.11).

It is observed that the RMI model is hierarchical, while the VMI model is integrated. In other words, VMI model aggregates all the objective functions as well as the constraints and centralizes the decision-making. Due to integration in the VMI model, the performance of the supply chain improves. This issue is discussed in the Lemma below.

Lemma: *VMI offers the optimal solution for the whole supply chain.*

Proof. Since under VMI, the objective function is the minimization of the total cost for the whole supply chain and all the constraints, including retailers and vendor constraints, are considered simultaneously, the obtained solution offers the minimum cost for the whole supply chain.

Since VMI offers the optimal solution for the supply chain, it is interesting to know how much the difference in total cost between RMI and VMI is.

Model VMI

$$\begin{aligned}\min \sum_i \sum_t \sum_j sr_{ijt} YR_{ijt} + \sum_i \sum_t \sum_j hr_{ijt} IR^+_{ijt} + \sum_i \sum_t \sum_j br_{ijt} IR^-_{ijt} + \sum_i \sum_t \sum_j o_{ijt} XR_{ijt} \\ + \sum_t \sum_i s_{it} Y_{it} + \sum_t \sum_i h_{it} I^+_{it} + \sum_t \sum_i bs_{it} I^-_{it} + \sum_t \sum_i c_{it} X_{it}\end{aligned} \tag{1.12}$$

s.t

$$\begin{aligned}IR^+_{ijt} - IR^-_{ijt} = IR^+_{ijt-1} - IR^-_{ijt-1} + XR_{ij,t-lr_{ij}} - d_{ijt}, \\ i = 1, ..., I; j = 1, \ldots, J; t = 1, ..., T,\end{aligned} \tag{1.13}$$

$$XR_{ijt} \le \left(\sum_{t'} d_{ij,t'}\right) YR_{ijt},\ i = 1, ..., I;\ j = 1, \ldots, J; t = 1, ..., T, \tag{1.14}$$

$$IR_{ijt}, XR_{ijt} \ge 0,\ i = 1, ..., I;\ j = 1, \ldots, j; t = 1, ...T, \tag{1.15}$$

$$YR_{ijt} \in \{0, 1\},\ i = 1, ..., I;\ j = 1, \ldots, J; t = 1, ...T, \tag{1.16}$$

$$I^+_{it} - I^-_{it} = I^+_{it-1} - I^-_{it-1} + X_{i,t-l_i} - \sum_j XR_{ijt},\ i = 1, ..., I; t = 1, ..., T, \tag{1.17}$$

$$X_{it} \le MY_{it},\ i = 1, ..., I; t = 1, ..., T, \tag{1.18}$$

$$\sum_i (f_{it}Y_{it} + v_{it}X_{it}) \le cap_t,\ t = 1, ...T, \tag{1.19}$$

$$I_{it}, X_{it} \ge 0,\ i = 1, ..., I; t = 1, ...T, \tag{1.20}$$

$$Y_{it} \in \{0, 1\}\ i = 1, ..., I; t = 1, ...T. \tag{1.21}$$

1.4 Numerical Experiments

In the previous section, we showed that the total cost of lot sizing under VMI is less than or equals to the RMI optimal solution. One important question is how much the maximum possible cost reduction applied by the VMI is. To answer this question, a numerical approach is applied. A set of instances is generated based on the characteristics and parameters depicted in Tables 1.2 and 1.3, respectively. For generating capacity, the formula (1.22) is used. Each instance is solved under RMI and VMI, and the relative gap (the difference between the total cost under RMI and VMI) based on Eq. (1.23) is calculated.

Table 1.2 Instance characteristics

Group 1 of instance	
$I \times J \times T$	$S_1 = 1 \times 2 \times 10$, $S_2 = 1 \times 10 \times 10$, $S_3 = 1 \times 20 \times 10$ $S_4 = 10 \times 2 \times 10$, $S_5 = 10 \times 10 \times 10$, $S_6 = 10 \times 20 \times 10$ $S_7 = 20 \times 2 \times 10$, $S_8 = 20 \times 10 \times 10$, $S_9 = 20 \times 20 \times 10$
Total	90 (10 instances for each $S_i, i = 1, \ldots, 9$)

Table 1.3 Parameters used to generate instances

Parameters	Interval
d_{ijt}	U[0,100]
sr_{ijt}	U[100,200]
o_{ijt}	U[1,5]
hr_{ijt}	U[2,20]
br_{ijt}	U[4,40]
s_{it}	U[250,1000]
c_{it}	U[2,10]
h_{it}	U[1,10]
bs_{it}	U[2,20]
v_{it}	U[1.5,2]
f_{it}	U[150,200]
cap_t	$1.2c_1$

$$c_1 = \left(\frac{\sum_i \sum_j \sum_{t=1}^{T-1} d_{ij,t+1} \times v_{it} + \sum_{t=1}^{T-1} f_{it}}{(T-1)} \right) \tag{1.22}$$

$$gap = \left(\frac{Z^{RMI} - Z^{VMI}}{Z^{RMI}} \times 100 \right). \tag{1.23}$$

Table 1.4 shows the gap between RMI and VMI for all the generated instances. It is observed that the range of cost reduction after VMI implementation is between 0 and 6.4%. Since the lot sizing cost can be very high, to reduce to even only 0.1% can lead to a remarkable saving for the parties.

Table 1.4 % Gap between RMI and VMI total cost for the generated instances

Instances												
	$I \times J \times T$	1	2	3	4	5	6	7	8	9	10	Average
Sizes	1 × 2 × 10	2.3	5.0	4.7	0.3	2.3	0.9	0	1.7	0.6	1.3	1.91
	1 × 10 × 10	1.0	0.5	1.1	1.4	6.4	1.4	1.7	0.6	5.3	1.5	2.09
	1 × 20 × 10	3.1	1.4	1.5	2.3	0.4	0.9	1.7	1.8	1.4	2.2	1.67
	10 × 2 × 10	1.3	3.7	2.0	1.8	1.9	2.1	2.6	2.4	1.9	2.8	2.25
	10 × 10 × 10	1.2	1.2	0.9	1.5	1.3	1.4	1.2	1.2	1.5	1.2	1.26
	10 × 20 × 10	2.0	1.1	0.9	1.2	1.5	1.4	1.4	2.0	1.3	1.0	1.38
	20 × 2 × 10	2.4	1.8	1.5	2.7	2.6	2.6	2.5	2.1	2.6	2.2	2.3
	20 × 10 × 10	1.0	1.2	1.7	1.2	1.1	1.2	1.5	1.0	1.3	1.3	1.25
	20 × 20 × 10	1.6	1.2	1.3	1.4	1.0	1.5	1.6	1.2	1.2	1.2	1.32

1.5 Analysis of Variation of Parameters

This section presents a study about the impact of different parameters on the gap between RMI and VMI. These parameters are: retailer holding cost (hr_{ijt}), vendor holding cost (h_{it}), vendor setup cost (s_{it}) and retailer fixed cost for ordering (sr_{ijt}). As mentioned in the previous section, each parameter is generated in a specific interval. In order to determine the impact of each parameter, we divide the interval into three equal sub-intervals and consider only the *low* and *high* intervals for these experiments. The low level refers to the first one third of parameters' interval, while the high refers to the last one third of the interval. For example for the parameter (hr_{ijt}) the low refers to the range [2, 8].

Since we are considering four parameters with two levels (high and low), we have 16 (=2^4) possible combinations which we consider each of them as a case. In order to analyse the impact of the parameters on the gap (between RMI and VMI), we generated two instances for each case in each size. In other words, for each case, we have 18 (=2 × 9) instances. The average of gap for all of these 18 instances for each case is shown in Table 1.5. It should be noted in Table 1.5, L and H refer to low and high levels, respectively.

In order to assess the impact of one specific parameter (target parameter) on the gap, we need to compare the instances with the same parameter intervals, except the target parameter. For example, in order to analyse the impact of the retailer holding

Table 1.5 Average gap between RMI and VMI for each case

Case	Vendor setup cost	Vendor holding cost	Retailer setup cost	Retailer holding cost	Gap (%)
1	L	L	L	L	2.83
2	L	L	L	H	5.38
3	L	L	H	L	2.30
4	L	L	H	H	5.24
5	L	H	L	L	0.39
6	L	H	L	H	0.47
7	L	H	H	L	0.36
8	H	H	H	H	0.58
9	H	L	L	L	2.60
10	H	L	L	H	5.34
11	H	L	H	L	3.33
12	H	L	H	H	5.12
13	H	H	L	L	0.35
14	H	H	L	H	0.76
15	H	H	H	L	0.38
16	H	H	H	H	0.70

cost on the gap, we compare the instances which the other parameters are produced in the same interval, for example, cases 1 and 2, or cases 3 and 4. The instances which are required to be compared to assess the impact of each parameter are listed below.

(a) Retailer holding cost: compared cases are {(1, 2), (3, 4), (5, 6), (7, 8), (9, 10), (11, 12), (13, 14), (15, 16)},
(b) Retailer setup cost (fixed cost of ordering): compared cases are {(1, 3), (2, 4), (5, 7), (6, 8), (9, 11), (10, 12), (13, 15), (14, 16)}.
(c) Vendor holding cost: compared cases are {(1, 5), (2, 6), (3, 7), (4, 8), (9, 13), (10, 14), (11, 15), (12, 16)}.
(d) Vendor setup cost: compared cases are {(1, 9), (2, 10), (3, 11), (4, 12), (5, 13), (6, 14), (7, 15), (8, 16)}.

Figure 1.1 shows the comparison of the results of paired cases when each parameter is set at its low or high level, respectively. In this figure, the Y axis shows the value of gap (%) for each case.

According to Fig. 1.1, the gap between RMI and VMI increases by increasing retailer holding cost and vendor's holding cost. Moreover, it shows that the vendor's and retailers' setup costs do not have a considerable impact on the gap of optimality. It is also possible to note that lower holding costs lead to lower optimality gap.

1.6 Conclusion

This paper presented two mixed-integer linear mathematical formulations for lot sizing under RMI and VMI paradigms. First, we developed a model (RMI) in which each member optimizes its own production plan individually. Then, in the second model (VMI), retailers share the demand information and transfer the decision-making to the vendor. Through a lemma, we showed that VMI always outperforms RMI and our numerical study showed VMI can reduce the total cost of supply chain by 6.42%, which can be a considerable value in a real-world scale. Our analysis of variation of parameters showed that the gap between VMI and RMI increases by an increase in holding costs of the retailer and the vendor and the setup costs do not have a considerable impact on the gap.

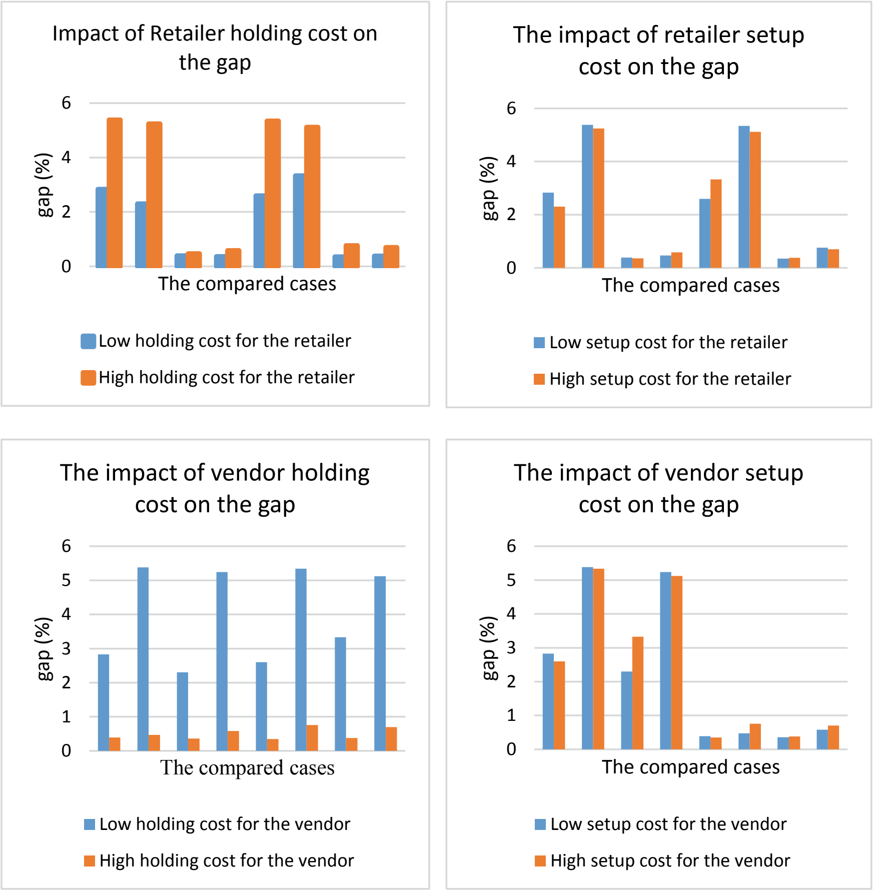


Fig. 1.1 The impact of parameters on the gap between RMI and VMI: **a** Retailer holding cost, **b** retailer setup cost, **c** vendor holding cost, **d** vendor setup cost

References

1. Al-Ameri TA, Shah N, Papageorgiou LG (2008) Optimization of vendor-managed inventory systems in a rolling horizon framework. Comput Ind Eng 54(4):1019–1047
2. Almada-Lobo B, Clark A, Guimarães L, Figueira G, Amorim P (2015) Industrial insights into lot sizing and scheduling modeling. Pesquisa Oper 35(3):439–464
3. Angulo A, Nachtmann H, Waller MA (2004) Supply chain information sharing in a vendor managed inventory partnership. J Bus Logist 25(1):101–120
4. Archetti C, Bertazzi L, Laporte G, Speranza MG (2007) A branch-and-cut algorithm for a vendor-managed inventory-routing problem. Transport Sci 41(3):382–391
5. Çetinkaya S, Lee C-Y (2000) Stock replenishment and shipment scheduling for vendor-managed inventory systems. Manag Sci 46(2):217–232
6. Clark A, Almada-Lobo B, Almeder C (2011) Lot sizing and scheduling: industrial extensions and research opportunities. Taylor & Francis

7. Gröning A, Holma H (2007) Vendor managed inventory: preparation for an implementation of a pilot project and guidance for an upcoming evaluation at Volvo Trucks in Umea
8. Harris FW (1913) How many parts to make at once. Factory Mag Manag 10(2):135–136
9. Hennet J-C (1999) From the aggregate plan to lot-sizing in multi-level production planning. In: Modeling manufacturing systems, pp 5–23. Springer
10. Jaruphongsa W, Çetinkaya S, Lee C-Y (2004) A two-echelon inventory optimization model with demand time window considerations. J Global Optim 30(4):347–366
11. Khator SK, Leung LC (1997) Power distribution planning: a review of models and issues. IEEE Trans Power Syst 12(3):1151–1159
12. Kuk G (2004) Effectiveness of vendor-managed inventory in the electronics industry: determinants and outcomes. Inf Manag 41(5):645–654
13. Marquès G, Thierry C, Lamothe J, Gourc D (2010) A review of vendor managed inventory (VMI): from concept to processes. Prod Plan Contr 21(6):547–561
14. Mishra BK, Raghunathan S (2004) Retailer-vs. vendor-managed inventory and brand competition. Manag Sci 50(4):445–457
15. Nia AR, Far MH, Niaki STA (2014) A fuzzy vendor managed inventory of multi-item economic order quantity model under shortage: An ant colony optimization algorithm. Int J Prod Econ 155:259–271
16. Pasandideh SHR, Niaki STA, Nia AR (2010) An investigation of vendor-managed inventory application in supply chain: the EOQ model with shortage. Int J Adv Manuf Technol 49(1–4):329–339
17. Sadeghi J, Mousavi SM, Niaki STA, Sadeghi S (2013) Optimizing a multi-vendor multi-retailer vendor managed inventory problem: two tuned meta-heuristic algorithms. Knowl-Based Syst 50:159–170
18. Sari K (2007) Exploring the benefits of vendor managed inventory. Int J Phys Distr Logist Manag 37(7):529–545
19. Van der Vlist P, Kuik R, Verheijen B (2007) Note on supply chain integration in vendor-managed inventory. Decis Support Syst 44(1):360–365
20. Wagner HM, Whitin TM (1958) Dynamic version of the economic lot size model. Manage Sci 5(1):89–96
21. Yao Y, Dong Y, Dresner ME (2007) Analyzing information-enabled stockout management under vendor-managed inventory. Inf Technol Manag 8(2):133–145
22. Yao Y, Evers PT, Dresner ME (2007) Supply chain integration in vendor-managed inventory. Decis Support Syst 43(2):663–674
23. Yao Y, Dong Y, Dresner M (2010) Managing supply chain backorders under vendor managed inventory: an incentive approach and empirical analysis. Eur J Oper Res 203(2):350–359
24. Yu Y, Hong Z, Zhang LL, Liang L, Chu C (2013) Optimal selection of retailers for a manufacturing vendor in a vendor managed inventory system. Eur J Oper Res 225(2):273–284
25. Yu Y, Huang GQ (2010) Nash game model for optimizing market strategies, configuration of platform products in a Vendor Managed Inventory (VMI) supply chain for a product family. Eur J Oper Res 206(2):361–373
26. Yugang Y, Liang L, Huang GQ (2006) Leader–follower game in vendor-managed inventory system with limited production capacity considering wholesale and retail prices. Int J Logist: Res Appl 9(4):335–350

Chapter 2
A Two-Stage Stochastic Model for Selection of Processing Hubs to Avoid Broccoli Losses

Rodolfo García-Flores, Elaine LeKhon Luc, Peerasak Sanguansri, and Pablo Juliano

Abstract It is estimated that, at present, around one third of all food produced is lost, either in production and distribution or after retail. To further complicate matters, uncertainty and variability in the commercial and natural environments must also be taken into account when trying to reduce food losses. The objective of the present paper is to develop a decision support system to increase the efficiency of the Australian broccoli supply chain and reduce food losses considering uncertainty. To that end, we develop a two-stage stochastic mixed-integer linear programming model to assist Australian broccoli producers in taking the most cost-effective investment decisions and, at the same time, reduce the losses by producing novel, high value-added products from produce discarded on the field or during transportation. The stochastic model we propose selects, in the first stage, the optimal location of processing facilities to add value to the produce that would otherwise be considered as food loss, and suggests transportation operations as the recourse decisions. The model is solved using Lagrangian decomposition and the subgradient method. The data used to feed the model was collected on the field through a survey applied to broccoli growers nationwide. The model suggests near-optimal investment decisions that are far from the worst possible outcome, had the final market and environmental conditions turned out to be very adverse. Our results represent viable operations for the industry in the medium term.

Keywords Stochastic programming · Supply chains · Hub selection

R. García-Flores (✉) · E. LeKhon Luc
CSIRO Data61, Locked Bag 38004, Docklands 3008, Australia
e-mail: Rodolfo.Garcia-Flores@csiro.au

P. Sanguansri · P. Juliano
CSIRO Agriculture and Food, 671 Sneydes Road, Werribee 3030, Australia

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_2

2.1 Introduction

Food production and distribution are currently undergoing dramatic changes. On one hand, it is expected that food consumption will continue to grow for at least the next 40 years [7], driven by a growing population and rising incomes. On the other hand, this increase in food demand will be constrained by scarcer resources, more attention to food security and changing dietary habits. It is also estimated that, at present, around one third of all food produced is lost, either in production and distribution or after retail. To further complicate matters, uncertainty and variability in the commercial and natural environments must be taken into account as part of the planning of all activities. In order to address all these challenges, the industry must leverage new food processing and digital data capturing technologies, as well as apply advanced quantitative decision-making techniques.

The objective of the present paper is to develop a decision support system for reducing food loss (i.e. before retail, as opposed to food waste, which occurs after retail) in the Australian broccoli supply chain by selecting, among a set of candidate sites, the most adequate to build processing sites intended to transform the produce that would otherwise become landfill or animal feed into processed snacks or food supplements. Broccoli and broccolini are grown in most Australian states, of which Victoria is the largest producer, with 48% share of total production. Figure 2.1 shows the amount and value of broccoli in the supply chain: for the year ending in June 2015, 68571 ton were produced, representing \$188.7 million in value (all figures in Australian dollars), with a wholesale value of the fresh supply of \$ 210.7 M. Exports represent \$10.5 M, 69% of which go to Singapore [9]. Figure 2.2 is a map of the region under study showing the location, demand and supply of consumers and producers of broccoli, respectively. Broccoli producers are marked in yellow and consumer

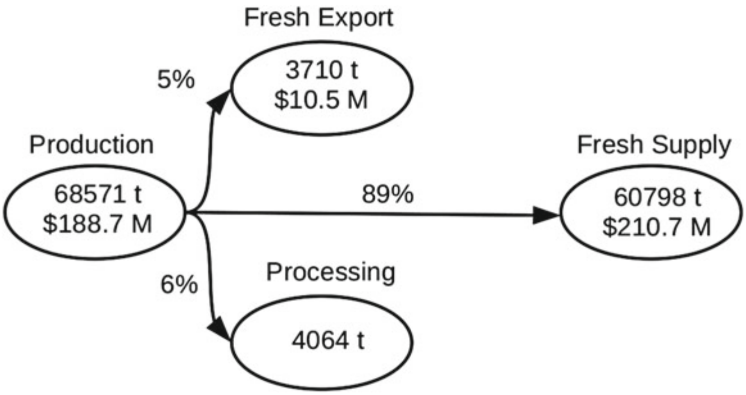


Fig. 2.1 Overview of brassicas production in Australia. For the year ending in June 2015, 68571 ton were produced, representing \$188.7 million dollars in value, with the wholesale value of the fresh supply of \$210.7 M. Exports represent \$10.5 M, 69% of which go to Singapore [9]

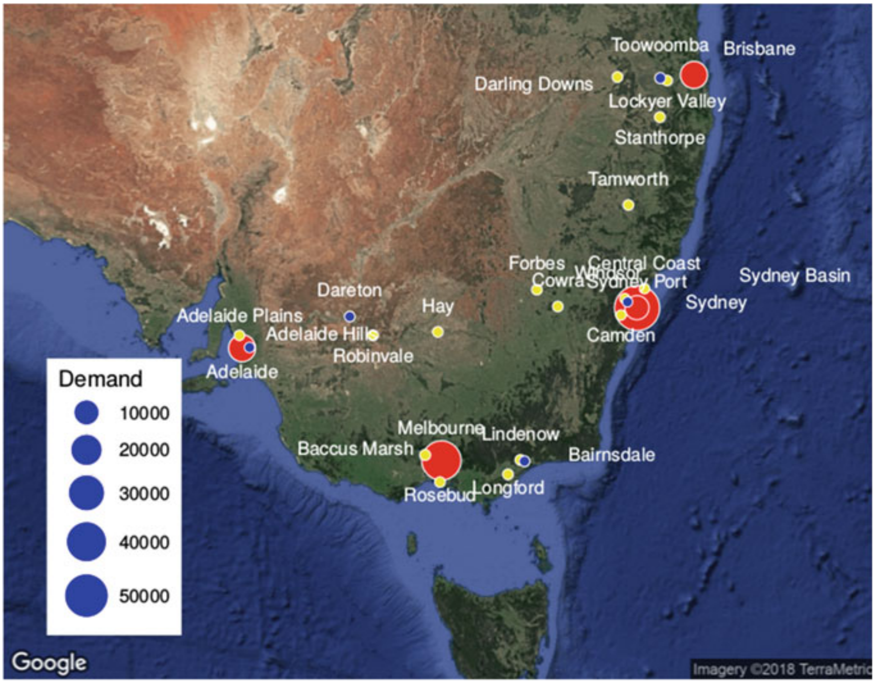


Fig. 2.2 Locations of broccoli producers (yellow) and consumer markets (red) in continental south-east Australia. Candidate processing sites are also producers and are marked in blue. The size of the circle is proportional to the volume in tonnes of broccoli produced or consumed

markets are marked in red. The producers that can be selected as processing sites are marked in blue, and the size of the circle is proportional to the volume in tonnes of broccoli produced or consumed.

Our model addresses the role of uncertainty and variability, which are distinguishing features of agricultural supply chains, and which are caused by its biological nature (e.g.. the genetics of vegetable varieties), the natural environment (like changing weather patterns) and socioeconomic factors (such as changes in demand patterns and government policies). To capture the effect of these factors, we develop a two-stage stochastic mixed-integer linear programming model to assist Australian broccoli producers in taking the most cost-effective investment decisions and, at the same time, reduce the losses by producing novel, high value-added products from produce discarded on the field or during transportation.

This paper is structured as follows. After a review of existing work, Sects. 2.2 and 2.3 describe the model and its stochastic extension, respectively. Section 2.4 discusses the results and Sect. 2.5 rounds up the discussion. The detailed formulation of the problem is included in the Appendix.

2.1.1 Previous Work

Previous research has addressed some aspects of this problem. Regarding the use of optimization models to manage fruit and vegetable supply chains, in general, the literature is extensive and we refer the reader to a recent and comprehensive review on agricultural optimization of supply chains presented in [12]. A review from the point of view of supply chain risk management is provided by [14]. In what follows, we will centre our attention in recent planning models using stochastic programming, modelling of uncertainty in vegetable supply chains, and on relevant projects related to broccoli production.

Atallah [1] presented a fixed charge location problem [3] that aimed at minimizing the total production and transportation costs of fresh broccoli, and analysed the effect of seasonality and of changing the set of production regions, located in the eastern United States, over these costs. Broccoli was chosen because it is a high-value and quite an adaptable vegetable, so it is possible to change the regions where it is produced if new varieties are grown, which makes it ideal to analyse reallocation of production to different zones. The solution of the model in [1] suggests production levels and seasonal product flow that minimize overall supply chain costs. Garcia-Flores et al. [5] and [6] introduced a deterministic model that is the precursor of the stochastic model described in the present paper.

Regarding models that analyse uncertainty in agricultural production and planning, [10] developed a two-stage stochastic programming model focused on minimizing the costs in a competitive market where companies can rent or contract farms to grow up fresh vegetables. Mateo et al. [10] model suggest the best set of farms to contract to the convenience of supermarkets and grocery shops by solving the stochastic model using parallel Lagrangean decomposition. [2] present a very complete two-stage stochastic optimization model that maximizes the economic and environmental benefits of food and biofuel production. These authors use Benders decomposition to solve, in the first stage, land allocation for different types of food and energy crops, while the second-stage variables are operational decisions related to harvesting, budget, location and amounts of different yield types, using also three plausible scenarios based on productivity. Regarding the importance of handling uncertainty in agricultural operations, McKeown [11] reported that an increase in average temperature reduces the yield of broccoli, decreases the contents of vitamin C, and increases the frequency of health-related plant disorders. McKeown [11] reached these conclusions through linear regression models using historical data from the Ontario region.

The mathematical model introduced in the present paper is related to the deterministic model proposed by [1], reviewed above. However, our model differs in that it emphasizes the need to process the product that is not of the right quality for retail by *determining the optimal location and number of processing facilities*, using relevant production data for the Australian market and supply chain conditions [9]. In addition, our model factors in uncertainty by considering a number of scenarios based on productivity and costs. These scenarios are assembled using data from a survey and other publicly available sources. Like the model presented in [1], our model does not factor in the cost of substitution if the growers decide to produce other vegetables with higher value added.

2.2 The Basic Model

The schematic in Fig. 2.3 shows that the broccoli supply chain is composed of *growers*, *packers* and *consumers*. The solid arrow in the figure indicates that the growers' *fresh produce* is sent to the packers to be put into crates, and then transported to the clients (markets and supermarkets). A fraction of the fresh produce becomes a *loss* at three different points of handling: first, some of the produce does not leave the growers' farms, so the broccoli is lost in production. Second, there are losses during transportation to the packers. Finally, the packers themselves also incur in losses due to mishandling. The total losses from these three sources are sent to *processing hubs* for value addition. These processing hubs are selected from among a set of *candidate sites*, which are a subset of the packer sites. Figure 2.3 shows the flow of losses as a dashed line. Some growers may have packing facilities within their premises, but this is not always the case. Consumers are, in general, large urban centres away from the farming areas.

The model addresses the need to process the produce that is not of the right quality for retail by *determining the optimal location and number of processing facilities*, using relevant production data for the Australian market and supply chain conditions, gathered from a field survey and other data sources. The aim of these processing facilities is to reduce or eliminate food loss by producing broccoli-based edible products of high nutritional value such as snacks or food supplement powder.

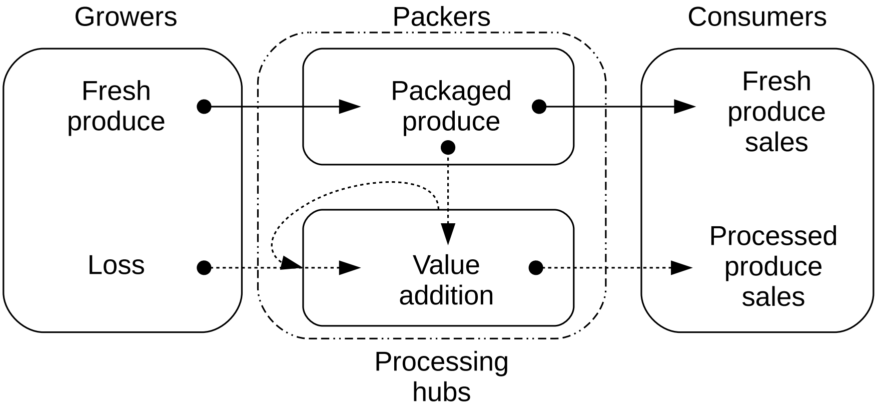


Fig. 2.3 The flows between sites. Fresh produce goes to packers, losses go to processing hubs, which are also packers. Losses may occur during production, transportation or packaging (dotted arrows). Consumers receive value-added products and fresh produce

The full formulation of the problem can be found in the Appendix; a summary of the constraints follows:

1. The *fresh supply* constraints (2.5) ensure that the amount of fresh produce transported from a grower to a packer is no greater than the amount produced minus the losses.
2. The *processed supply* constraints (2.6) ensure that the amount of produce *to be processed* (i.e. originating from losses) transported from a grower to a packer no greater than the losses. Naturally, if there are no losses, there is no supply of produce to the hub.
3. The *yield* constraints (2.7) state the maximum amount that can be produced by a unit of land area.
4. Constraints (2.8) ensure that every grower sends all its produce to be processed to only *one candidate site*.
5. The amount of product to be processed that enters the candidates from packers that were not selected as processing hubs is a fraction of what enters the packers as fresh produce (2.9).
6. The amount of fresh produce that enters a packer is the same amount that leaves, minus the packer's own losses (2.10).
7. The amount of product to be processed that enters a candidate packer, including the packers' losses, is the same amount that leaves (2.11).
8. Constraints (2.12) state that food losses must end up in *one selected candidate processing site*.
9. The selected candidate sites are constrained in their *capacity* to process food loss, as expressed in constraints (2.13). The produce to be processed comes from both growers and packers.
10. The *number of allowed processing sites* may be limited, constraint (2.14).
11. The *fresh demand* constraints (2.15) state that the transported product must satisfy the demand of the consumers.
12. The *processed demand* constraints (2.16) state that the transported product must satisfy the demand of the consumers.

The input data to the model were obtained on the field through a survey applied to broccoli growers nationwide. This survey provides a snapshot of the current situation of this economic activity and reveals the extent of the problem and the interest of stakeholders in solving it.

2.3 The Stochastic Model

The general form of a stochastic optimization problem is

$$\min_{z \in Z} \{g(z) = c^T z + \mathbb{E}[Q(z, \xi)]\} \tag{2.1}$$

where $Q(z, \xi)$ is the optimal value of the second-stage problem

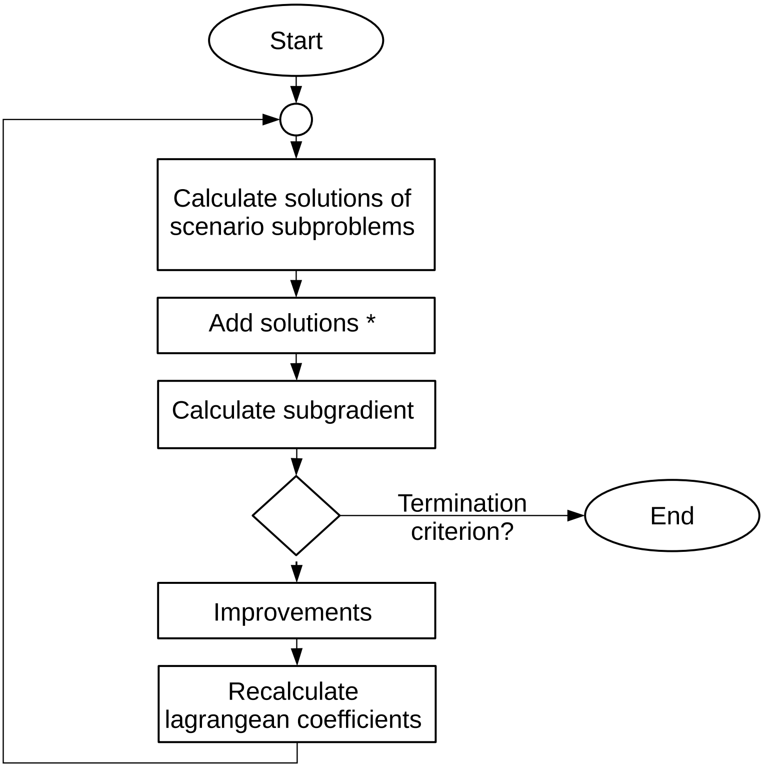

Start
Calculate solutions of scenario subproblems
Add solutions *
Calculate subgradient
Termination criterion?
End
Improvements
Recalculate lagrangean coefficients

Fig. 2.4 Lagrangean decomposition using the subgradient method. The step of addition of solutions, marked with a star, is justified in [8]

$$\min_{y}\ q^T y \quad \text{subject to} \quad Tz + Wy \leq h\ . \tag{2.2}$$

Here, z selects in the first stage of the stochastic model we propose, the optimal location of processing facilities to add value to the produce that would otherwise be considered as food loss, Z is a polyhedral set defined by a finite number of linear constraints, y is the second-stage decision vector comprising production and amount transported, and $\xi = (q, T, W, h)$ contains the data of the second-stage problem [13]. A simplified schematic of the Lagrangean decomposition algorithm and subgradient method are shown in Fig. 2.4. Full details of the methodology can be found in [4] and [10], and are not shown here due to space limitations.

The model described in Sect. 2.2 is extended into a two-stage stochastic version, which must consider a set of scenarios. Scenarios are parameter sets that capture currently available information about possible realizations of market and environment conditions, as well as expected future developments. These scenarios are a *boom*, *fair* and *poor scenarios*. For the purpose of our model, a scenario is defined by the following parameters:

1. *Yield*, which are projections of the productivity of the number of hectares dedicated to broccoli production.
2. *Transportation costs*, defined per tonne per Km, and *production costs* per tonne of produce.
3. *Losses at production, transportation and handling*, which are estimates of the relative losses as a percentage.
4. *Demand for fresh and processed product*, which are estimations of how favourable the produce will be received commercially. The figures of processed product, that is, the losses that become higher value-added food, could be refined by a market analysis of how the consumers will react to new products.

Associated to each scenario is also the *probability* of its occurrence, which is a figure provided by experts based on their knowledge of likely developments in the market and of environmental conditions. The stochastic formulation requires the model in Sect. 2.2 to be extended with *non-anticipativity constraints*, which essentially stipulate that the decision-maker has no way to predict the best scenario, and therefore the decisions taken in the first stage should be the same regardless of the scenario that materializes [4]. These complicating non-anticipativity constraints are added to the objective function with non-negative Lagrangean multipliers. The recourse actions represent the decision-maker's opportunity to compensate.

The model considers three scenarios: a *poor* scenario, characterized by decreasing productivity, high costs, relatively high losses and low demand. In this scenario, the productivity forecast assumes a linear decrease of 1% in productivity starting from the average value per state, an increasing transportation cost along with the scenario from the current figure of $ 0.10/(km ton)–$ 0.15/(km ton), production costs increased from the current $ 1.28 /kg–$ 1.65 at the end of the 3-year horizon. Next, a *fair* scenario, where productivity, costs, losses and demand are approximately the same as at present. This scenario uses a constant transportation cost of $ 0.10/(km ton) and a constant production cost of $ 1.28/kg. Finally, a *boom* scenario defined by increasing productivity, low costs, small losses and high demand. The productivity forecast, in this case, is assumed to increase 1% per quarter starting from the average value per state, a decreasing transportation cost from $ 0.10/(km ton) to $ 0.05/(km ton) at the end of the time horizon, a decreasing production cost until reaching $ 0.92 /kg at the end of the 3 years. The probabilities assigned to *poor*, *boom* and *fair* scenarios are 0.2, 0.5 and 0.3, respectively.

The objective of the stochastic version of the problem is now to determine the location of the processing sites that minimize the overall cost of operating the supply chain along the full time horizon so as to satisfy the uncertain demand amount of produce losses, the uncertain market conditions and the uncertain availability of production resources. In order to generate plausible scenarios to feed our model, we estimate parameter values by means of a survey prepared especially for this purpose, as well as from data available from publicly available sources.

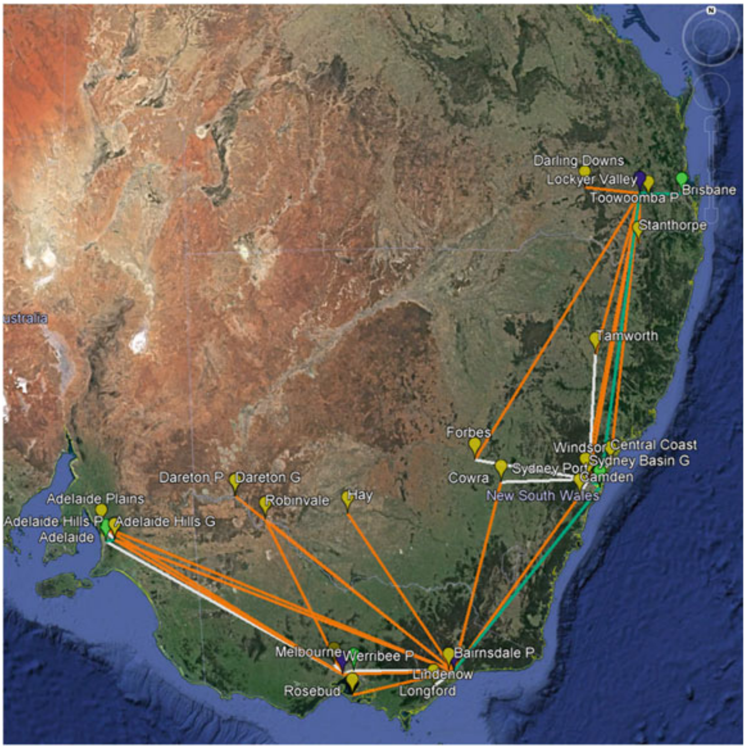


Fig. 2.5 Poor scenario

2.4 Results and Discussion

The problem was implemented in version 1.9 of the Clojure[1] language with an Excel interface. We obtained the following results using version 12.4 of the CPLEX[2] optimizer in a 64-bit Intel Xeon CPU with two processors of eight cores (2.27 GHz) each and 48 GB of RAM. Each of the scenario subproblems has 16539 variables, of which 12564 are real and 3975 are binary and 20536 constraints. A typical run takes around 22 min to complete when running the decomposition problems sequentially.

The number of sites selected is limited to three. Figures 2.5, 2.6 and 2.7 shows the location of the processing hubs, selected from among a set of five candidates that comprised Adelaide Hills, Werribee, Bairnsdale, Sydney Basin and Toowoomba. The processing hubs selected as the first stage of the stochastic model correspond to Werribee, Bairnsdale and Toowoomba and are shown with a dark blue marker. A link in the map is shown if, during the time horizon, there is at least a period during

[1] http://clojure.org/, accessed on the 9 of May 2018.

[2] https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer, accessed on the 9 of May 2018.

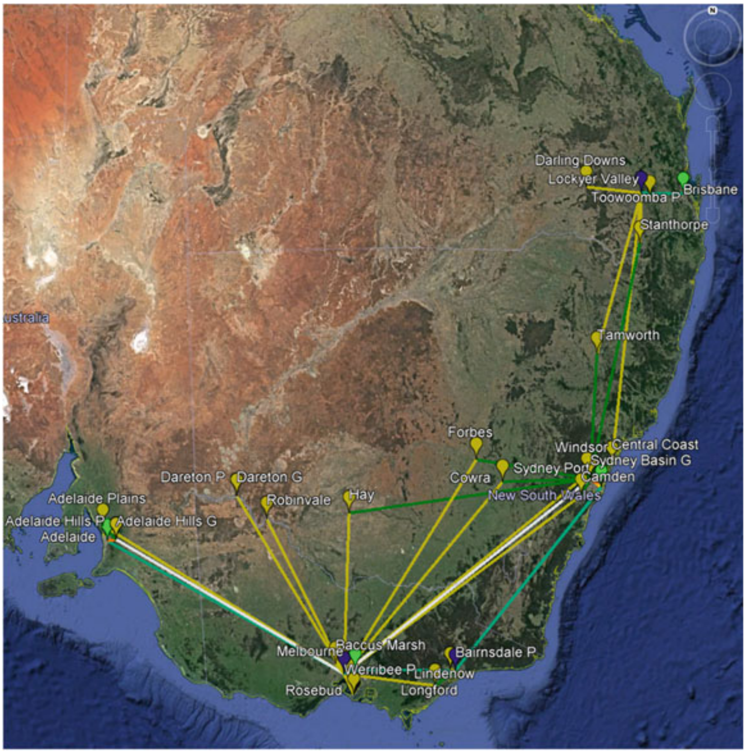


Fig. 2.6 Fair scenario

which the flow through that link was non-zero, which implies that these figures are a limited representation of the solution since the actual solution provides detailed season-by-season flows.

It can be seen from the map that the poor scenario with its declining productivity seems to produce longer connections. As expected, the "middle" regions of Victoria (the most productive) and New South Wales (with the highest demand) concentrate much of the activity under all scenarios.

We also analysed the volumes of produce handled by packers or processors and hubs given the projections for each scenario. Figure 2.8 shows the changes in flows of fresh broccoli over time for each packer under different scenarios. Bairnsdale is the packer that handles the largest volume under all scenarios, due to its centrality. The poor scenario is associated with higher losses, lower yields and longer distances to packers and markets, and as a consequence, all packers handle lower volumes of fresh produce under this scenario. Adelaide Hills and Toowoomba pack nearly identical amounts of fresh broccoli in the boom and fair scenarios, but Toowoomba does surprisingly well in the poor scenario, even increasing its volume as time goes by. The worst performer in the poor scenario is Sydney Basin.

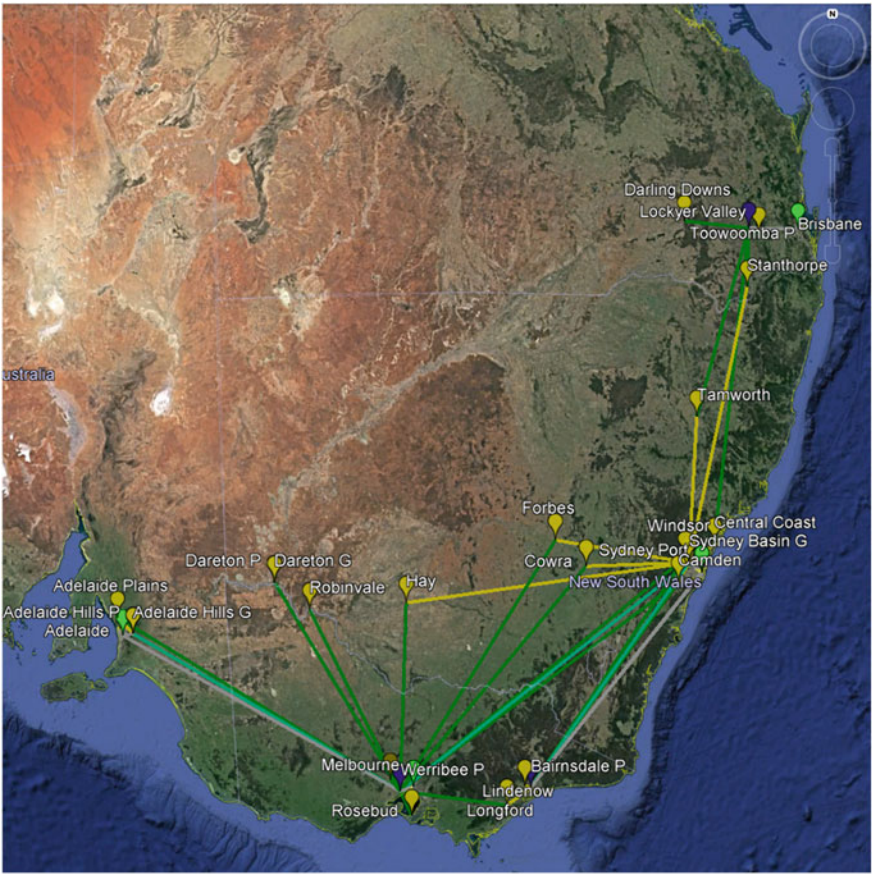


Fig. 2.7 Boom scenario

2.5 Concluding Remarks

It is necessary to increase the efficiency of supply chain operations in order to avoid food loss. Uncertainty and variability in the commercial and natural environments are complicating factors that must be considered when improving the efficiency of agricultural supply chain operations. In the present paper, we introduced a model capable of robust analysis under uncertainty of the broccoli supply chain in the eastern states of Australia. The model aims at reducing food losses by processing the product that is not of the right quality for retail by determining the optimal location and number of processing facilities, using relevant production data for the Australian broccoli market and supply chain conditions.

The two-stage stochastic mixed-integer linear programming model is meant to assist Australian broccoli producers in taking the most cost-effective investment

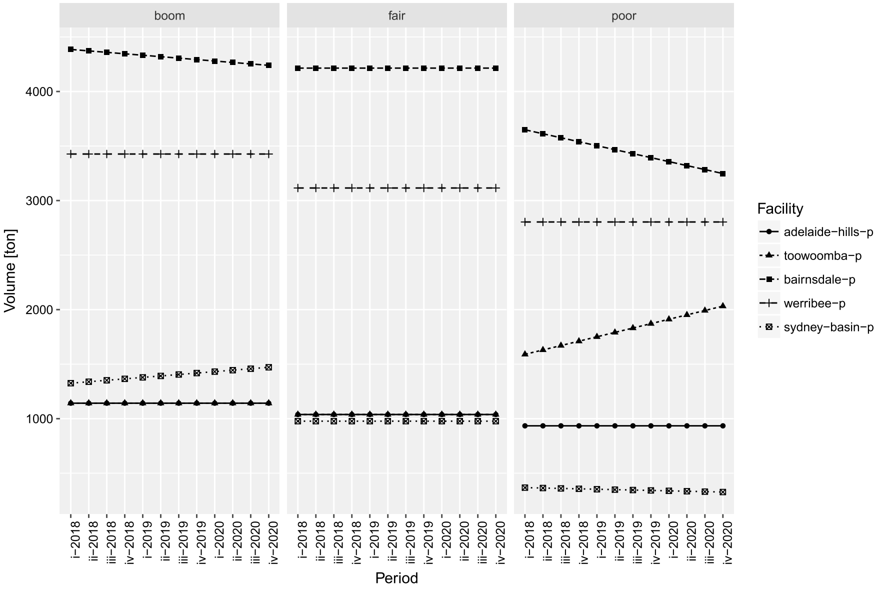


Fig. 2.8 Volumes of fresh broccoli handled by packers on each period for each scenario

decisions and, at the same time, reduce the losses by producing novel, high value-added products from vegetable produce discarded on the field or during transportation. The solution of the model suggests that the optimal location of these facilities are Werribee, Bairnsdale and Toowoomba and gives the opportunity to the decision-maker to compensate by modifying his operations on a second stage, or recourse, decisions. To illustrate, our results show the optimal flow of produce to the packers in the cases of different scenario realizations. This means that the results presented are robust respect to uncertainty in productivity and market conditions.

In combination with the survey applied to the stakeholders, the optimization model suggests near-optimal investment decisions that are far from the worst possible outcome, had the final market and environmental conditions turned out to be very adverse. In summary, our model provides valuable suggestions for practical, realistic and profitable operations to the industry in the medium term.

Appendix

The problem requires the following decision variables. Let w_{itm} the production of product m at site i and season t; x_{ijtm}^{AB} the amount of transported product m from site i to j at season t, where A and B represent the set of the origin and destination sites, respectively; y_{ijtm}^{AB} the amount of transported loss m from site i to j at season t, where A and B represent the set of the origin and destination sites, respectively; z_i

an indicator variable taking the value one if a packer i is selected as a candidate site and zero otherwise, and v_{ijtm} an indicator variable that takes the value one if the loss m is transported from i to j at season t and zero otherwise.

Let the sum of losses at the production and transport stages of product m from grower i at season t.

$$\begin{aligned} H_{itm} &= TL_{itm}\,(1 - PL_{itm})\,\eta_{itm}\,A_{itm} + PL_{itm}\,\eta_{itm}\,A_{itm} \\ &= (TL_{itm} + PL_{itm} - TL_{itm}PL_{itm})\,\eta_{itm}\,A_{itm} \qquad \forall i \in \mathscr{S},\ \forall j \in \mathscr{C},\ \forall k \in \mathscr{K}\ . \end{aligned} \tag{2.3}$$

The objective function is

$$\begin{aligned} \text{Minimize} & \sum_{i\in\mathscr{G}}\sum_{t\in\mathscr{T}}\sum_{m\in\mathscr{M}} PC_{itm}\,w_{itm} \\ & + \sum_{t\in\mathscr{T}}\sum_{m\in\mathscr{M}} TC_{tm}\Big[\sum_{i\in\mathscr{G}}\sum_{j\in\mathscr{P}} D_{ij}\,x^{GP}_{ijtm} + \sum_{i\in\mathscr{P}}\sum_{j\in\mathscr{S}} D_{ij}\,x^{PS}_{ijtm}\Big] \\ & + \sum_{t\in\mathscr{T}}\sum_{m\in\mathscr{M}} TC_{tm}\Big[\sum_{i\in\mathscr{G}}\sum_{j\in\mathscr{C}} D_{ij}\,y^{GC}_{ijtm} + \sum_{i\in\mathscr{C}}\sum_{j\in\mathscr{S}} D_{ij}\,y^{CS}_{ijtm} + \sum_{i\in\mathscr{P}}\sum_{j\in\mathscr{C}} D_{ij}\,y^{PC}_{ijtm}\Big] \\ & + \sum_{i\in\mathscr{C}} F_i\,z_i\ , \end{aligned} \tag{2.4}$$

Subject to

$$\sum_{j\in\mathscr{P}} x^{GP}_{ijtm} = w_{itm} - H_{itm} \qquad \forall i \in \mathscr{G},\ \forall t \in \mathscr{T},\ \forall m \in \mathscr{M}\ . \tag{2.5}$$

$$y^{GC}_{ijtm} = H_{itm}\,v_{ijtm} \qquad \forall i \in \mathscr{G},\ \forall j \in \mathscr{C},\ \forall t \in \mathscr{T},\ \forall m \in \mathscr{M}\ . \tag{2.6}$$

$$w_{itm} \le \eta_{itm}\,A_{itm} \qquad \forall i \in \mathscr{G},\ \forall t \in \mathscr{T},\ \forall m \in \mathscr{M}\ , \tag{2.7}$$

where η_{itm} is the yield in kg per unit of area and A_{itm} is the land area available for cultivation.

$$\sum_{j\in\mathscr{C}} v_{ijtm} = 1 \qquad \forall i \in \mathscr{G},\ \forall t \in \mathscr{T},\ \forall m \in \mathscr{M}\ . \tag{2.8}$$

$$\begin{aligned} &\sum_{j\in\mathscr{C}} y^{PC}_{ijtm} = KL_{itm} \sum_{j\in\mathscr{G}} x^{GP}_{jitm} \\ &\qquad\qquad \forall i \in \mathscr{P},\ \forall t \in \mathscr{T},\ \forall m \in \mathscr{M}\ . \end{aligned} \tag{2.9}$$

$$\sum_{i \in \mathscr{G}} x^{GP}_{ijtm} = \sum_{i \in \mathscr{S}} x^{PS}_{jitm} + \sum_{i \in \mathscr{C}} y^{PC}_{jitm} \qquad \forall j \in \mathscr{P},\ \forall t \in \mathscr{T},\ \forall m \in \mathscr{M}\ . \tag{2.10}$$

$$\sum_{i \in \mathscr{G}} y^{GC}_{ijtm} + \sum_{i \in \mathscr{P} \supset \mathscr{C}} y^{PC}_{ijtm} = \sum_{i \in \mathscr{S}} y^{CS}_{jitm} \qquad \forall j \in \mathscr{C},\ \forall t \in \mathscr{T},\ \forall m \in \mathscr{M}\ . \tag{2.11}$$

$$v_{ijtm} - z_j \leq 0 \qquad \forall i \in \mathscr{G},\ j \in \mathscr{C},\ \forall t \in \mathscr{T},\ \forall m \in \mathscr{M}\ . \tag{2.12}$$

$$\sum_{i \in \mathscr{G}} y^{GC}_{ijtm} + \sum_{i \in \mathscr{P} \setminus \mathscr{C}} y^{PC}_{ijtm} - C_j\, z_j \leq 0 \qquad \forall j \in \mathscr{C},\ \forall t \in \mathscr{T},\ \forall m \in \mathscr{M}\ , \tag{2.13}$$

where C_j is the capacity of candidate processing site j.

$$\sum_{i \in \mathscr{C}} z_i \leq MAX\ , \tag{2.14}$$

where MAX is the maximum number of processing sites allowed.

$$\sum_{i \in \mathscr{P}} x^{PS}_{ijtm} \geq W^{\text{fresh}}_{jtm} \qquad \forall j \in \mathscr{S},\ \forall t \in \mathscr{T},\ \forall m \in \mathscr{M}\ . \tag{2.15}$$

$$\sum_{i \in \mathscr{P}} y^{CS}_{ijtm} \geq W^{\text{processed}}_{jtm} \qquad \forall j \in \mathscr{S},\ \forall t \in \mathscr{T},\ \forall m \in \mathscr{M}\ . \tag{2.16}$$

References

1. Atallah S, Gómez M, Björkman T (2014) Localization effects for a fresh vegetable product supply chain: Broccoli in the eastern United States. Food Policy 49:151–159
2. Cobuloglu H, Büyüktahtakin I (2017) A two-stage stochastic mixed-integer programming approach to the competition of biofuel and food production. Comput Industr Eng 107:251–263
3. Current J, Daskin M, Schilling D (2004) Facility location: applications and theory, 2nd edn, Springer, New York, chap 3—Discrete network location models, pp 81–118
4. Escudero L, Garín M, Pérez G, Unzueta A (2012) Lagrangian decomposition for large-scale two-stage stochastic mixed 0–1 problems. TOP 20(2):347–374
5. García-Flores R, Juliano P, Cohan S, Petkovic K (2017) A basic supply chain optimisation model to reduce broccoli and banana losses. Commercial in Confidence EP178474, CSIRO
6. García-Flores R, Juliano P, Petkovic K (2018) Handling food waste and losses. Criticalities and methodologies, vol Sustainable Food Supply Chains: Planning, Design, and Control through Interdisciplinary Methodologies, Elsevier, chap 16. Submitted

7. Godfray H, Beddington J, Crute I, Haddad L, Lawrence D, Muir J, Pretty J, Robinson S, Thomas S, Toulmin C (2010) Food security: the challenge of feeding 9 billion people. Science 327(5967):812–818
8. Guignard M, Kim S (1987) Lagrangian decomposition. a model yielding stronger Lagrangian bounds. Math Prog 39:215–228
9. Horticultural Innovation Australia Limited (2017) Australian horticulture statistics handbook 2015/2016. Tech. rep., Horticultural Innovation Australia Limited, Sydney, NSW, https://horticulture.com.au/wp-content/uploads/2017/10/Horticulture-Statistics-Handbook-2015-16-Vegetables-New.pdf, Accessed on 21 May 2018
10. Mateo J, Pla L, Solsona F, Pagès A (2016) A production planning model considering uncertain demand using two-stage stochastic programming in a fresh vegetable supply chain context. SpringerPlus 5(839):1–16
11. McKeown A, Warland J, McDonald M (2004) Cool season crop production trends: A possible signal for global warming. In: Bertschinger L, Anderson J (eds) Proceedings of the XXVI IHC – Sustainability of Horticultural Systems, no. 638 in Acta Horticulturae, pp 241–248
12. Paam P, Berretta R, Heydar M, Middleton R, García-Flores R, Juliano P (2016) Planning models to optimize the agri-fresh food supply chain for food loss minimisation: A review, Elsevier. Elsevier's Reference Module in Food Science
13. Shapiro A, Philpott A (2007) A tutorial on stochastic programming. Tech. rep., http://www.isye.gatech.edu/people/faculty/Alex_Shapiro/TutorialSP.pdf, accessed on 21 May 2018
14. Sodhi M, Son B, Tang C (2012) Researchers' perspectives on supply chain risk management. Product Operat Manag 21(1):1–113

Chapter 3
Integrating Shelf Life Constraints in Capacitated Lot Sizing and Scheduling for Perishable Products

Shuo Chen, Regina Berretta, Alexandre Mendes, and Alistair Clark

Abstract In this study, we consider a multi-item capacitated lot sizing and scheduling problem for perishable food products, which have a fixed shelf life period due to depreciation issues, such as physical deterioration or perceived value loss. We incorporate shelf life constraints within a classical lot sizing and scheduling problem that considers lot sizing and partial sequencing of production on a single machine over a finite planning horizon associated with demand in each period. The model includes setup times and setup costs. Moreover, it considers that disposal occurs when products reach their shelf life in inventory. We present two variants of this lot sizing and scheduling problem integrating shelf life constraints and also with and without disposal. We test the performance of the proposed formulations on a set of instances from the literature.

Keywords Lot sizing and scheduling · Perishability · Shelf life · Disposal

S. Chen (✉) · R. Berretta · A. Mendes
School of Electrical Engineering and Computing, The University of Newcastle,
Callaghan 2308, NSW, Australia
e-mail: sylvia.chen@uon.edu.au

R. Berretta
e-mail: regina.berretta@newcastle.edu.au

A. Mendes
e-mail: alexandre.mendes@newcastle.edu.au

A. Clark
Director of Research and Scholarship, Faculty of Environment and Technology,
University of the West of England, Bristol BS16, 1QY, UK
e-mail: Alistair.Clark@uwe.ac.uk

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_3

3.1 Introduction

Perishability of products makes food companies arrange their production planning carefully [9]. When products are subject to perishability or deterioration, the time these products can spend in storage is limited. In the worst case, such products cannot be used for their original purpose and need to be disposed of. Perishability within the food manufacturing industry is responsible for 39% of annual food losses, estimated at 179 kg/head [11].

In the food industry, companies are responsible for placing a date on the food package to guide how long food can be kept before it begins to deteriorate or may become unsafe to eat, because of related quality requirements and safety regulations. Food consumers are susceptible to health and safety issues; they do not like to buy food after the expiry date or the "best before" date. Food becomes unusable at some point in time, such as dairy goods [5], considering the desired sensory, chemical, physical and microbiological characteristics as well as complying with the label's declaration of nutritional data. Accordingly, the realization of a specific demand must take place within a certain period of time, i.e. within a *fixed shelf life*. Thus, the shelf life constraints are considered as the fixed maximum time during which the food can stay in inventory. During the shelf life period, the value of the product is assumed to be constant.

There are few lot sizing models in the literature discussing the incorporation of shelf life constraints in mathematical models for production planning [1, 2, 5, 10, 12, 14]. However, the simultaneous consideration of lot sizing and scheduling is rare [16]. For reviews on how deterioration and lifetime constraints are integrated into production and supply chain planning, see Pahl and Voβ [9] and the references therein.

In this paper, we integrate shelf life constraints into a dynamic capacitated lot sizing problem with linked lot sizes (CLSP-L). The CLSP-L model considers a single machine producing multiple products over a finite planning horizon, which aims at fulfilling all products' demands within the planning horizon and minimizing total costs, i.e. setup costs and inventory holding costs. Demand is dynamic and deterministic, derived from forecasting, and must be satisfied at a given moment in time without backlogging [15]. A setup (such as the cleaning and warm-up of a machine) is required when a new product is processed, which results in a setup cost and a setup time that precedes the processing of each change of product. CLSP-L allows a setup state to be carried over from one period to the next in order to reduce cost and time consumption in consecutive periods.

Another important feature considered in this study is food *disposal*, which is wastage that occurs when the shelf life of food in inventory is exceeded. It is responsible for the largest food losses in the retail sector. Around one-third of all fresh fruit and vegetables produced worldwide are lost before reaching consumers [8]. Thus, disposal should be considered in the modelling of perishable products, as it imposes additional characteristics, such as more frequent deliveries compared to non-perishable products [7].

Shelf life constraints on a lot sizing problem were studied by Raiconi et al. [12] using the Plant location formulation (PLF) and the Shortest path formulation (SPF). Tempelmeier et al. [16] considered a CLSP-L model with multiple machines and shelf life constraints using the PLF formulation. Both papers use an index transformation method that restricts production due to shelf life. The production must be less than or equal to the sum of demands over the shelf life period [9]. In this paper, instead of using index transformation, we integrate shelf life constraints into the production flow balance.

In the rest of this paper, we first present the CLSPL-L base model in Sect. 3.2, which will be extended to incorporate perishability constraints. We then introduce two variants (Models 1 and 2) to integrate fixed shelf life constraints into the CLSP-L model. Next, we extend the models to consider the disposal of products. In Sect. 3.3, we report computational results where all models are tested on a set of instances from the literature. Section 3.4 gives the final remarks and conclusion.

3.2 Mathematical Formulations

3.2.1 CLSP-L Model

The CLSP-L assumes multiple products are produced over a finite planning horizon on a single machine with limited capacity. The finite planning horizon consists of several periods with the given length. The production of a product incurs a setup operation resulting in a sequence-independent setup time and setup cost. The triangle inequality holds for all setup times and costs, which means the direct changeover between two products is never slower or more expensive than the changeover via a third product. A setup state for a product in a period can be maintained over several consecutive periods if there is no other product to process. Table 3.1 shows the notations for the mathematical models in this paper, followed by the CLSP-L mathematical formulation (from [6]).

$$\textbf{[CLSP-L]}\quad \text{Min} \quad \sum_{i=1}^{P}\sum_{t=1}^{T} h_i I_{it} + \sum_{i=1}^{P}\sum_{t=1}^{T} sc_i (Y_{it} - W_{it}) \tag{3.1}$$

$$s.t. \quad I_{i,t-1} + Q_{it} = d_{it} + I_{it} \qquad \forall i, t \tag{3.2}$$

$$\sum_{i=1}^{P} u_i Q_{it} + \sum_{i=1}^{P} st_i (Y_{it} - W_{it}) \leq C_t \qquad \forall t \tag{3.3}$$

$$Q_{it} \leq U_{it} Y_{it} \qquad \forall i, t \tag{3.4}$$

$$\sum_{i=1}^{P} W_{it} \leq 1 \qquad \forall t \tag{3.5}$$

Table 3.1 Parameters and variables used in the mathematical formulations

Indices	**Definition**
i, j	Index for products, $i, j = 1, 2, ..., P$
t	Index for periods in the planning horizon, $t = 1, 2, ..., T$
Parameters	**Definition**
u_i	Capacity needed to produce one unit of product i
h_i	Unit holding cost of product i
C_t	Capacity available in period t
d_{it}	Demand for product i in period t
sc_i	Setup cost for product i
st_i	Setup time for product i
U_{it}	Upper bound for the production of product i in period t
Variables	**Definition**
I_{it}	Inventory for product i at the end of period t
Q_{it}	Production quantity for product i in period t
S_t	Single-item production indicator (=0 if at most one product is produced in period t; otherwise 1 when more than one product is produced)
Y_{it}	Setup state (=1 if a setup for product i occurs in period t; otherwise 0)
W_{it}	Carryover state (=1 if a setup state for product i is carried over from period $t-1$ to period t; otherwise 0)

$$W_{it} \le Y_{i,t-1} \qquad \forall i, t = 2, ..., T \tag{3.6}$$

$$W_{it} \le Y_{i,t} \qquad \forall i, t = 2, ..., T \tag{3.7}$$

$$1 - \sum_{i=1}^{P} Y_{it} + P \cdot S_t \ge 0 \qquad \forall i, t = 2, ..., T-1 \tag{3.8}$$

$$W_{i,t+1} + W_{it} + S_t \le 2 \qquad \forall i, t = 2, ..., T-1 \tag{3.9}$$

$$Q_{it}, I_{it} \ge 0 \qquad \forall i, t \tag{3.10}$$

$$Y_{it}, W_{i,t} \in \{0, 1\} \qquad \forall i, t \tag{3.11}$$

$$0 \le S_t \le 1 \qquad \forall t \tag{3.12}$$

The objective function (3.1) minimizes the total holding cost and setup cost over the planning horizon. Constraints (3.2) represent production and inventory balance and assure the demand d_{it} is fulfilled without backlogging. Constraints (3.3) ensure that production does not exceed the capacity available in period t, considering setup time, setup carryover and production time. Constraints (3.4) guarantee that production of a product can only take place if the machine is configured for that product. Constraints (3.5) ensure that only one setup can be carried over from one period to the next. Constraints (3.6) and (3.7) link the carryover variables to the setup variables, whereas constraints (3.8) and (3.9) make sure that the setups are calculated correctly if the same setup is preserved on two consecutive periods. The production quantity Q_{it} and inventory I_{it} variables are non-negative, and setup state S_t, single production indicator Y_{it} and carryover W_{it} variables are binary, as indicated in (3.10), (3.11) and (3.12).

3.2.2 Incorporating Shelf Life Constraints

To extend the model (3.1)–(3.12) in order to integrate shelf life constraints, we first define a new parameter and two new decision variables as follows:

- θ_i—shelf life of product i, defined as the maximum number of periods that product i can be stored in the inventory.
- Q_{i,t_1,t_2}—disaggregated production quantity, which represents the amount of product i produced in period t_1 to satisfy the demand in period t_2, $t_2 \leq t_1 + \theta_i$.
- I_{i,t_1,t_2}—disaggregated inventory, which represents the inventory of product i produced in period t_1 that is kept at the end of period t_2, $t_2 \leq t_1 + \theta_i$.

We need to replace only constraints (3.2) with a new set of constraints since shelf life imposes direct restrictions on inventory rather than capacity and scheduling. In addition, this replacement makes the shelf life constraints easily transferred to other lot sizing and scheduling models. We present two variants in the next two subsections.

3.2.2.1 Shelf Life Constraints—Model 1

The first approach uses the disaggregated production and disaggregated inventory variables to track the flow of production and inventory. It takes into account the shelf life to decide when to produce and when products are used to satisfy the demand. Constraints (3.2) are replaced by the following constraints:

$$\textbf{[Model 1]} \quad Q_{it} = Q_{itt} + I_{itt} \qquad \forall i, t \tag{3.13}$$

$$I_{i,t,t_2} = I_{i,t,t_2-1} - Q_{i,t,t_2} \qquad \forall i, t = 1, ..., t_2 - 1 \tag{3.14}$$

$$d_{it} = \sum_{t_1=max\{1,t-\theta_i\}}^{t} Q_{i,t_1,t} \qquad \forall i, t \tag{3.15}$$

$$I_{it} = \sum_{t_1=1}^{t} I_{i,t_1,t} \qquad \forall i, t \tag{3.16}$$

$$I_{i,T} = 0 \qquad \forall i \qquad (3.17)$$

Constraints (3.13) and (3.14) refer to the inventory balance. Constraints (3.13) state that product i produced in period t (Q_{it}) can be used to either satisfy demand (Q_{itt}) or be stored in inventory at the end of this period (I_{itt}). Constraints (3.14) state that the inventory of product i produced in period t at the end of period t_2 (I_{i,t,t_2}) is composed of the inventory from the previous period $t_2 - 1$, less the disaggregated production Q_{i,t,t_2} in that period. Constraints (3.15) indicate that the demand of product i in period t can be met by production over several periods. The demand of product i in periods $t \leq \theta_i$ can be satisfied by the production in all previous periods from 1 to t. The demand of product i in periods $t > \theta_i$ can be satisfied only with the production in the interval $[t - \theta_i, t]$. The length of the interval is decided by the shelf life of product i. The period $t - \theta_i$ denotes the earliest period in which the product i can be produced and still be used for satisfying demand in period t. Constraints (3.16) state that the inventory of product i in period t is equal to the inventory of product i produced in all previous periods and stored until period t. Finally, constraint (3.17) states the final inventory ($I_{i,T}$) for all products is 0.

3.2.2.2 Shelf Life Constraints—Model 2

The second approach uses the disaggregated quantity to decompose the relationship between demand, inventory and production. Constraints (3.2) are replaced by the following constraints:

$$\textbf{[Model 2]} \qquad I_{i,t-1} + \sum_{t_2=t}^{min\{T,t+\theta_i\}} Q_{i,t,t_2} = I_{it} + d_{it} \qquad \forall i, t \qquad (3.18)$$

$$I_{it} = \sum_{t_1=1}^{t} \sum_{t_2=t+1}^{min\{T,t_1+\theta_i\}} Q_{i,t_1,t_2} \qquad \forall i, t \qquad (3.19)$$

$$Q_{it} = \sum_{t_2=t}^{min\{T,t+\theta_i\}} Q_{i,t,t_2} \qquad \forall i, t \qquad (3.20)$$

Constraints (3.18) represent the inventory balance. Constraints (3.19) indicate that the inventory of product i in period t is equal to the production in all previous periods $t_1 \in [1, t]$ that will be used to satisfy the demand in late periods from $t + 1$ to the end of the planning horizon or the end of their shelf life ($min\{T, t_1 + \theta_i\}$), which imposes the final inventory to be 0. Next, constraints (3.20) state that the total amount of production of i in period t (Q_{it}) is equal to the total production of i in period t that is used to satisfy the demand from period t to the end of its shelf life or the end of planning horizon (Q_{i,t,t_2}).

3.2.3 Incorporating Disposal Decisions

In general, food products, such as agricultural products, meat and dairy have a minimum production quantity, due to the nature of supply of their raw materials. This can result in large amounts of waste when products expire while in inventory. In addition, with the trend towards greater uncertainty of demand and an increase in the sales of fresh food products, the food industry has to maintain large amounts of product inventories to satisfy demand on time [13], which can also result in disposal as waste. This leads to the need to consider the costs of disposal in the mathematical formulation.

In this section, we extend the two models presented in Sect. 3.2.2 to incorporate disposal with shelf life constraints into the capacitated lot sizing and scheduling model. Two additional parameters and an extra variable are introduced to model disposed quantity and associated cost, as follows:

- D_{i,t_1,t_2}—disaggregated disposed quantity, which represents the amount of production of i in period t_1 that is discarded in period t_2, where $t_2 \geq t_1$.
- dc_i—unit disposal cost, which is assumed to be relatively higher than inventory holding costs, due, for example, to environmental reasons [10].
- M_i—minimum production quantity of product i.

The aim of the new model becomes to determine an optimal production planning such that the total holding costs, setup costs and disposal costs are minimized. Thus, the objective function (3.1) is extended as shown in (3.21). In addition, the minimum production quantity constraint (3.22) is added into the model, which assumes a minimum production amount is imposed on the lot sizes when a setup occurs even if the setup state is carried to next period.

$$Min \quad \sum_{i=1}^{P}\sum_{t=1}^{T} h_i I_{it} + \sum_{i=1}^{P}\sum_{t=1}^{T} sc_i (Y_{it} - W_{it}) + \sum_{i=1}^{P}\sum_{t=1}^{T} dc_i D_{it} \tag{3.21}$$

$$Q_{it} \geq M_i (Y_{it} - W_{it}) \qquad \forall i, t \tag{3.22}$$

In the next two subsections, we extend the set of constraints presented in Sects. 3.2.2.1 and 3.2.2.2 to incorporate disposal decisions.

3.2.3.1 Incorporating Disposal Decisions in Model 1

When disposal is introduced in the Model 1, the constraints (3.2) are replaced as follows:

[Model 1-d]

$$Q_{it} = I_{itt} + Q_{itt} + D_{itt} \qquad \forall i, t \tag{3.23}$$

$$I_{i,t,t_2} = I_{i,t,t_2-1} - Q_{i,t,t_2} - D_{i,t,t_2} \qquad \forall i, t_2 = 2, ..., T, t = 1, ..., t_2 - 1 \tag{3.24}$$

$$D_{i,t,t_2} \geq Q_{it} - \sum_{\tau=t}^{t+\theta_i} Q_{i,t,\tau} - \sum_{\tau=t}^{t_2-1} D_{i,t,\tau} \qquad \forall i, t_2 - t \geq \theta_i \tag{3.25}$$

$$d_{it} = \sum_{t_1=max\{1,t-\theta_i\}}^{t} Q_{i,t_1,t} \qquad \forall t \tag{3.26}$$

$$I_{it} = \sum_{t_1=1}^{t} I_{i,t_1,t} \qquad \forall i, t \tag{3.27}$$

$$D_{it} = \sum_{t_1=1}^{t} D_{i,t_1,t} \qquad \forall i, t \tag{3.28}$$

$$I_{i,T} = 0 \qquad \forall i \tag{3.29}$$

Constraints (3.23) and (3.24) indicate the inventory balance, which is similar to constraints (3.13) and (3.14), but allowing disposal. Constraints (3.25) define the amount of production of i in period t that is disposed in period t_2 (D_{i,t,t_2}). It is calculated by the amount of production of i in period t subtracted by the sum of used quantity to satisfy demand in the interval $[t, t + \theta_i]$ and the quantity disposed in the interval $[t, t_2 - 1]$. Note that the condition $t_2 - t > \theta_i$ implies that the production of i in period t can only be used to meet the demand from period t to $t + \theta_i$. Constraints (3.27) state that the inventory of product i in the period t is equal to the disaggregated inventory of i in all previous periods and stored until period t. Constraints (3.28) state that the amount of disposal of product i in period t (D_{it}) is equal to the total amount of disposal of product i in period t that is produced from period 1 to the current period t ($D_{i,t_1,t}$). Finally, the inventory at the end of planning horizon for all products ($I_{i,T}$) is set to 0.

It is worthy mentioning that constraint (3.23) reflects an unrealistic disposal of product i in period t. In the real world, manufactured products are used to satisfy demand or to store in the inventory, even if the quantity is more than the demand and the products reach their shelf life limitation in the inventory. In general, disposal happens at the end of the product's shelf life due to uncertain and fluctuant demand. However, the setting in constraint (3.23) allows a possible decision for sales promotion.

3.2.3.2 Incorporating Disposal Decisions in Model 2

Similarly, the equation constraint (3.2) in CLSP-L is replaced by the following equations to integrate disposal into Model 2.

[Model 2-d]

$$I_{i,t-1} + \sum_{t_2=t}^{min\{T,t+\theta_i\}} Q_{i,t,t_2} = I_{it} + d_{it} + D_{it} \qquad \forall i,t \qquad (3.30)$$

$$I_{it} = \sum_{t_1=1}^{t} \sum_{t_2=t+1}^{min\{T,t_1+\theta_i\}} Q_{i,t_1,t_2} + \sum_{t_1=1}^{t} \sum_{t_2=t+1}^{T} D_{i,t_1,t_2} \qquad \forall i,t \qquad (3.31)$$

$$Q_{it} = \sum_{t_2=t}^{min\{T,t+\theta_i\}} Q_{i,t,t_2} + \sum_{t_2=t}^{T} D_{i,t,t_2} \qquad \forall i,t \qquad (3.32)$$

$$D_{it} = \sum_{t_1=1}^{t} D_{i,t_1,t} \qquad \forall i,t \qquad (3.33)$$

Constraints (3.30), (3.31) and (3.32) are similar to constraints (3.18), (3.19) and (3.20), but include the disposal quantity relationship. Finally, the constraint (3.33) states that the amount of disposal of product i in period t (D_{it}) is equal to the total amount of disposal of product i in period t that is produced from period 1 to the current period t ($D_{i,t_1,t}$).

3.3 Computational Experiments and Analysis

In this section, we present the computational results to compare the performance of the mathematical formulations presented in the previous sections (Model 1, Model 2, Model 1-d and Model 2-d). We have used 55 instances from a dataset from the literature [15], which are grouped as follows:

- Group 1: 5 instances with 4 products and 15 periods.
- Group 2: 5 instances with 6 products and 15 periods.
- Group 3: 5 instances with 8 products and 15 periods.
- Group 4: 5 instances with 12 products and 15 periods.
- Group 5: 20 instances with 10 products and 20 periods.
- Group 6: 10 instances with 20 products and 20 periods.
- Group 7: 5 instances with 30 products and 20 periods.

Since the original instances do not take into account shelf life constraints and disposal, we added that extra information to run our models. For each instance, we added different values for shelf life (2, 4, 6 and 10 periods), with minimum production quantity of 100 for each product, and disposal cost of 1.75 times of holding cost for each product.

All models were implemented using Python 3.6.5 programming language and solved using Gurobi[1] optimization solver. The computational tests were done on an AMD Opteron(tm) Processor 6386 SE (2.8 gigahertz) workstation with 1 core used

[1] http://www.gurobi.com/.

and 6 gigabytes of RAM. The running time was limited to 1 h. For all tests, the relatively mixed-integer programming (MIP) gap tolerance value was set to 0 which is 1e-4 by default. The results are shown in Tables 3.2-3.4.

In Table 3.2, it is shown how many times the models found an optimal solution (represented as 'O' in the headings), how many times they finished with a feasible solution that is not the optimal one (represented as 'F' in the headings) and how many times no feasible solution was found after 1 hour of computational time (represented as 'I' in the headings). From Table 3.2, we can observe that both models can find optimal solutions in most cases. Model 1 and Model 2 have the same proportion of optimal, feasible and infeasible solutions (113, 99 and 8, respectively). Model 1-d find the optimal solution in two instances more than Model 2-d (108 vs 106). However, again, the results are very similar. In particular, the optimal solution is always found in groups 1–4. In groups 5–7, the small shelf life (when shelf life $= 2$) means that the instances are infeasible in some cases. In groups 6 and 7, an optimal is never encountered for longer shelf lives (when shelf life ≥ 4), perhaps because the number of variables is larger. However, infeasibility is never encountered since a longer shelf life maybe provides more inventory flexibility.

Next, Table 3.3 presents the average running time in seconds for each group of instances (column 1), for each shelf life value (column 2) for each model (columns 3–6). The number in parentheses indicates the quantity of instances where Model 1 was faster than Model 2 (under column Model 1), the quantity of instances where Model 2 was faster than Model 1 (under column Model 2), and analogously for columns Model 1-d and Model 2-d. The last row shows the total number of instances where the specific model was faster. Note in Table 3.3, when the shelf life increases, the computational time also increases. One possible reason is that with a higher shelf life, the model will have more variables (see the definition of Q_{i,t_1,t_2} and I_{i,t_1,t_2}). Note as well that Model 1 is slightly faster, but Model 1-d is significantly faster in the number of instances when compared with Model 2 (61 x 60) and Model 2-d (87 x 31), respectively. Groups 3 and 4 are exceptions, where Model 2 (Model 2-d) is faster than Model 1 (Model 1-d) in a greater number of instances with higher shelf life (4, 6 and 10).

Table 3.4 compares the objective function for the groups where the model has not achieved the optimal solution after 1 hour of computational time (groups 5, 6 and 7). The heading 'MY x MZ' indicates the number of instances, where Model Y achieved a better solution than Model Z when a feasible solution is found. Note that Models 2 and 2-d (with or without disposal) achieve better solutions than Model 1 and Model 1-d, respectively. However, in groups 5 and 6, Models 2 and 2-d tend to find a better solution for higher d-values (when there are more variables), but this is not shown in Group 7 (where maybe the feasible solution is a long way from optimality given that there are 30 products and 20 time periods).

Table 3.2 Number of times that the models found an optimal solution (labelled "O"): suboptimal, but feasible solutions (labelled "F"); or infeasible solutions (labelled "I"), for each group of instances. The maximum allowed processing time was 1 h for each run

Groups	Shelf life	Model 1	Model 2	Model 1-d	Model 2-d
		O/F/I	O/F/I	O/F/I	O/F/I
Group 1	2	5/-/-	5/-/-	5/-/-	5/-/-
	4	5/-/-	5/-/-	5/-/-	5/-/-
	6	5/-/-	5/-/-	5/-/-	5/-/-
	10	5/-/-	5/-/-	5/-/-	5/-/-
Group 2	2	5/-/-	5/-/-	5/-/-	5/-/-
	4	5/-/-	5/-/-	5/-/-	5/-/-
	6	5/-/-	5/-/-	5/-/-	5/-/-
	10	5/-/-	5/-/-	5/-/-	5/-/-
Group 3	2	5/-/-	5/-/-	5/-/-	5/-/-
	4	5/-/-	5/-/-	5/-/-	5/-/-
	6	5/-/-	5/-/-	5/-/-	5/-/-
	10	5/-/-	5/-/-	5/-/-	5/-/-
Group 4	2	5/-/-	5/-/-	5/-/-	5/-/-
	4	5/-/-	5/-/-	5/-/-	5/-/-
	6	5/-/-	5/-/-	5/-/-	5/-/-
	10	5/-/-	5/-/-	5/-/-	5/-/-
Group 5	2	9/7/4	9/7/4	8/8/4	7/9/4
	4	5/15/-	5/15/-	5/15/-	5/15/-
	6	5/15/-	5/15/-	5/15/-	5/15/-
	10	5/15/-	5/15/-	5/15/-	5/15/-
Group 6	2	6/1/3	6/1/3	5/-/5	4/1/5
	4	-/10/-	-/10/-	-/10/-	-/10/-
	6	-/10/-	-/10/-	-/10/-	-/10/-
	10	-/10/-	-/10/-	-/10/-	-/10/-
Group 7	2	3/1/1	3/1/1	-/4/1	-/4/1
	4	-/5/-	-/5/-	-/5/-	-/5/-
	6	-/5/-	-/5/-	-/5/-	-/5/-
	10	-/5/-	-/5/-	-/5/-	-/5/-
Total	220	113/99/8	113/99/8	108/102/10	106/104/10

Table 3.3 Mean running time (in seconds) for each group of instances for each model with different shelf life values (in periods). The number in parentheses indicates the number of instances where Model 1 was faster than Model 2 (and vice versa) and the number of instances where Model 1-d was faster than Model 2-d (and vice versa)

Groups	Shelf life	Model 1	Model 2	Model 1-d	Model 2-d
Group 1	2	2.48 (4)	2.57 (1)	2.45 (5)	6.76 (0)
	4	6.01 (5)	7.29 (0)	5.72 (5)	10.10 (0)
	6	7.37 (3)	8.42 (2)	7.71 (4)	11.58 (1)
	10	6.66 (4)	7.19 (1)	7.93 (3)	10.97 (2)
Group 2	2	0.30 (4)	0.40 (1)	1.02 (5)	5.19 (0)
	4	5.45 (3)	5.03 (2)	7.60 (4)	7.98 (1)
	6	7.04 (3)	5.31 (2)	9.53 (3)	8.97 (2)
	10	8.73 (3)	6.78 (2)	16.31 (4)	10.97 (1)
Group 3	2	13.54 (2)	13.46 (3)	22.95 (5)	83.47 (0)
	4	167.63 (2)	116.71 (3)	171.71 (0)	102.14 (5)
	6	195.01 (2)	161.00 (3)	171.61 (2)	127.13 (3)
	10	254.61 (1)	146.32 (4)	260.57 (1)	212.76 (4)
Group 4	2	1.64 (3)	1.57 (2)	4.45 (5)	20.31 (0)
	4	433.69 (0)	44.41 (5)	384.32 (1)	86.55 (4)
	6	331.54 (0)	221.75 (5)	831.26 (5)	83.67 (4)
	10	579.25 (1)	192.96 (4)	306.80 (1)	138.57 (4)
Group 5	2	1410.67 (2)	1398.60 (11)	1629.55 (12)	1778.71 (0)
	4	2700.04 (5)	2700.09 (0)	2700.15 (5)	2700.39 (0)
	6	2700.05 (5)	2700.11 (0)	2700.18 (5)	2700.45 (0)
	10	2700.06 (5)	2700.13 (0)	2700.18 (5)	2700.47 (0)
Group 6	2	782.74(4)	868.69 (5)	149.88 (10)	756.28 (0)
	4	3600.00 (0)	3600.00 (0)	3600.00 (0)	3600.00 (0)
	6	3600.00 (0)	3600.00 (0)	3600.00 (0)	3600.00 (0)
	10	3600.00 (0)	3600.00 (0)	3600.00 (0)	3600.00 (0)
Group 7	2	2205.24 (0)	1862.38 (4)	2880.06 (1)	2880.06 (0)
	4	3600.00 (0)	3600.00 (0)	3600.00 (0)	3600.00 (0)
	6	3600.00 (0)	3600.00 (0)	3600.00 (0)	3600.00 (0)
	10	3600.00 (0)	3600.00 (0)	3600.00 (0)	3600.00 (0)
Total		61	60	87	31

Table 3.4 Performance comparison between each two models in the groups of instances where the optimal was not achieved after 1 hour of computational time. The heading ‘MY x MZ’ indicates how many times Model Y achieved a better solution than Model Z, when a feasible solution was found

Groups	Shelf life	M1 x M2	M2 x M1	M1-d x M2-d	M2-d x M1-d
Group 5	2	5	2	5	5
	4	5	8	7	8
	6	2	12	4	11
	10	8	6	4	10
Group 6	2	1	–	1	–
	4	3	4	3	6
	6	1	9	3	5
	10	3	5	4	6
Group 7	2	1	–	1	2
	4	2	3	–	5
	6	5	–	3	2
	10	2	3	3	2
Total		38	52	38	62

3.4 Conclusion and Further Research

In this work, we consider a CLSP-L model to integrate shelf life constraints. We present two variants that include shelf life constraints and extend both to include disposal quantity and its cost. The issue of disposability occurs when raw materials have to be consumed in batch, such as milk.

Four models are tested on instances belonging to an established dataset available in the literature [15]. Regarding of the objective function value, the Model 2 (2-d) find a better feasible solution more often than the Model 1 (1-d) but takes more computational time in a higher number of instances. As the problem size grows (more products and periods, and longer shelf life), the four models taken an increasing amount of computing time to solve, passing the 1 h time limitation.

The models provide valuable insights inyo computationally demonstrating that shorter shelf lives are associated more often with model infeasibility than longer shelf lives. This chimes with an operations manager’s instinctive knowledge that a short shelf life can negatively impact on a producer’s ability to satisfy demand by limiting the amount of time that products can be kept in inventory to satisfy demand further in the future. The extent to which demand cannot be promptly satisfied can be measured by including stockouts as “negative stock” in the model [3, 4].

Further research will be focused on developing heuristics and meta-heuristic algorithms to find (near) optimal solutions for larger instances in a more reasonable time. More real-world applications, such as the effects of sales promotions and different deterioration features will be considered to extend our models in this work. In particular, a sales promotion can significantly affect the demand motivating the need to address disposal with an additional marketing plan.

Acknowledgements The authors are grateful to Aaron Scott for his support with The University of Newcastle's High-Performance Computing (HPC) cluster.

References

1. Acevedo-Ojedaa A, Contrerasa I, Chenb M (2015) Two-level lot-sizing with raw-material perishability and deterioration. Int J Product Res:1–16
2. Amorim P, Antunes CH, Almada-Lobo B (2011) Multi-objective lot-sizing and scheduling dealing with perishability issues. Industr Eng Chem Res 50(6):3371–3381. https://doi.org/10.1021/ie101645h
3. Clark AR (2003) Hybrid heuristics for planning lot setups and sizes. Computers & Industrial Engineering 45(4):545–562
4. Clark AR, Clark SJ (2000) Rolling-horizon lot-sizing when set-up times are sequence-dependent. Int J Product Res 38(10):2287–2307
5. Entrup ML, Gnther HO, Beek PV, Grunow M, Seiler T (2005) Mixed-integer linear programming approaches to shelf-life-integrated planning and scheduling in yoghurt production. Int J Product Res 43(23):5071–5100. https://doi.org/10.1080/00207540500161068
6. Haase, K.: Lotsizing and scheduling for production planning, vol. 408. Springer Science & Business Media (2012)
7. Huber J, Gossmann A, Stuckenschmidt H (2017) Cluster-based hierarchical demand forecasting for perishable goods. Expert systems with applications 76:140–151
8. Lin, C.S.K., Pfaltzgraff, L.A., Herrero-Davila, L., Mubofu, E.B., Abderrahim, S., Clark, J.H., Koutinas, A.A., Kopsahelis, N., Stamatelatou, K., Dickson, F., et al.: Food waste as a valuable resource for the production of chemicals, materials and fuels. current situation and global perspective. Energy & Environmental Science **6**(2), 426–464 (2013)
9. Pahl, J., Voβ, S.: Integrating deterioration and lifetime constraints in production and supply chain planning: A survey. European Journal of Operational Research **238**(3), 654–674 (2014). https://doi.org/10.1016/j.ejor.2014.01.060
10. Pahl, J., Voss, S., Woodruff, D.L.: Discrete lot-sizing and scheduling with sequence-dependent setup times and costs including deterioration and perishability constraints. In: 2011 44th Hawaii International Conference on System Sciences, pp. 1–10 (2011). https://doi.org/10.1109/HICSS.2011.169
11. Raak N, Symmank C, Zahn S, Aschemann-Witzel J, Rohm H (2017) Processing- and product-related causes for food waste and implications for the food supply chain. Waste Management 61:461–472. https://doi.org/10.1016/j.wasman.2016.12.027
12. Raiconi, A., Pahl, J., Gentili, M., Voβ, S., Cerulli, R.: Tactical production and lot size planning with lifetime constraints: A comparison of model formulations. Asia-Pacific Journal of Operational Research p. 1750019 (2017)
13. Romsdal A, Strandhagen JO, Dreyer HC (2014) Can differentiated production planning and control enable both responsiveness and efficiency in food production? International Journal on Food System Dynamics 5(1):34–43
14. Soman, C.A., Donk, D.P.v., Gaalman, G.J.C.: A basic period approach to the economic lot scheduling problem with shelf life considerations. International Journal of Production Research **42**(8), 1677–1689 (2004). https://doi.org/10.1080/00207540310001645165
15. Suerie C, Stadtler H (2003) The capacitated lot-sizing problem with linked lot sizes. Management Science 49(8):1039–1054
16. Tempelmeier H, Copil K (2016) Capacitated lot sizing with parallel machines, sequence-dependent setups, and a common setup operator. OR Spectrum 38(4):819–847. https://doi.org/10.1007/s00291-015-0410-2

Chapter 4
An Application of a Vessel Route Planning Model to Second the Import/Export of Seasonal Products

Riccardo Accorsi, Emilio Ferrari, Riccardo Manzini, and Alessandro Tufano

Abstract The distribution of food products represents a great trade opportunity for maritime carriers and shipping companies, especially within the Mediterranean Basin which concentrates many important food processing and consuming countries. Fresh products, as fruit and vegetables, are characterized by seasonal and climate-driven volumes, and logistics networks and distribution (i.e. shipping) operations should be designed and planned in agreement with such trends. In this working paper, an application of the *Vessel Routing Problem with Selective Pickups and Deliveries* (VRPSPD) to the maritime import/export of food seasonal product is illustrated. The VRPSPD belongs to a well-known class of vehicle routing problems intended to plan the routes of the maritime distribution of commodities between sources and destinations. A time-dependent formulation of the VRPSPD is applied in this paper to maximize the profit of a maritime carrier involved in the import/export of fruits among Mediterranean ports. A simple numerical example is used to validate the model and to identify opportunities for future problem investigations in the seaborne trade of seasonal and perishable products.

Keywords Vessel routing problem · Import/export · Seasonal products · Mediterranean routes

4.1 Introduction

As 80% of global trade by volume is carried on ships [1], the design and planning of vessels distribution networks and maritime operations reach wide interest among both scholars and logistics practitioners. The liners' and ports' shipping connectivity provide options to connect to overseas markets and to explore trade opportunities among countries and enterprises. Three are the main actors involved in maritime transportation: the shipper, the carrier or vessel company, and the supplier. The

R. Accorsi (✉) · E. Ferrari · R. Manzini · A. Tufano
Department of Industrial Engineering, Alma Mater Studiorum—University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy
e-mail: riccardo.accorsi2@unibo.it

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_4

shipper operates at the port with his own docks or warehouses and organizes the shipments, behaving as an interface between the supplier and the carrier. The supplier is indeed who provides the goods, while the carrier physically transport the freight from a port to another.

The relationship between these actors is also influenced by the chosen transport strategy. Practice suggests three widely adopted strategies: *industrial shipping*, *liner shipping* and *tramp shipping*. In the *industrial shipping,* the supplier coincides with the shipper and the carrier, owns the vessel and serves his clients trying to minimize the transportation costs from an origin to many destinations. In the *liner shipping,* the carrier provides a schedule (i.e. time table) of picking and delivery services and sells such services to the supplier through the interface of the shipper. Lastly, the *tramp shipping* is when the vessel company receives orders from the shippers and decides whether or not to serve those orders and the routes to follow alike, in order to maximize the profit with pickup and delivery services. Especially in *tramp shipping*, the adequate coordination among these subjects and a shared planning process is crucial to exploit at best the distribution capacity of the network.

In the market of food commodities and seasonal products, such trading opportunities are often and further underemphasized because of the complex management of extremely climate- and seasonal-driven volumes. These may require optimization tools supporting the planning of vessels routes able to address to import/export orders occurring at different source and destination ports over the seasons.

In such a context, a new formulation of the *Vessel Routing Problem with Selective Pickups and Deliveries* (VRPSPD) is necessary to identify the optimal sequence of ports to visit in a *tramp shipping* with the goal of maximizing the trade of seasonal products among ports and the profit of carriers. This working paper formulates a time-dependent VRPSPD to support the planning of vessel routes that respond to seasonal needs of pickup and delivery services in agreement with the import/export flows of fresh food products.

The remainder of this paper is organized as follows. Section 4.2 provides a short and focused overview of the recent literature in the field. Section 4.3 presents the problem formulation and explains sets, variables and constraints. Section 4.4 illustrates a small numerical example used to validate the model and to showcase the potential benefits of real-world and large-scale applications. Lastly, Sect. 4.5 concludes the paper with necessary topics for future research developments.

4.2 Literature Review

The literature on vessels routing problem (VRP) is ample and many formulations are provided by scholars to address real-world applications. Early, it mainly focuses on the development of minimal time ship routing analytical models involving ocean waves and weather conditions [2], while in the last decades, with the intensification of maritime global trade, papers have been intended to fleets routing, and shipping liners tours design [3]. The design of time-dependent VRP is quite novel in the field and recent formulations are provided to design time-constrained liner shipping networks and vessel routing with deadlines. Vad Karsten et al. [4] take into account coordination between a fleet of vessels and time restriction on the cargo flows with the goal of maximizing the profit of the vessel company. Their main contribution lies in the design of an efficient solving algorithm proposed for large medium and instances. Wang and Meng [5] formulate a multi-vessels routing problem with deadlines and implement a column generation-based heuristic to solve the problem in a reasonable time. Others focus on flow-driven liner shipping network design problems, where the vessels route is constrained by selective pickups and delivery services to fulfil [6]. Karlaftis et al. [7] use a genetic algorithm to solve a fleet routing problem performing selective pickups and deliveries with deadlines.

Whilst lots of work has been done on the formulation of original VRPs and on the development of efficient solving algorithms, ample opportunities remain for the application of such models in practice to address to new instances and real-world applications.

4.3 Problem Formulation

The mathematical formulation of the proposed VRPSPD is described as follows. This formulation, inspired by the recent literature [8, 9], implements the *One-to-Many-to-One* (1-M-1) which refers to the case where the origin and destination of the vessel route coincide. Indeed, as the model is aimed at maximizing the profit of the vessel company the vessel, which starts at port j*:* $a_j = 1$, does not need to visit all the ports but concludes the tour at the origin necessarily. The following notation describes the sets, the parameters, the decision variables and the problem objective function and constraints.

Sets

$i, j \in \aleph$ Set of ports.
$i = 1, \ldots, n \in \aleph$ Set of pick-up docks.
$i = n + 1, \ldots, 2n \in \aleph$ Set of delivery docks.
$(i, j) \in \mathcal{M}$ Set of feasible arcs.
$p \in \mathrm{P}$ Set of products.
$(p, i, j) \in \mathrm{O}$ Set of orders.

Parameters

S_{pij} Size of order of product p from dock i to j.
T_{ij} Time (periods) to travel on the arc *(i,j).*
$\mathrm{T}_i^{\mathrm{F}}$ Fixed time (periods) spent at port *i.*
$\mathrm{C}_{ij}^{\mathrm{V}}$ Variable costs to travel on the arc *(i,j).*
$\mathrm{C}_i^{\mathrm{F}}$ Fixed costs to visit port *i.*

v^{1}_{pi} Economic value (price) of product p at port I [€/unit].
a_i 1 if route departs from port i; 0 otherwise.
$\mathrm{t}^{\mathrm{ord}}_{pij}$ time/period the order *(p,i,j)* is released.
Q Vessel capacity.

Decision variables

x_{ij} 1 if arc the vessel depart from port i along arc *(i,j)*; 0 otherwise.
v_i 1 if port i is visited.
$\mathrm{t}^{\mathrm{out}}_i$ time/period the vessel leaves port i.
u_{pij} 1 if the order *(p,i,j)* is served.
l_{pi} Upper bound of pick-up quantity when leaving node i.
ul_{pi} Upper bound of delivered quantity when leaving node i.

Objective function

$$\max \sum_i \sum_j \sum_p \mathrm{u}_{pij} \cdot \mathrm{S}_{pij} \cdot \mathrm{v}^{1}_{\mathrm{pi}} - \sum_i \mathrm{C}^{\mathrm{F}}_i \cdot \mathrm{v}_i - \sum_{(i,j)\in \mathrm{M}} \mathrm{C}^{\mathrm{V}}_{ij} \cdot \mathrm{x}_{ijt} \tag{4.1}$$

subject to

$$\sum_{j\in\aleph} \mathrm{x}_{ij} = \mathrm{v}_i i \in \aleph \tag{4.2}$$

$$\sum_{i\in\aleph} \mathrm{x}_{ij} = \mathrm{v}_j j \in \aleph \tag{4.3}$$

$$\mathrm{t}^{\mathrm{out}}_i = 0 \quad i \in \aleph\text{: } \mathrm{a}_i = 1 \tag{4.4}$$

$$\mathrm{t}^{\mathrm{out}}_j \geq \mathrm{t}^{\mathrm{out}}_i + \mathrm{T}_{ij} - \mathrm{M} \cdot (1 - \mathrm{x}_{ij}) \;\; i, j \in \mathrm{M}, i, j \in \aleph \backslash \{ j : \mathrm{a}_j = 1 \} \tag{4.5}$$

$$\mathrm{t}^{\mathrm{out}}_j \leq \mathrm{t}^{\mathrm{out}}_i + \mathrm{T}_{ij} - \mathrm{M} \cdot (1 - \mathrm{x}_{ij}) \;\; i, j \in \mathrm{M}, i, j \in \aleph \backslash \{ j : \mathrm{a}_j = 1 \} \tag{4.6}$$

$$\mathrm{v}_j \geq \mathrm{u}_{pij}(pij) \in \mathrm{O} \tag{4.7}$$

$$\mathrm{v}_i \geq \mathrm{u}_{pij}(pij) \in \mathrm{O} \tag{4.8}$$

$$\mathrm{l}_{pi} = \sum_j \mathrm{S}_{pij} \cdot \mathrm{u}_{pij} i \in \aleph\text{: } \mathrm{a}_i = 1, p \in \mathrm{P} \tag{4.9}$$

$$\mathrm{l}_{pj} - \mathrm{l}_{pi} = \sum_k \mathrm{S}_{pjk} \cdot \mathrm{u}_{pjk} - \mathrm{M} \cdot (1 - \mathrm{x}_{ij}) i, j \in \mathrm{M}, i, j \in \aleph \backslash \{ j\text{: } \mathrm{a}_j = 1 \}, p \in \mathrm{P} \tag{4.10}$$

$$\mathrm{ul}_{pi} = 0 i \in \aleph\colon \mathrm{a}_i = 1, p \in \mathrm{P} \tag{4.11}$$

$$\mathrm{ul}_{pj} - \mathrm{ul}_{pi} = \sum_k \mathrm{S}_{pk\mathrm{j}} \cdot \mathrm{u}_{pkj} - \mathrm{M} \cdot (1 - \mathrm{x}_{ij}) i, j \in \mathrm{M}, i, j \in \aleph \backslash \{ j\colon \mathrm{a}_j = 1 \}, p \in \mathrm{P} \tag{4.12}$$

$$0 \le \sum_p \mathrm{l}_{pi} - \sum_p \mathrm{ul}_{pi} \le \mathrm{Q}\ i \in \aleph \tag{4.13}$$

$$\mathrm{t}^{\mathrm{ord}}_{pij} \cdot \mathrm{u}_{pij} \le \mathrm{t}^{\mathrm{out}}_i + \mathrm{T}_{ij} \quad (pij) \in \mathrm{O} \tag{4.14}$$

$$\mathrm{x}_{ij} \in \{0, 1\} (i, j) \in \mathrm{M} \tag{4.15}$$

$$\mathrm{v}_i \in \{0, 1\} i \in \aleph \tag{4.16}$$

$$\mathrm{u}_{pij} \in \{0, 1\} (pij) \in \mathrm{O} \tag{4.17}$$

$$\mathrm{t}^{\mathrm{out}}_i \ge 0 i \in \aleph \tag{4.18}$$

$$\mathrm{l}_{pi}, \mathrm{ul}_{pi} \ge 0 i \in \aleph\ , \ p \in \mathrm{P} \tag{4.19}$$

The objective function (4.1) maximizes the profit of the maritime carrier in term of exported freight values minus the costs of visiting a port i $\in \aleph$ and the costs of travelling along the tour. Some variation of (4.1) can be formulated to address different goals. As an example, the cost of the pickups can be also removed to the profit of the vessels company which provides a carrying service for both import and export flows. Constraints (4.2–4.3) ensure that all visited nodes have just one inbound and one outbound arc. Constraint (4.4) imposes the starting time of the tour for the first port. Constraints (4.5) and (4.6) track and control the leaving time from each served port. Constraints (4.7–4.8) state that either pick-up and delivery orders cannot be served whether that port is not visited. Constraint (4.9) sets the pick-up quantity at the first port, while Constraint (4.10) do the same for the following ports. The same is ensured for delivery quantities by Constraints (4.11) and (4.12).

While Constraint (4.13) control to not exceed the loading vessel capacity, Constraint (4.14) state that an order can be served if released before a port is visited. Lastly, Constraints (4.15–4.19) denote the nature and domain of the decision variables.

This formulation is inspired by the service flow-driven model proposed in [6] which allowed to find the optimal sequence of port visits (considering even multiple calls by a port) that maximizes the generated profit of the network. As the problem remains NP-hard, efficient solving algorithms are necessary to solve large instances, while small problems can be solved by using commercial solvers in a reasonable time, as we did in the following numerical application.

4.4 Numerical Application

The VRPSPD model is applied to a set of instances inspired to the maritime import/export flows of fruits across the Mediterranean Basin. The maritime trade across the Mediterranean Sea is growing not just as a consequence of the global shipping routes, but because of the intensification of policymaking towards a Euro-Mediterranean free trade area which yet contributes to make Europe the main market for Mediterranean fruits and vegetables [10].

The following numerical example is developed within the FUTUREMED project [11], that is aimed at supporting the design and planning of logistics and distribution channels for food supply chains across the Mediterranean countries. Optimization is hence used to study the optimal visiting tours for a tramp shipping vessel and to explore the impact of choosing potential destinations and serving different import/export orders on the maximization the carrier's profit.

We considered the seasonal demand of import and export services which varies with the port in agreement with the fruits availability and production/consumption flows. Table 4.1 and Fig. 4.1 illustrate the characteristics of the containership involved and the value of fixed time spent at each port T_i^F, respectively.

Four scenarios have been built to study the effect of variable fruits offer and availability on the seasonal tours of a containership. The four scenarios involve different ports according to the seasonal availability of apples, pears, strawberries

Table 4.1 Container ship characteristics

Fuel	Fuel cost [€/ton]	Consumption [ton/h]	Container ship	Capacity [FEUs]	Operational cost [€/day]
HSFO	260	7	Post-Panamax	1800	40,000

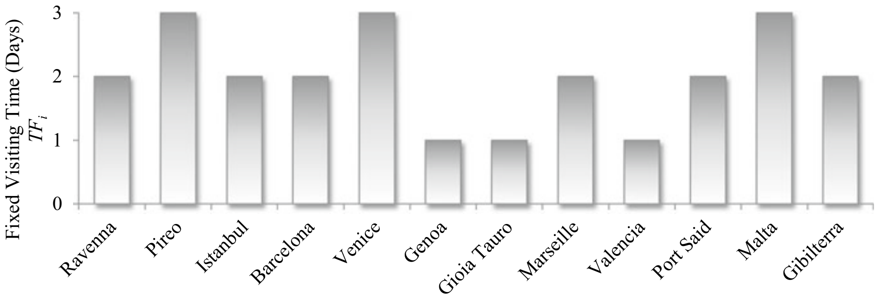


Fig. 4.1 Fixed visiting time (days) per port

Table 4.2 Cumulated export/import orders (tons/month) (Product legenda: strawberries (*s*), apples (*a*), pears (*p*), apricots (*c*))

Export/Import Seasonal Orders S_{pij} (tons/month)		Winter (February)				Spring (May)				Summer (July)			Autum (October)		
From/to Ports		Strawberries	Apples	Pears	Apricots	Strawberries	Apples	Pears	Apricots	Strawberries	Apples	Pears	Strawberries	Apples	Pears
Barcelona															
	Catania	341	3	65	2868	50	42	101	96	7	127	2794	14	28	48
	Patrasso	341	3	65	2868	50	42	101	96	7	127	2794	14	28	48
	Salerno	341	3	65	2868	50	42	101	96	7	127	2794	14	28	48
Catania															
	Barcelona	433		32	112	48			47			98	29		21
	Patrasso	433		32	112	48			47			98	29		21
	Salerno	433		32	112	48			47			98	29		21
Patrasso		165			45	35			20						
	Barcelona	55			15	12			7						
	Catania	55			15	12			7						
	Salerno	55			15	12			7						
Salerno															
	Barcelona	282	144	32	755	1730	19	9	12	16	19	170	194	137	21
	Catania	282	144	32	755	1730	19	9	12	16	19	170	194	137	21
	Patrasso	282	144	32	755	1730	19	9	12	16	19	170	194	137	21

and apricots from some Mediterranean producing/consuming regions. Values of the input parameter S_{pij} per each scenario/season and product are reported in Table 4.2 (expressed in tons per month).

The proposed model is applied to establish the optimal routing sequence that maximizes the profit of the carrier season by season. The model is run four times, one per scenario, and the solutions obtained through branch-and-bound using the solver Gurobi, run on an Intel i7 3.20 GHz with 32 GB of RAM, with a computational time of 10,800 s.

The different export and import orders result in modifying the tramp shipping service month by month. Indeed, the optimal visiting sequence varies with the season as highlighted in the solution networks of Figs. 4.2 and 4.3. The obtained results, summarized in Table 4.3, identify the distribution of the import/export orders in the Mediterranean Basin as a key decisional driver for maritime tramp carriers. In the light of this, decision support models incorporating the variability and uncertainty of the order profiles in the planning of maritime routes are strongly required.

4.5 Discussion and Future Research Developments

Even though the obtained results are not generalizable, they contribute to conclude that the routing of a tramp shipping is extremely influenced by the seasonal demand of maritime import and export services. As a consequence, the Mediterranean Sea, which concentrates many important producers and consumers of typical seasonal products as fresh fruits and vegetables, represents an important context of the application of this and similar models.

As this working paper illustrates just an application of a time-dependent VRPSPD model to design optimal vessel routes involving some Mediterranean ports, many are the potential and ongoing, research developments required.

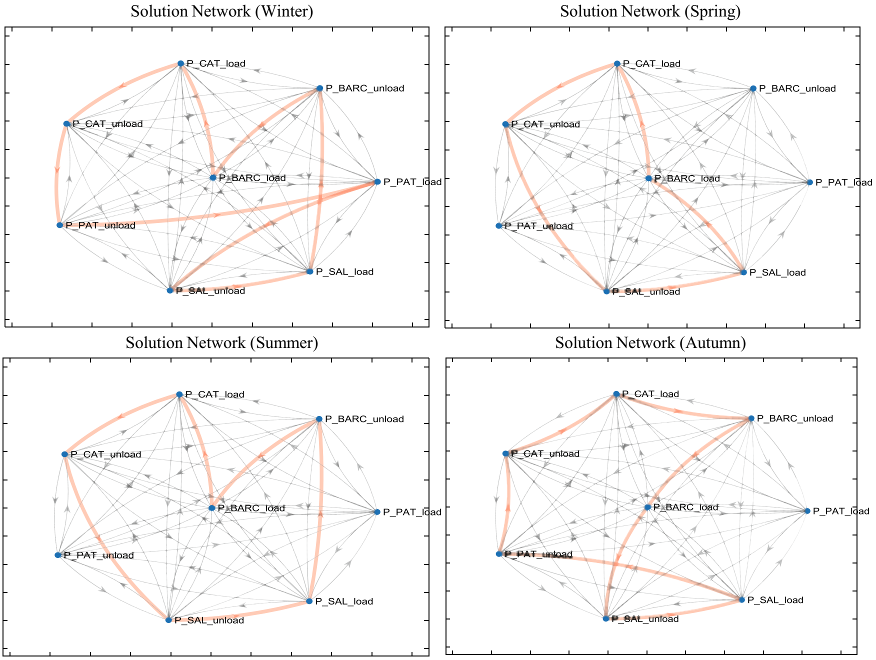


Fig. 4.2 Optimal visiting sequence: Seasonal graphs of tours

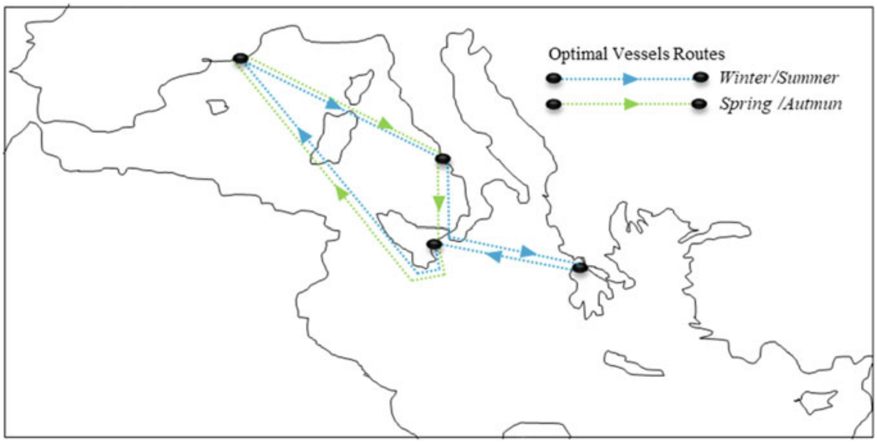


Fig. 4.3 Optimal seasonal maritime routes for the distribution of fruits across Mediterranean ports

Table 4.3 Served import/export orders

Seasons	Delivered orders	Shipped volume [tons]	Orders served [%]
Winter (February)	27	4163	100
Strawberries	12	3335	100
Apples	6	440	100
Pears	9	387	100
Spring (May)	23	12,217	64
Apricots	9	8257	75
Strawberries	8	3679	67
Apples	2	60	33
Pears	4	221	67
Summer (July)	31	10,158	94
Apricots	12	485	100
Strawberries	4	47	67
Apples	6	439	100
Pears	9	9187	100
Autumn (October)	24	1473	100
Strawberries	9	710	100
Apples	6	495	100
Pears	9	268	100

First, since we deal with fruits and vegetable products, a new release of the model will necessarily include a set of linear constraints that manage the perishability of products and their shelf life decay along shipping [12]. The shelf life of fresh food may indeed affect the shipping service since products must be unloaded and delivered before they expire. Furthermore, climate conditions may influence energy costs [13] as well as routing and shipping operations and can be considered as drivers of analysis as yet implemented in [8] and in [14] for the case of truck deliveries of food and temperature-sensitive products.

Other objective functions may refer to the benefits of other maritime trade stakeholders (e.g. suppliers, importers, regional policymakers, logistics operators), and cooperative vs. competitive approaches in the vessels routing planning studied accordingly, as typically done in the truck delivery service.

Lastly, since the development of efficient metaheuristics and solving algorithms is expected with the increasing complexity of the problem formulation and the decisional levers involved, the authors of this paper believe that future integration of a hierarchy of vessel routing and truck routing problems will be at the basis of the design and control of hyper-connected food supply chains according to the physical internet and Internet-of-things paradigms.

References

1. United Nation (UN): Review of Maritime Transport. UNCTAD/RMT/2017 (2017)
2. Bleick WE, Faulkner FD (1964) Minimal-time ship routing. J Appl Meteorol 4:217–221
3. Plum CEM, Pisinger D, Salazar-Gonzalez JJ, Sigurd MM (2014) Single liner shipping service design. Comput Oper Res 45:1–6
4. Vad Karsten C, Brouer BD, Desaulniers G, Pisinger D (2017) Time constrained liner shipping network. Transp Res Part E 105:152–162
5. Wang S, Meng Q (2014) Liner shipping network design with deadlines. Comput Oper Res 41:140–149
6. Plum CEM, Pisinger D, Sigurd MM (2014) A service flow model for the liner shipping network design problem. Eur J Oper Res 235:378–386
7. Karlaftis MG, Kepaptsoglou K, Sambracos E (2009) Containership routing with time deadlines and simultaneous deliveries and pick-ups. Transp Res Part E 45:210–221
8. Kepaptsoglou K, Karlaftis MG, Fountas G (2015) Weather impact on containership routing in closed seas: a chance-constraint optimization approach. Transp Res Part C 55:139–155
9. Cuesta FE, Andersson H, Fagerholt K, Laporte G (2017) Vessel routing with pickups and deliveries: an application to the supply of offshore oil platforms. Comput Oper Res 79:140–147
10. Emlinger C, Chevassus Lozza E, Jacquet F (2006) EU market access for mediterranean fruit and vegetables: a gravity model assessment. In: Proceeding of the 98th EAAE seminar: Chania: marketing dynamics within the global trading system: new perspectives. 29 June – 2 July, 1–14
11. Accorsi R, Cholette S, Manzini R, Tufano A (2018) A hierarchical data architecture for sustainable food supply chain management and planning. J Clean Prod 203:1039–1054
12. Rong A, Akkerman R, Grunow M (2011) An optimization approach for managing fresh food quality throughout the supply chain. Int J Prod Econ 131:421–429
13. Gallo A, Accorsi R, Baruffaldi G, Manzini R (2017) Designing sustainable cold chains for long-range food distribution: energy-effective corridors on the Silk Road belt. Sustainability 9(11):2044
14. Accorsi R, Gallo A, Manzini R (2017) A climate-driven decision-support model for the distribution of perishable products. J Clean Prod 165:917–929

Part II
Transport and Logistics

Chapter 5
A Branch-and-Price Framework for the Maximum Covering and Patrol Routing Problem

Paul A. Chircop, Timothy J. Surendonk, Menkes H. L. van den Briel, and Toby Walsh

Abstract The Maximum covering and patrol routing problem (MCPRP) is concerned with the allocation of police patrol cars to accident hotspots on a highway network. A hotspot is represented as a time window at a precise location on the network at which motor vehicle accidents have a high probability of occurring. The nature of these accidents may be due to speeding, driver fatigue or blind-spots at intersections. The presence of police units at hotspots serves as an accident prevention strategy. In many practical applications, the number of available cars cannot cover all of the hotspots on the network. Hence, given a fleet of available units, an optimization problem can be designed which seeks to maximize the amount hotspot coverage. The cars must be routed in such a way as to avoid multiple contributions of the patrol effort to the same hotspot. Each police car is active over a predefined shift, beginning and ending the shift at a fleet station. In this paper, we introduce a method for constructing a time-space network of the MCPRP which is suitable for the application of a branch-and-price solution approach. We propose some large-scale test problems and compare our approach to a state-of-the-art Minimum cost network flow problem (MCNFP) model. We show that our branch-and-price approach can outperform the MCNFP model on selected large-scale networks for small to medium fleet sizes. We also identify problems which are too large for the MCNFP model to solve, but which can be easily handled by our approach.

P. A. Chircop (✉) · T. J. Surendonk
Defence Science and Technology Group, Sydney, NSW, Australia
e-mail: paul.chircop@dst.defence.gov.au

T. J. Surendonk
e-mail: timothy.surendonk@dst.defence.gov.au

M. H. L. van den Briel
Hivery, Sydney, NSW, Australia
e-mail: menkes@hivery.com

T. Walsh
UNSW Sydney, Data61, Kensington, NSW, Australia
e-mail: toby.walsh@data61.csiro.au

TU Berlin, Berlin, Germany

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_5

Keywords Route planning · Surveillance scheduling · Branch-and-price

5.1 Introduction and Background

The Maximum covering and patrol routing problem (MCPRP) was first studied by Keskin et al. [10]. Given a set of highway locations and time intervals at which traffic accidents have a high probability of occurring, the problem is to find patrol routes for a set of police cars so that the aggregate coverage of all the accident hotspots is maximized. Each patrol car begins and ends its route at a fleet station on a predefined shift. Keskin et al. [10] modelled the MCPRP using a Mixed-integer programming (MIP) formulation and found that state-of-the-art commercial solvers were not always able to find good quality solutions. Hence, a number of heuristic techniques (local and tabu search) were introduced and benchmarked on a range of test problem instances. The test problems solved by Keskin et al. [10] were generated with randomized and real-world data with up to 40 hotspots and 8 patrol cars. These test problem instances were created to reflect the circumstances faced by (and the resources available to) law enforcement agencies in a particular region of the United States.[1] The paper by Keskin et al. [10] reports that the heuristic techniques were able to produce good quality but not optimal solutions to the larger problem instances with 40 hotspots.[2]

The paper published by Çapar et al. [3] showed that significant improvements could be made to the MIP formulation of [10]. With information on the structure of candidate routes in an optimal solution, the authors demonstrated that the number of variables in the formulation of [10] can be reduced. The authors also incorporated a number of bounds constraints which provide additional strength to their reformulation. The enhanced efficiency given through this reformulation of the MCPRP was demonstrated on the benchmark instances introduced in the original paper by Keskin et al. [10].[3]

More recent work by Dewil et al. [7] has shown that the MCPRP can be modelled as a Minimum cost network flow problem (MCNFP).[4] The MCNFP is solvable in

[1] The literature review conducted by Keskin et al. [10] notes that the MCPRP bears similarities to the Team orienteering problem with time windows (TOPTW). However, the distinguishing characteristic of the MCPRP is that the profit associated with visiting a hotspot is not fixed, but is rather a function of the amount of "dwell time" within that hotspot's time window. The authors state that the range of time window lengths used in the problems of their study varied from $1-270$ minutes (usually assuming an 8 hour shift).

[2] The results of this study can also be found in the PhD thesis by Li [12].

[3] Çapar et al. [3] considered a number of extensions to the standard MCPRP paradigm. These extensions included the incorporation of shift breaks and allowing the patrol vehicles to begin the shift at different locations, possibly with delayed starting times.

[4] The MCNFP possesses the integrality property when the arc capacities are integer (see Ahuja et al. [1]). This means that the optimal solution is naturally integer if the problem is solved as a linear program.

polynomial time, and thus, the authors correct the claim by Keskin et al. [10] that the MCPRP is NP-hard. The study sets out a time-space network formulation of the problem on which an MCNFP model is defined. The network formulation divides individual hotspots into time sections or segments, which are constructed by considering possible transitions of vehicles which depart from the end of a hotspot (and arrive at another hotspot) or arrive at the beginning of a hotspot (having departed from another hotspot). The MCNFP paradigm also permits the time sections to be weighted differently, thus constituting an extension of the standard MCPRP. The authors demonstrate the superiority of their approach by comparing their computational results with those of Keskin et al. [10]. The MCNFP model is extended by the authors to a Multi-commodity minimum cost network flow problem (MCMCNFP) model which aims to handle overlapping shifts and different start/end locations for the patrol vehicles. In order to test the scalability of the model, the authors state that they could solve a 100 hotspot instance to optimality with up to 23 patrol cars. However, the authors also report that they could not run a 500 hotspot problem instance, even with 3 patrol cars.

Given the limitations on the MCNFP model to solve large-scale problem instances of the MCPRP (as reported by Dewil et al. [7]), our paper aims to investigate the applicability and feasibility of a branch-and-price (column generation with branch-and-bound) approach to the problem. Given that similar approaches have recently proved to be effective at solving closely related patrol routing and scheduling problems, a branch-and-price approach constitutes a natural and promising candidate for solving large-scale instances of the MCPRP.[5]

We begin our study by outlining a process for the construction of a time-space network, which provides an appropriate modelling framework for a path-based linear programming formulation of the MCPRP. A column generation master problem, reduced costs, subproblem, seed column construction and pricing strategies are then subsequently outlined. We then propose a simple branch-and-price paradigm to obtain integer solutions through the incorporation of branching cuts to the master problem. The paper concludes with a presentation and discussion of a number of computational tests performed on a range of benchmark problems, and the results are compared with the MCNFP model of Dewil et al. [7].

5.2 Preliminary Notation

The patrol operations network is a directed graph $G_S = (V_S, A_S)$, where $V_S = \{0\} \cup \{1, \dots, n\}$ is the set of geographical locations and $A_S \subseteq \{(i, j) \mid i, j \in V_S,\ i \neq j\}$ is the set of feasible transitions between the elements of V_S. The singleton set $\{0\}$ is used to denote the fleet station, whereas the set $\{1, \dots, n\}$ represents the number of distinct locations at which hotspots may be found. For each $i \in V_S \backslash \{0\}$ there is a set

[5] For example, see previous work on the Patrol boat scheduling problem with complete coverage (PBSPCC) by the authors of this paper [5] or the PhD thesis by Chircop [4].

of non-overlapping hotspots, where each hotspot is represented by a time window with a start time and an end time. The number of hotspots at location i is given by h_i, and the mth hotspot at location i is denoted by $(i, [e^i_m, l^i_m])$, where $e^i_m < l^i_m$ for all $m \in \{1, \dots, h_i\}$ and for all $i \in V_S \backslash \{0\}$. Without loss of generality, at any given location $i \in V_S \backslash \{0\}$, if $m' < m''$, then $l^i_{m'} \le e^i_{m''}$, where $m', m'' \in \{1, \dots, h_i\}$. The set of all hotspots is given by W and is indexed by ℓ. The opening (start time) of hotspot ℓ is denoted by $\min(\ell)$ while the close (finish time) of the hotspot is denoted by $\max(\ell)$. The set of hotspots can be expressed as $W := \bigcup_{i \in V_S \backslash \{0\}} W(i)$, where, $W(i)$ is the set of time windows $\{(i, [e^i_1, l^i_1]), \dots, (i, [e^i_{h_i}, l^i_{h_i}])\}$ at location i. We also define a function ω which maps hotspots to their geographical locations: $\omega : W \to V_S \backslash \{0\}$.

5.3 Network Construction

Given $G_S = (V_S, A_S)$, the set of hotspots W and a shift duration T, we can construct a time-space network $G_R = (V_R, A_R)$ for the MCPRP according to a transformation $(G_S, W, T) \mapsto G_R$. On this expanded time-space network, we have a set of patrol arcs $A_{\mathrm{P}} \subset A_R$, a set of waiting arcs $A_{\mathrm{W}} \subset A_R$, and a set of transit arcs $A_{\mathrm{T}} \subset A_R$. The set of patrol arcs in time window $\ell \in W$ is expressed as $A_{\mathrm{P}}(\ell) \subseteq A_{\mathrm{P}}$. We define $t_{uv} \in \mathbb{Z}^+$ to be the transit time of traversing arc $(u, v) \in A_R$. For each $v \in V_R$, let $A_+(v)$ be the set of all arcs emanating from v, and let $A_-(v)$ be the set of all arcs terminating at v. The source and sink vertices (representing the fleet station) are s and τ, respectively. Note that $A_-(s) = A_+(\tau) = \varnothing$. Equipped with the preceding definitions and notation, the time-space network construction begins with an initialization procedure which creates a source and a sink vertex, along with a layer of vertices for each spatial location $V_S \backslash \{0\}$. Each layer will initially contain $T + 1$ vertices, where the horizontal spacing between the vertices defines the time discretization. Hence, each vertex $u \in V_R$ can be expressed in terms of a location-time pair (i, t), where $i \in V_S$ and $t \in \{0, \dots, T\}$. The initialization procedure can be found in Algorithm 1.

Algorithm 1 MCPRP: Initialization of a time-space network

```
Input: A spatial network $G_S = (V_S, A_S)$ and a shift length $T$
procedure InitializeTimeSpaceNetwork($G_S, T$)
    $V_R, A_R \leftarrow \varnothing$
    Create source vertex $s = (0, 0)$ and sink vertex $\tau = (0, T)$
    $V_R \leftarrow V_R \cup \{s, \tau\}$
    for $i \in V_S \backslash \{0\}$ do
        for $t = 0, \dots, T$ do
            Create vertex $u$ with $u = (i, t)$
            $V_R \leftarrow V_R \cup \{u\}$
    return $G_R = (V_R, A_R)$
```

Once the initialization procedure has been executed, the next step is to define the hotspots on the time-space network. The hotspots for each location are represented by a series of patrol arcs. The design choice for the hotspots is based on the following insightful theorem from Keskin et al. [10], which is stated below.

Theorem 5.1 (Keskin et al. [10]) *Let K^* be an optimal solution to an instance of the MCPRP. For each hotspot $\ell \in W$ visited by a patrol vehicle $k \in K^*$, the time at which k finishes patrolling ℓ is* $\min\left\{\max(\ell), T - t_{\{\omega(\ell)\}0}\right\}$ *if ℓ is the last hotspot visited on k's route, and* $\max(\ell)$ *otherwise.*

Theorem 5.1 states that in an optimal solution to the MCPRP, a given patrol vehicle will remain at hotspot ℓ until the close of the time window if hotspot ℓ is not the last hotspot on k's route. If, on the other hand, the hotspot ℓ is the last hotspot visited on patrol vehicle k's route, then k remains at hotspot ℓ until the close of the time window or until the latest time at which k can leave the hotspot and arrive back at the fleet station within the shift T. Given this result, we can represent each hotspot with a series of patrol arcs which collectively terminate at the vertex corresponding to the end of the time window or at the vertex corresponding to the latest possible time at which a patrol vehicle must return to the fleet station. The first patrol arc in the series emanates from the vertex corresponding to the start of the time window, with the subsequent patrol arc in the series originating at the next chronological vertex within the time window, and so on. This process is illustrated in Fig. 5.1. The formal procedure for constructing the patrol arcs over the hotspots on a time-space network can be found in Algorithm 2. This procedure also includes a test to check whether a time window lies within, crosses the boundary of or lies outside of the feasibility interval $[t_{0i}, T - t_{i0}]$. Any hotspot which crosses the boundary of the feasibility interval must have its start and end time updated accordingly, while any hotspot lying entirely outside the feasibility window should be discarded.

As we will see in a later section, a set of packing constraints is required in the column generation master problem to avoid multiple contributions to the patrol effort in each hotspot. By adopting the patrol arc construction shown in Fig. 5.1, only one

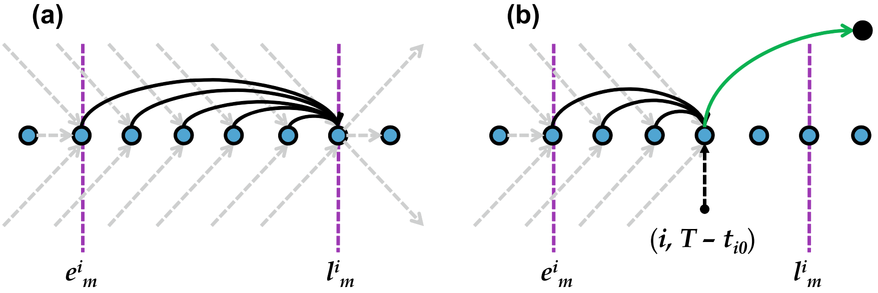


Fig. 5.1 Part **a** corresponds to a hotspot which lies entirely within the timespan $[t_{0i}, T - t_{i0}]$. Part **b** is indicative of a time window which has a closing time greater than $T - t_{i0}$. The green arc in part **b** is the transit arc which traces back to the fleet station

Algorithm 2 MCPRP: Patrol arc construction for hotspots

```
**Input**: A spatial network $G_S = (V_S, A_S)$, a shift length $T$, a set of hotspots $W$
**procedure** ConstructPatrolArcs($G_S$, $W$, $T$)
    $G_R \leftarrow$ InitializeTimeSpaceNetwork($G_S$, $T$)
    $A_P \leftarrow \varnothing$
    **for** $\ell \in W$ **do**
        $A_P(\ell) \leftarrow \varnothing$
        **if** $\big(\max(\ell) < t_{0\{\omega(\ell)\}}\big) \bigvee \big(\min(\ell) > T - t_{\{\omega(\ell)\}0}\big)$ **then**
            **continue**
        **else**
            **if** $\min(\ell) < t_{0\{\omega(\ell)\}}$ **then**
                $\min(\ell) \leftarrow t_{0\{\omega(\ell)\}}$
            **if** $\max(\ell) > T - t_{\{\omega(\ell)\}0}$ **then**
                $\max(\ell) \leftarrow T - t_{\{\omega(\ell)\}0}$
            **for** $t = \min(\ell), \ldots, \max(\ell) - 1$ **do**
                Create arc $(u, v)$ with $u = (\omega(\ell), t)$ and $v = (\omega(\ell), \max(\ell))$
                $A_P \leftarrow A_P \cup \{(u, v)\}$
                $A_P(\ell) \leftarrow A_P(\ell) \cup \{(u, v)\}$
    $A_R \leftarrow A_R \cup A_P$
    **return** $G_R = (V_R, A_R)$
```

packing constraint is required per hotspot. Without the insight of Theorem 5.1, a naive alternative would be to construct a patrol arc for each time interval in each hotspot, with a corresponding packing constraint for the patrol arcs in the master problem. Such a naive construction would increase the runtime for the solution of both the master problem and subproblem.

Once the hotspots have been identified and constructed on the time-space network, the next step is to create transit arcs between the source vertex and each location. In addition, another set of transit arcs is required to connect each geographical location with the sink vertex. Following the construction of these transit arcs, a set of waiting arcs is required for each layer of vertices in the time-space network. Waiting arcs correspond to dead time, where a patrol vehicle is stationed at a geographical location but is not actively contributing to the patrol effort. First, waiting arcs are constructed between the end of each time window and the start of all subsequent time windows at the same location. Second, a set of waiting arcs is created to connect a location's arrival vertex with the start of each time window at that location. Finally, waiting arcs are constructed which connect the end of each time window at a given location with that location's termination vertex.[6] The procedure is fully described in Algorithm 3.

Following the construction of the primary transit and waiting arcs, we need to account for secondary transit and waiting arcs which correspond to potential move-

[6] The arrival vertex at location $i \in V_S$ is the vertex $v \in V_R$ such that $v = (i, t_{0i})$. The termination vertex at location $i \in V_S$ is the vertex $u \in V_R$ such that $u = (i, T - t_{i0})$.

Algorithm 3 MCPRP: Primary transit and waiting arc construction

```
Input: A spatial network $G_S = (V_S, A_S)$, a shift length $T$, a set of hotspots $W$
procedure CONSTRUCTTRANSITWAITINGARCS($G_S, W, T$)
    $G_R \leftarrow$ CONSTRUCTPATROLARCS($G_S, W, T$)
    $A_\mathrm{T}, A_\mathrm{W} \leftarrow \varnothing$
    for $i \in V_S \backslash \{0\}$ do
        Create arcs $(s, v)$ and $(u, \tau)$ with $v = (i, t_{0i})$ and $u = (i, T - t_{i0})$
        $A_\mathrm{T} \leftarrow A_\mathrm{T} \cup \{(s, v), (u, \tau)\}$
        for $\ell \in W(i)$ do
            Create arcs $(v, w)$ and $(x, u)$ with $w = (i, \min(\ell))$ and $x = (i, \max(\ell))$
            $A_\mathrm{W} \leftarrow A_\mathrm{W} \cup \{(v, w), (x, u)\}$
            for $\ell' \in W(i)$ do
                if $\min(\ell') > \max(\ell)$ then
                    Create arc $(y, z)$ with $y = (i, \max(\ell))$ and $z = (i, \min(\ell'))$
                    $A_\mathrm{W} \leftarrow A_\mathrm{W} \cup \{(y, z)\}$
    $A_R \leftarrow A_R \cup A_\mathrm{T} \cup A_\mathrm{W}$
    return $G_R = (V_R, A_R)$
```

Algorithm 4 MCPRP: Secondary transit and waiting arc construction

```
Input: A spatial network $G_S = (V_S, A_S)$, a shift length $T$, a set of hotspots $W$
procedure CONNECTLOCATIONS($G_S, W, T$)
    $G_R \leftarrow$ CONSTRUCTTRANSITWAITINGARCS($G_S, W, T$)
    for $i \in V_S \backslash \{0\}$ do
        for $\ell \in W(i)$ do
            for $j \in V_S \backslash \{0, i\}$ such that $b_{ij} = 1$ do
                if $\max(\ell) + t_{ij} \le T - t_{j0}$ then
                    Create arc $(u, v)$ with $u = (i, \max(\ell))$ and $v = (j, \max(\ell) + t_{ij})$
                    $A_\mathrm{T} \leftarrow A_\mathrm{T} \cup \{(u, v)\}$
                    if $\exists\, \ell' \in W(j)$ such that $\min(\ell') \le \max(\ell) + t_{ij} \le \max(\ell')$ then
                        continue
                    else
                        for $\ell' \in W(j)$ do
                            if $\min(\ell') > \max(\ell) + t_{ij}$ then
                                Create arc $(v, w)$ with $w = (j, \min(\ell'))$
                                $A_\mathrm{W} \leftarrow A_\mathrm{W} \cup \{(v, w)\}$
    $A_R \leftarrow A_R \cup A_\mathrm{T} \cup A_\mathrm{W}$
    return $G_R = (V_R, A_R)$
```

ments of patrol vehicles between different geographical locations. For the end of each time window $\max(\ell)$ at a given geographical location $i \in V_S \backslash \{0\}$, a set of transit arcs is created to link location i with all other locations $j \in V_S \backslash \{0, i\}$ for which a feasible transit lane exists, that is, for all j such that $b_{ij} = 1$ and $\max(\ell) + t_{ij} \le T - t_{j0}$,

Algorithm 5 MCPRP: Post processing of the time-space network

```
Input: A spatial network $G_S = (V_S, A_S)$, a shift length $T$, a set of hotspots $W$
procedure PostProcessing($G_S, W, T$)
    $G_R \leftarrow$ ConnectLocations($G_S, W, T$)
    $V_{\text{temp}}, A_{\text{temp}} \leftarrow \varnothing$
    for $(u, v) \in A_{\mathrm{P}}$ do
        if $(A_-(u) = \varnothing) \wedge (u \neq s)$ then
            $A_{\text{temp}} \leftarrow A_{\text{temp}} \cup \{(u, v)\}$
        else
            continue
    for $u \in V_R \backslash \{s, \tau\}$ do
        if $A_-(u) = \varnothing$ then
            $V_{\text{temp}} \leftarrow V_{\text{temp}} \cup \{u\}$
        else
            continue
    $V_R \leftarrow V_R \backslash V_{\text{temp}}, A_R \leftarrow A_R \backslash A_{\text{temp}}, A_{\mathrm{P}} \leftarrow A_{\mathrm{P}} \backslash A_{\text{temp}}$
    return $G_R = (V_R, A_R)$
```

where $(b_{ij})_{i,j=1,\ldots,n}$ is the adjacency matrix of the spatial network. For each feasible transit arc constructed between $u = (i, \max(\ell))$ and $v = (j, \max(\ell) + t_{ij})$, we need to check if there exists an $\ell' \in W(j)$ such that $\min(\ell') \leq \max(\ell) + t_{ij} \leq \max(\ell')$. If this condition is satisfied, then the connecting transit arc between i and j hits a hotspot at location j, and no further work is required. However, if the condition is not satisfied, then a series of waiting arcs are required to connect node $v = (j, \max(\ell) + t_{ij})$ with each node $w = (j, \min(\ell'))$ such that $\ell' \in W(j)$ and $\min(\ell') > \max(\ell) + t_{ij}$. The procedure to construct the secondary transit and waiting arcs is formally outlined in Algorithm 4.

The final stage of the time-space network construction is a post-processing phase. At this final stage, any vertices in the time-space network which do not contain any incoming arcs are deleted. In other words, a vertex $u \in V_R \backslash \{s, \tau\}$ will be deleted if $A_-(u) = \varnothing$. In addition, all arcs $(u, v) \in A_+(u)$ which proceed from such a vertex must be deleted from the network. These arcs, if they exist, will be patrol arcs, since all waiting and transit arcs are constructed from vertices with non-empty in-arc sets. The post-processing phase is illustrated in Fig. 5.2 and formally summarized in pseudocode in Algorithm 5.

5.4 Master Problem

The time-space network $G_R = (V_R, A_R)$ constructed according to the procedures of the previous section can be used to formulate the MCPRP as a linear program, suitable for the application of column generation. Let P be the set of all feasible paths through

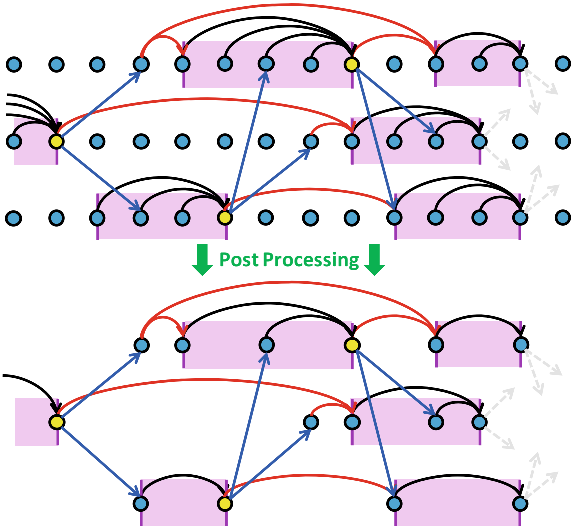


Fig. 5.2 (TOP) Construction of secondary transit and waiting arcs on the time-space network. The yellow vertices correspond to the end times of hotspots from which transit (blue) arcs are constructed. If a transit arc emanating from a yellow vertex does not land within a hotspot at another location, waiting (red) arcs are constructed from the target vertex to the opening of all subsequent hotspots. Additional waiting arcs are constructed between the closing of a hotspot and the opening of the following hotspot at the same geographical location. (BOTTOM) The time-space network after the post-processing phase (deletion of superfluous arcs and vertices). A vertex, with or without an outgoing patrol (black) arc, is deleted if there are no arcs pointing to it

the network G_R from s to τ. For each $p \in P$, let $x_{uvp} = 1$ if path p uses arc $(u, v) \in A_R$ and $x_{uvp} = 0$ otherwise. As the time-space network G_R contains no cycles, we can express the integral flow $x_{uv} \in \mathbb{Z}^+$ over an arc $(u, v) \in A_R$ in terms of path variables $\lambda_p \in \{0, 1\}$ for all $p \in P$, where $\lambda_p = 1$ if path p is used and $\lambda_p = 0$ otherwise. Hence, we can write $x_{uv} = \sum_{p \in P} x_{uvp}\lambda_p$. Denote $\bar{c}_p = \sum_{(u,v) \in A_\mathrm{P}} t_{uv} x_{uvp}$ to be the total time spent on patrol for path $p \in P$, and $\bar{c}_{pi} = \sum_{\ell_i \in W} \sum_{(u,v) \in A_\mathrm{P}(\ell_i)} t_{uv} x_{uvp}$ to be the time path $p \in P$ spends on patrol in location $i \in V_S \backslash \{0\}$, where $\ell_i \in \{\ell \in W \mid \omega(\ell) = i\}$. Moreover, let $a_{\ell p} = \sum_{(u,v) \in A_\mathrm{P}(\ell)} x_{uvp}$ for all $p \in P$ and $\ell \in W$, so that $a_{\ell p} = 1$ if $p \in P$ patrols hotspot $\ell \in W$ and $a_{\ell p} = 0$ otherwise. By relaxing the integrality constraints on the path variables, that is, setting $\lambda_p \geq 0$ for all $p \in P$, we can formulate a master linear programming problem for the MCPRP as follows:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{p \in P} \bar{c}_p \lambda_p, && && && (5.1)\\
\text{subject to} \quad & \sum_{p \in P} a_{\ell p} \lambda_p \leq 1, && \forall \ell \in W, && [\pi_\ell] && (5.2)\\
& \sum_{p \in P} \bar{c}_p \lambda_p \leq \bar{C}, && && [\alpha] && (5.3)\\
& \sum_{p \in P} \bar{c}_{pi} \lambda_p \leq \bar{C}_i, && \forall i \in V_S \backslash \{0\}, && [\beta_i] && (5.4)\\
& \sum_{p \in P} \lambda_p \leq \kappa_{\max}, && && [\gamma] && (5.5)\\
& \lambda_p \geq 0, && \forall p \in P. && && (5.6)
\end{aligned}
$$

The objective function (5.1) seeks to maximize the aggregate time spent on hotspot patrol. The statement of the objective is followed by a series of packing constraints given through (5.2). There is a single packing constraint for each hotspot $\ell \in W$. The packing constraints are included to ensure that each hotspot is patrolled by at most one vehicle, thereby prohibiting multiple contributions to a single hotspot. The constraints (5.3) and (5.4) are optional bounds constraints for the formulation. Constraint (5.3) stipulates an upper bound $\bar{C}$ on the aggregate patrol time delivered by the vehicles across all hotspots on the network. The constraints given by (5.4) use a set of values $\{\bar{C}_i \mid i \in V_S \backslash \{0\}\}$ to enforce upper bounds on the aggregate patrol effort delivered to all the hotspots at each geographical location. The upper bounds C and $\bar{C}_i$ are given through (5.7) below. Constraint (5.5) provides an upper bound on the number of available patrol vehicles through $\kappa_{\max}$. Finally, the constraints (5.6) enforce non-negativity conditions on the flow of each vehicle over each arc in the time-space network.

$$
\bar{C} := \sum_{i=1}^{n} \sum_{m=1}^{h_i} \left(l_m^i - e_m^i\right), \qquad \bar{C}_i := \sum_{m=1}^{h_i} \left(l_m^i - e_m^i\right), \quad \forall i \in V_S \backslash \{0\}. \qquad (5.7)
$$

If the set P of feasible paths through the time-space network is large, then it may not be practicable to write out the full master problem (5.1)–(5.6). In such cases, the problem can be initialized with a subset of paths $P' \subset P$ to form an initial Restricted master problem (RMP). Columns representing paths of negative reduced cost can then be added to the RMP in an iterative fashion by feeding the dual variables[7] of the RMP to a column generation subproblem.[8] A candidate initialization procedure for the RMP of the MCPRP is proffered in Sect. 5.6, while the nature of the column generation subproblem is discussed in Sect. 5.5.

5.5 Reduced Costs and Subproblem

Paths of negative reduced cost are determined by solving a pricing subproblem over the underlying time-space network G_R. Uncovering the algebraic form of the reduced cost of a path $p \in P$ naturally leads to revealing the structure of the pricing subproblem. First, we can construct some useful surjective mappings in order to define the reduced cost of a path: $\phi : A_{\text{P}} \rightarrow W'$ and $\psi : A_{\text{P}} \rightarrow V_S \backslash \{0\}$. The set $W' = \{1, \ldots, |W|\}$ is an integer-valued index set corresponding to the hotspots on the network. Thus, for each patrol arc, ϕ maps to the hotspot index, while ψ maps to the geographical location. The reduced cost $\bar{r}_p$ of a path $p \in P$ through the time-space network G_R is given through $\bar{r}_p := \boldsymbol{v}^T \mathbf{A}_p - \bar{c}_p$, where $\boldsymbol{v}^T$ is a row vector of the dual variables of the master problem (5.1)–(5.6), $\mathbf{A}_p$ is the column corresponding to variable λ_p, and $\bar{c}_p$ is the cost coefficient of λ_p in the objective function (5.1). Therefore, the reduced cost $\bar{r}_p$ of a path $p \in P$ through the time-space network G_R can be written in terms of the underlying arc variables as follows:

$$\begin{aligned}
\bar{r}_p &= \boldsymbol{v}^T \mathbf{A}_p - \bar{c}_p, && (5.8)\\
&= \left(\sum_{\ell \in W} \pi_\ell a_{\ell p} \right) + \alpha \bar{c}_p + \beta_i \bar{c}_{pi} + \gamma - \bar{c}_p, && (5.9)\\
&= \left(\sum_{\ell \in W} \pi_\ell \sum_{(u,v) \in A_{\text{P}}(\ell)} x_{uvp} \right) + \alpha \left(\sum_{(u,v) \in A_{\text{P}}} t_{uv} x_{uvp} \right) + \left(\sum_{(u,v) \in A_{\text{P}}} \beta_{\psi(u,v)} t_{uv} x_{uvp} \right) && (5.10)\\
&\quad + \gamma \left(\sum_{(s,v) \in A_+(s)} x_{svp} \right) - \left(\sum_{(u,v) \in A_{\text{P}}} t_{uv} x_{uvp} \right),\\
&= \left(\sum_{(u,v) \in A_{\text{P}}} \left[\pi_{\phi(u,v)} + \left(\alpha + \beta_{\psi(u,v)} - 1 \right) t_{uv} \right] x_{uvp} \right) + \left(\sum_{(s,v) \in A_+(s)} \gamma \, x_{svp} \right). && (5.11)
\end{aligned}$$

[7] The dual variables of the RMP can be found in square parentheses along the right-hand side of (5.1)–(5.6).

[8] This essentially describes the column generation technique for solving prohibitively large linear programs (see [6] for a comprehensive introduction). The fundamental insight of the column generation approach is to generate the columns of the constraint matrix on-the-fly by recourse to an optimization subproblem. The idea was originally suggested by Ford and Fulkerson [8], but was first implemented by Gilmore and Gomory [9].

Hence, the reduced cost of path $p \in P$ can be expressed as $\bar{r}_p = \sum_{(u,v)\in p} \mu_{uv} x_{uv}$, where the coefficients μ_{uv} are given through: $\mu_{uv} = \pi_{\phi(u,v)} + (\alpha + \beta_{\psi(u,v)} - 1)t_{uv}$ if $(u, v) \in A_\text{P}$, $\mu_{uv} = \gamma$ if $(u, v) \in A_+(s)$ and $\mu_{uv} = 0$ otherwise. Therefore, the pricing subproblem can be written as a shortest path problem over G_R as follows:

$$\text{minimize} \quad \sum_{(u,v)\in A_R} \mu_{uv} x_{uv}, \tag{5.12}$$

$$\text{subject to} \quad \sum_{(s,v)\in A_+(s)} x_{sv} = 1, \tag{5.13}$$

$$\sum_{(u,v)\in A_-(v)} x_{uv} = \sum_{(v,w)\in A_+(v)} x_{vw}, \quad \forall v \in V_R \backslash \{s, \tau\}, \tag{5.14}$$

$$\sum_{(u,\tau)\in A_-(\tau)} x_{u\tau} = 1, \tag{5.15}$$

$$x_{uv} \in \{0, 1\}, \quad \forall (u, v) \in A_R. \tag{5.16}$$

Given that G_R is a directed acyclic graph, the shortest path problem given through (5.12)–(5.16) can be solved by first applying the dual costs from the RMP and then performing edge relaxation over a topologically sorted list of the vertices in G_R. This procedure is given through DAG-SP(G_R, μ, s, τ), which outputs a shortest path p through G_R from s to τ assuming the cost structure μ and the associated path cost $\delta_p(s, \tau)$. The entire shortest path procedure is summarized in Algorithm 6.

Algorithm 6 MCPRP: Dual based shortest path through a time-space network

```
Input: A time-space network $G_R = (V_R, A_R)$ with source $s \in V_R$ and sink $\tau \in V_R$, vector of dual variables $\boldsymbol{v}$
procedure MCPRP_DualShortestPath($G_R, s, \tau, \boldsymbol{v}$)
    for $(u, v) \in A_\text{P}$ do
        $\mu_{uv} \leftarrow \pi_{\phi(u,v)} + \big(\alpha + \beta_{\psi(u,v)} - 1\big) t_{uv}$
    for $(u, v) \in A_+(s)$ do
        $\mu_{uv} \leftarrow \gamma$
    for $(u, v) \in A_R \backslash (A_\text{P} \cup A_+(s))$ do
        $\mu_{uv} \leftarrow 0$
    $(p, \delta_p(s, \tau)) \leftarrow$ DAG- SP($G_R, \mu, s, \tau$)
    $\bar{r}_p^* \leftarrow \delta_p(s, \tau)$
    return $(p, \bar{r}_p^*)$
```

5.6 Seed Columns

The path-based linear programming formulation of the MCPRP is of set packing type, and thus, the problem of constructing a feasible initial primal basis is not onerous (since the patrol coverage constraints are non-binding). We have chosen to adopt a Randomized shortest path heuristic (RSPH) to generate a set of candidate paths for the initialization of the RMP. The random cost structure derived from the RSPH is intended to produce an initial set of paths which share the patrol coverage effort as evenly as possible. This was preferred to a straightforward application of a greedy shortest path heuristic in which the hotspots are equally weighted. For large fleet sizes relative to the number of hotspots on the network, the application of a straightforward greedy heuristic would most likely produce an initial basis consisting of both good and bad quality columns. This would be an undesirable outcome compared to an initial basis in which the patrol effort is more evenly distributed. Hence, the RSPH was implemented in an attempt to increase the likelihood of producing initial basis sets of better quality.

The RSPH first applies a cost $\mu_{ij} = -T$ to each patrol arc $(i, j) \in A_{\mathrm{P}}$ and cost $\mu_{ij} = 0$ to each arc $(i, j) \in A_R \backslash A_{\mathrm{P}}$. For each hotspot $\ell \in W$ in the time-space network, a patrol arc (i, j) is chosen at random from the set $A_{\mathrm{P}}(\ell)$. The time index t at the tail of arc (i, j), denoted by $(i, j)_t$, is then multiplied by a random number $\tilde{r}$ drawn from the uniform probability distribution U(0, 1). This value is then negated and added to the cost of each patrol arc in the hotspot. Once the patrol arc costs have been updated in this manner, a shortest path is invoked over the time-space network. By examining the returned shortest path p, we can determine all arcs $(i, j) \in p$ such that $(i, j) \in A_{\mathrm{P}}$. We then update the costs of all $(u, v) \in A_{\mathrm{P}}$ such that $\phi(u, v) = \phi(i, j)$ according to $\mu_{uv} = T$. On the other hand, if $(i, j) \in p$ and $(i, j) \in A_R \backslash A_{\mathrm{P}}$, then the arc cost is updated through $\mu_{ij} = 0$. A new shortest path is subsequently returned from the network with the updated cost structure, and the heuristic continues in a cyclical manner, terminating when the aforementioned procedure has been called $\kappa_{\max}$ times. The procedure is formally outlined in Algorithm 7.

5.7 Pricing Out Candidate Paths

The column generation procedure is initialized by running the randomized construction heuristic presented in Algorithm 7 over the time-space network. Starting from the initial basis produced by the construction heuristic, columns are generated one-at-a-time by solving the pricing subproblem using the dual costs from the current iteration of the RMP. As long as paths of negative reduced cost are returned from the pricing subproblem, the procedure continues on in a cyclical fashion and the paths are added as columns/variables to the RMP. The column generation procedure

Algorithm 7 MCPRP: Randomized shortest path heuristic for column generation initialization

```
Input: A time-space network $G_R = (V_R, A_R)$ with source $s \in V_R$ and sink $\tau \in V_R$
procedure RSPH($G_R, s, \tau$)
    for $(u, v) \in A_R \backslash A_\mathrm{P}$ do
        $\mu_{uv} \leftarrow 0$
    for $\ell \in W$ do
        Randomly select an arc $(i, j) \in A_\mathrm{P}(\ell)$
        for $(u, v) \in A_\mathrm{P}(\ell)$ do
            $\mu_{uv} \leftarrow -T - \tilde{r}(0, 1) \times (i, j)_t$
    $P' \leftarrow \varnothing$
    for $k = 1, \ldots, \kappa_{\max}$ do
        $(p, \delta_p(s, \tau)) \leftarrow$ DAG- SP($G_R, \mu, s, \tau$)
        $P' \leftarrow P' \cup \{p\}$
        for $(i, j) \in p$ do
            if $(i, j) \in A_\mathrm{P}$ then
                for $(u, v) \in A_\mathrm{P}$ such that $\phi(u, v) = \phi(i, j)$ do
                    $\mu_{uv} \leftarrow T$
    return $P'$
```

terminates once the reduced cost of a path returned from the pricing subproblem is non-negative.[9] A straightforward implementation of the procedure is summarized in Algorithm 8.

5.8 Branch-and-Price

As column generation is directly applicable to real variable problems, it can be embedded within a branch-and-bound tree structure in order to solve large-scale integer programming problems. This augmented application of column generation is known as branch-and-price (see [2]). In this section, we propose a straightforward branch-and-price approach to the MCPRP on the time-space network outlined heretofore. If the application of column generation at the root node fails to return an integer solution, we can impose branching cuts to the RMP with respect to the most

[9] By solving the RMP as a linear program and obtaining its dual variables, a subproblem can be solved to determine a new column (variable) to add to the RMP. The subproblem can accomplish this by casting the pricing step of the simplex algorithm (find a variable with negative reduced cost to enter the basis) as an optimization problem. The process iterates between the RMP and the subproblem, terminating when no variable can price-out favourably.

Algorithm 8 MCPRP: Column generation procedure

```
1: **Input**: A time-space network $G_R = (V_R, A_R)$ with source $s \in V_R$ and sink $\tau \in V_R$
2: $P' \leftarrow \text{RSPH}(G_R, s, \tau)$
3: **procedure** MCPRP_GenerateColumns($G_R, s, \tau$)
4:     Construct RMP from $P'$ with associated variables $\{\lambda_p \mid p \in P'\}$
5:     Solve the RMP to get dual variables $\boldsymbol{v}$
6:     $(p, \bar{r}_p^*) \leftarrow$ MCPRP_DualShortestPath($G_R, s, \tau, \boldsymbol{v}$)
7:     **if** $\bar{r}_p^* \geq 0$ **then**
8:         **break**
9:     **else**
10:        Add new variable $\lambda_p$ and associated column to RMP
11:        $P' \leftarrow P' \cup \{p\}$
12:        **goto** 4
```

fractional transit arc in the time-space network.[10] The procedure works as follows. Assume we have a fractional (non-integral) solution to the MCPRP with a basis defined by a set of paths $P' \subset P$. Let $(u, v) \in A_{\text{T}}$. The flow F over the arc (u, v) is given by the sum of the path variables in the current basic solution which use that arc, that is, $F(u, v) := \sum_{p \in \{q \in P' \mid (u,v) \in q\}} \lambda_p$. The most fractional transit arc in the current non-integral solution is therefore given by $(u, v)^* := \arg\min_{(u,v) \in A_{\text{T}}} \tilde{F}(u, v)$, where we have:

$$\tilde{F}(u, v) := \begin{cases} \frac{1}{2} - (F(u, v) - \lfloor F(u, v) \rfloor) & \text{if } F(u, v) - \lfloor F(u, v) \rfloor < \frac{1}{2}, \\ \frac{1}{2} - (\lceil F(u, v) \rceil - F(u, v)) & \text{otherwise.} \end{cases} \tag{5.17}$$

Once the most fractional transit arc has been identified, we can create left and right disjunctive branches under the current fractional solution (a node in the exploratory tree). Therefore, new restricted master problems are required for each branching decision on the left and right. A new RMP created from a branching decision inherits the form of its antecedent tree node with the addition of the following constraint (cut):

$$\sum_{p \in \{q \in P' \mid (u,v)^* \in q\}} \lambda_p \leq \lfloor F((u, v)^*) \rfloor \quad \text{(LEFT)}, \tag{5.18}$$

$$\sum_{p \in \{q \in P' \mid (u,v)^* \in q\}} \lambda_p \geq \lceil F((u, v)^*) \rceil \quad \text{(RIGHT)}. \tag{5.19}$$

[10] An investigation of alternative branching strategies, for example, selecting various combinations of arc variables at a time, is beyond the scope of this paper, but is recommended for future research.

Note that the incorporation of a branching cut (5.18)/(5.19) to the master problem requires that the subproblem's cost structure be modified by adding a dual penalty cost to the arc $(u, v)^*$.

Branching cuts are added to various fractional nodes in the tree in order to find an integer solution to the full problem. The search tree maintains a best (relaxed) upper bound z_{UB} and a best (integer) lower bound z_{LB}. Given a list of unfathomed nodes, we select the node with objective z' such that $z_{\text{UB}} - z'$ is a minimum. Ties between nodes with the same objective can be broken by preferring the node with the greatest ratio of integral non-zero arc variables to the number of non-zero arc variables. In the event that this ratio is unity, we have an integral solution over the arc variables of the underlying network. However, integrality of the network arc variables is not a sufficient condition for the integrality of the path variables. There may be tree nodes for which a solution has $x_{uv} \in \mathbb{Z}^+$ for all $(u, v) \in A_R$, where $\lambda_p \notin \{0, 1\}$ for some $p \in P'$.[11] Therefore, integrality of the path variables must be checked at each node in order to obtain a feasible integer solution. Any nodes which are arc integral but not path integral are pruned from the tree. When a new integer solution has been found, it can be checked against the current best integer lower bound z_{LB}. If the new integer solution is better than the current best lower bound, the lower bound is updated. Otherwise, the newly found integer solution can be pruned from the tree. When the gap between the best relaxed upper bound and the best integer lower bound is closed, and if complementary slackness and feasibility conditions are satisfied, an optimal solution z^* has been found, and the branch-and-price procedure can be terminated.

5.9 Computational Results

5.9.1 Results on Sample Test Problems

In order to benchmark our proposed branch-and-price approach to the MCPRP, we randomly generated a set of 40 geographical networks to be used as the basis for the design of a broad range of test problem instances. Two grid sizes were used to generate the test networks: 30 × 30 min and 60 × 60 min.[12] Given a grid structure, the test problems were generated by randomly placing hotspot locations within the grid, and then assigning time windows of various lengths to each location.[13]

[11] Vanderbeck [13] notes that this phenomenon (i.e. fractional path flows translating to integral arc flows) can occur when the subproblem is a shortest path problem, which is precisely our case. To see why this is so, recall that $x_{uv} = \sum_{p \in P} x_{uvp}\lambda_p$ for all $(u, v) \in A_R$. It is straightforward to see that if $\lambda_p \in \{0, 1\}$ for all $p \in P$, then $x_{uv} \in \mathbb{Z}^+$ for all $(u, v) \in A_R$, since $x_{uvp} \in \{0, 1\}$ for each $(u, v) \in A_R$ and $p \in P$. However, the converse does not hold. That is, it is mathematically possible to find a set of paths taking fractional values which translates into an integral arc flow solution.

[12] For a grid structure of size 30 × 30 min, a patrol vehicle will traverse the breadth/length of the structure in 30 min.

[13] In total, we generated 8 networks with 40 hotspots, 8 networks with 60 hotspots, 16 networks with 80 hotspots and 8 networks with 100 hotspots. Each test network was randomly generated over

In total, 478 separate test problem instances were run, where the number of cars ranged from 2 to at most 35.[14] The branch-and-price approach was always able to find a provably optimal integer solution. The identification of optimality came through the observation that the objective function value at the root node (which is an upper bound on the objective of the optimal integer solution) always matched the objective function value of the integer solution found in each test problem instance. Given the observed absence of an intergrality gap, we can conclude that the time-space network construction of the subproblem is obviously a strong formulation for the MCPRP.

A general observation is that when the vehicle flow on the underlying time-space network was small, that is, when the fleet size was small compared to the number of hotspots to be covered, the branch-and-price approach consistently produced integer root node solutions. The amount of branching required generally increased as the number of vehicles increased, as did the CPU runtime. Another general trend is that for an equivalent number of patrol vehicles, small grid sizes with short hotspot durations were harder to solve (in terms of the number of branching decisions and the CPU runtime) than larger grid sizes with long hotspot durations. This can be attributed to the increased number of routing choices for instances with short hotspot durations and shorter travel times between hotspots (especially given that the shift duration was constant across the entire problem space).

The test problem instances for the 80 hotspot network included 8 test problems containing 80 locations with a single hotspot affixed to each location and another 8 test problems with 40 locations and 2 hotspots for each location. The general runtime trend was better for the second set of instances (that is, the ones with two hotspots per location). Again, this can be attributed to the increased number of routing choices incurred with an increased number of locations on the network grid. This highlights the importance of distinguishing the number of locations from the number of hotspots on the network. The results for the 80 hotspot category suggest that this distinction is non-trivial.

Finally, we observed that the RSPCH was able to solve the 1 min time window problem instances at the root node, that is, no additional columns needed to be generated and no branching was required, even for instances with fleet sizes yielding complete patrol coverage. We note that a problem instance of the MCPRP with 1 min time windows constitutes a special rendering of the Team Orienteering Problem (TOP). This special case is called the TOPTW, in which the profit is 1 for each vertex visited, but with additional constraints imposing strict visiting times for profit collection at the vertices (see [14]).

a grid structure of size 30 × 30 min or 60 × 60 min. The time window length of the hotspots was either 1 min, 5−15 min, 30 min or 30−90 min. All test problem instances were run with an 8 hour shift (that is, 480 min) and a time discretization of 1 min.

[14] The computational results of all test problem instances can be found in Appendix G of the PhD thesis by Chircop [4].

5.9.2 Benchmarking Against the MCNFP Model of Dewil et al. [7]

In order to benchmark and validate our branch-and-price approach to the MCPRP, we compared its performance on the test problem instances of the previous section against the MCNFP model presented in [7]. The MCNFP model was implemented with the network simplex algorithm from the open-source LEMON C++ libraries (see [11]). For each of the test problems, the MCNFP model produced the same optimal objective as the branch-and-price approach. This provides a strong validation for the correctness of the time-space network construct outlined in this article.[15] The MCNFP model demonstrated superior runtime performance to the branch-and-price approach on almost all of the test problem instances. The performance differential became more apparent when the number of patrol cars started to saturate the underlying time-space network. These particular instances required a considerable amount of branching with the branch-and-price approach, and hence, the runtime increased with the number of patrol cars. This demonstrates that the MCNFP formulation of the MCPRP is still the gold standard for networks of the scale tested here.

5.9.3 Large-Scale Problem Instances

In addition to the small- to medium-scale networks of the previous section, we also designed two large-scale problems for which the branch-and-price approach could outperform the MCNFP model over a broad range of fleet sizes. The first large scale network consisted of 200 hotspots over a 100×100 min grid, with the time window lengths ranging between 30 and 90 min. The second network, designed over the same grid size, contained 250 hotspots, with the time window lengths ranging between 30 and 60 min. For both networks, we considered fleet sizes from 2 to 25 patrol cars. On these problem instances, the branch-and-price approach was able to outperform the MCNFP model, with the exception of a small number of cases. Figure 5.3 shows the results for the 200 hotspot case, while Fig. 5.4 contains the results for the 250 hotspot case. These problem instances correspond to situations in which the underlying time-space network is unsaturated with patrol cars, and so many of the integer solutions found with the branch-and-price approach solve at the root node in a shorter time frame than the MCNFP model. However, when we increased the fleet size beyond 25 patrol cars, more intensive branching was required

[15] The computational results for the branch-and-price approach to the MCPRP were produced on a 2.70 GHz dual-core processor on a 32-bit operating system with 4.00 GB of RAM. All primal and dual solutions to the linear programs were obtained with CPLEX 12.6. The column generation and shortest path algorithms, along with the required data structures for the master problem and the time-space network, were coded using the Java programming language and the Eclipse Integrated development environment (IDE). The MCNFP model was run from an executable (on the same operating system) compiled from source code supplied by the authors of [7] and the LEMON C++ libraries using the Microsoft Visual Studio (2012) IDE.

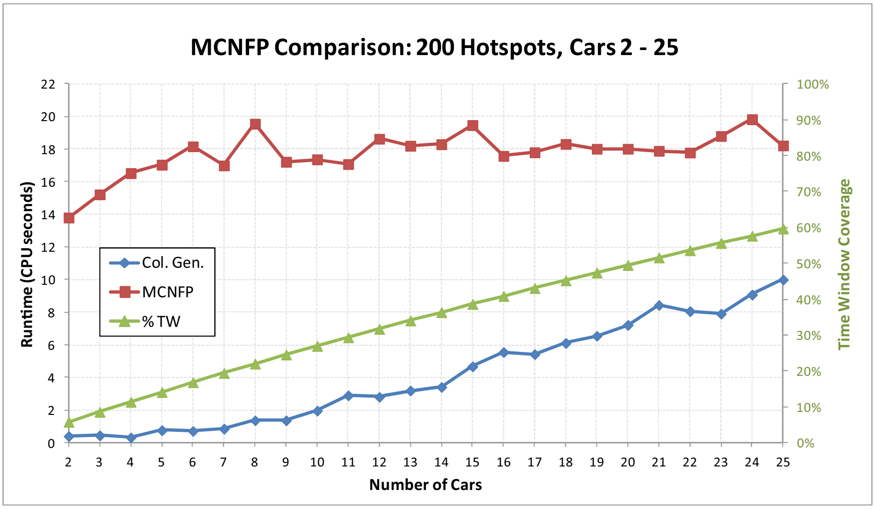


Fig. 5.3 Comparison of the runtime performance of the MCNFP model with the branch-and-price approach on a 200 hotspot test problem. The time windows range between 30 and 90 min, the fleet size ranges from 2 to 25 patrol cars and the amount of time window coverage is shown on the right-hand vertical axis

for the branch-and-price approach, and the MCNFP model began to exhibit better runtime performance. In fact, we observed that the solution time for the MCNFP was independent of the number of patrol cars, in contrast to the branch-and-price approach.

Finally, we generated some further test problem instances on two large-scale networks which could not be solved by the MCNFP model of [7], but which could be easily handled by the branch-and-price approach.[16] These test problems are the largest instances of the MCPRP solved to-date. The size of the network grid structure was 100 × 100 min for both networks. The first network ($H300\_a$) contained 300 spatial locations and 300 hotspots, with time window lengths in the range 30−90 min. The second network ($H500\_a$) contained 250 locations with 500 hotspots of length 30 min, with two hotspots allotted to each location on the network grid.[17] The results are summarized in Table 5.1 for problem instances with 2−25 patrol vehicles.[18] The majority of these problem instances were solved at the root node (no branching

[16] The MCNFP model crashed (due to memory capacity constraints) when we attempted to solve these large-scale instances.

[17] The aggregate duration of all the hotspots in $H300\_a$ was 17,606 min. The aggregate duration for $H500\_a$ was 15,000 min. The shift duration in both cases was 8 hours (480 min).

[18] The headings used in Table 5.1 are as follows: Car—the number of vehicles. R.Obj.—the objective function value at the root node. R.Ti.—the CPU time (seconds) at the root node. R.Col.—the number of columns generated at the root node. No.—the number of nodes explored (fathomed) in the branch-and-price tree. S.Ti.—the CPU time (seconds) taken to find the integer solution. Hot.—the number of hotspots visited.

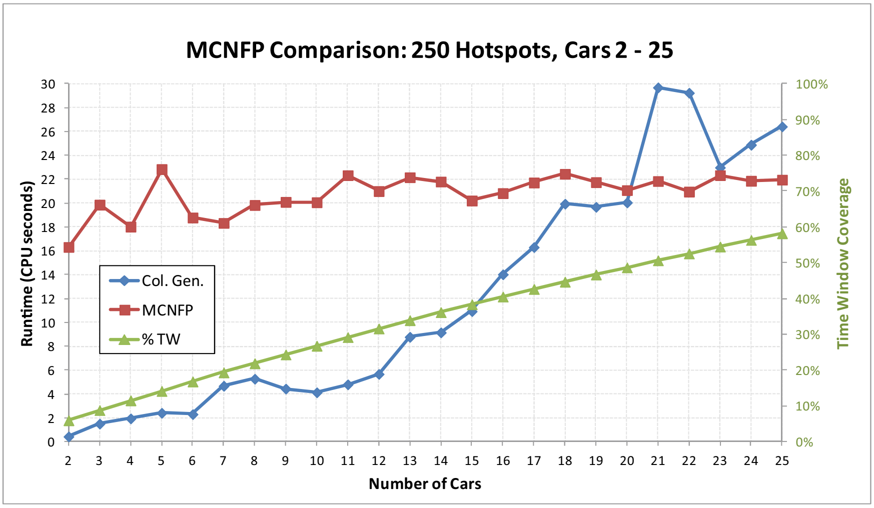


Fig. 5.4 Comparison of the runtime performance of the MCNFP model with the branch-and-price approach on a 250 hotspot test problem. The time windows range between 30 and 60 min, the fleet size ranges from 2 to 25 patrol cars and the amount of time window coverage is shown on the right-hand vertical axis

required). However, given the extremely large size of the underlying time-space networks, the column generation process was much slower than the previously tested networks. For example, when $H500\_a$ was solved with 25 cars, only two branching decisions were required to find an integer solution, but the runtime was approximately 5 min. We again observed that more intensive branching was required as the number of cars began to saturate each network. For example, we tested $H500\_a$ with 48 vehicles, which took approximately 40 min to solve with 22 branching decisions required.

5.10 Conclusions and Future Work

This paper has introduced a branch-and-price framework, underpinned by a specially tailored time-space network, for obtaining solutions to the MCPRP. We introduced a number of test problems for benchmarking, consisting of different numbers of hotspots, time window durations and grid sizes. These test problems were used to validate our approach against the MCNFP model of [7]. While the MCNFP model outperforms the runtime efficiency of the branch-and-price approach on small- to medium-scale problems, it was shown that on large-scale problem instances with certain fleet sizes, the branch-and-price approach can outperform the MCNFP model. We also introduced two large-scale problems, one with 300 hotspots and another with 500 hotspots, which could not be solved by the MCNFP model, but which could be

Table 5.1 Summary of computational results for two large-scale networks

Car	Test problem $H300\_a$						Test problem $H500\_a$					
	R.Obj.	R.Ti.	R.Col.	No.	S.Ti.	Hot.	R.Obj.	R.Ti.	R.Col.	No.	S.Ti.	Hot.
2	735	0.6	7	0	0.6	13	676	2.2	16	0	2.2	27
3	1,085	1.3	12	0	1.3	19	995	2.6	20	0	2.6	41
4	1,434	2.2	23	0	2.2	26	1,312	3.6	28	0	3.6	52
5	1,767	3.2	33	0	3.2	32	1,623	7.5	54	0	7.5	63
6	2,096	4.5	48	0	4.5	38	1,930	10.1	74	0	10.1	76
7	2,423	5.2	56	0	5.2	45	2,234	13.6	100	0	13.6	87
8	2,745	5.7	56	2	11.0	51	2,533	16.4	118	0	16.4	98
9	3,067	5.7	60	0	5.7	56	2,829	22.3	155	0	22.3	108
10	3,385	8.1	86	0	8.1	63	3,123	22.8	162	0	22.8	121
11	3,696	7.1	79	1	10.2	68	3,411	27.3	192	0	27.3	131
12	4,007	9.7	105	0	9.7	73	3,695	32.7	229	1	50.5	143
13	4,308	10.7	119	0	10.7	79	3,973	39.1	271	0	39.1	154
14	4,609	14.0	136	0	14.0	83	4,246	44.1	306	3	84.6	166
15	4,900	14.1	152	0	14.1	89	4,519	49.9	346	0	49.9	176
16	5,189	16.6	176	0	16.6	96	4,789	67.0	447	0	67.0	187
17	5,477	18.1	191	0	18.1	103	5,053	63.6	436	0	63.6	198
18	5,764	16.7	180	0	16.7	108	5,313	73.8	507	0	73.8	207
19	6,048	20.5	213	0	20.5	113	5,572	94.4	633	0	94.4	216
20	6,325	23.9	247	0	23.9	117	5,829	93.2	636	0	93.2	229
21	6,602	27.2	279	0	27.2	121	6,083	117.0	764	1	140.5	238
22	6,872	27.4	280	0	27.4	126	6,332	105.8	727	0	105.8	247
23	7,140	27.6	290	0	27.6	130	6,579	130.7	888	0	130.7	258
24	7,408	30.6	316	1	37.2	136	6,823	160.3	1,019	1	201.5	266
25	7,673	27.4	288	0	27.4	140	7,065	224.4	1,374	2	301.4	274

easily handled with the branch-and-price framework. These large-scale problems are the largest instances of the MCPRP solved to-date.

One avenue for further research of solution approaches to the MCPRP is to investigate the applicability of the branch-and-price framework to the network model developed by Dewil et al. [7]. Utilizing the model of [7] within a branch-and-price framework would not significantly change the structure of the master problem introduced in this paper, except that the packing constraints would correspond to time sections of the hotspots. The master problem's objective function would also need to be modified to incorporate any weights applied to the time sections. The branch-and-price approach could also be applied to variants of the MCPRP which account for overlapping shifts and/or different start/end locations for the patrol cars. The consideration of a heterogeneous fleet for the MCPRP is another possible growth path for the framework introduced in this paper. In this case, we hypothesize that separate subproblems on distinct and specially tailored time-space networks would need to be considered for each vehicle type. We note that if the patrol vehicles do not share a single transit speed, then Theorem 5.1 (see [10]) is no longer valid, and therefore, the patrol arc construct for the hotspots introduced in this paper would need to be revised.

Acknowledgements The authors would like to sincerely thank Dr Reginald Dewil (KU Leuven) for supplying the source code of the Minimum cost network flow problem (MCNFP) model at short notice.

References

1. Ahuja R, Magnanti T, Orlin J (1993) Network flows: theory, algorithms, and applications. Prentice Hall
2. Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch-and-price: column generation for solving huge integer programs. Oper Res 46:316–329
3. Çapar İ, Keskin BB, Rubin PA (2015) An improved formulation for the maximum coverage patrol routing problem. Comput Oper Res 59:1–10
4. Chircop PA (2017) Column generation approaches to patrol asset scheduling with complete and maximum coverage requirements. PhD thesis, University of New South Wales, Sydney, Australia
5. Chircop PA, Surendonk TJ, van den Briel MHL, Walsh T (2013) A column generation approach for the scheduling of patrol boats to provide complete patrol coverage. In: Piantadosi J, Anderssen RS, Boland J (eds) Proceedings of the 20th international congress on modelling and simulation. Modelling and Simulation Society of Australia and New Zealand, pp 1110–1116
6. Desrosiers J, Lübbecke ME (2005) A primer in column generation. In: Desaulniers G, Desrosiers J, Solomon MM (eds) Column generation. Springer, US, pp 1–32
7. Dewil R, Vansteenwegen P, Cattrysse D, Oudheusden DV (2015) A minimum cost network flow model for the maximum covering and patrol routing problem. Eur J Oper Res 247:27–36
8. Ford LR, Fulkerson DR (1958) A suggested computation for maximal multi-commodity network flows. Manag Sci 5:97–101
9. Gilmore PC, Gomory RE (1961) A linear programming approach to the cutting-stock problem. Oper Res 9:849–859
10. Keskin BB, Li SR, Steil D, Spiller S (2012) Analysis of an integrated maximum covering and patrol routing problem. Transp Res Part E(mohana) Logist Transpo Rev 48:215–232
11. Király Z, Kovács P (2012) Efficient implementations of minimum-cost flow algorithms. Acta Univ Sapientiae, Inform 4:67–118
12. Li SR (2012) Vehicle routing models in public safety and health care. PhD thesis, The University of Alabama TUSCALOOSA
13. Vanderbeck F (2005) Implementing mixed integer column generation. In: Desaulniers G, Desrosiers J, Solomon MM (eds) Column generation. Springer, US, pp 331–358
14. Vansteenwegen P, Souffriau W, Van Oudheusden D (2011) The orienteering problem: a survey. Eur J Oper Res 209:1–10

Chapter 6
Linear Complexity Algorithms for Visually Appealing Routes in the Vehicle Routing Problem

Philip Kilby and Dan C. Popescu

Abstract The vehicle routing problem consists of finding cost-effective routes for fleets of trucks to serve customers. Logistics managers often prefer routes to also be "visually appealing" because of the better flexibility they provide in coping with small alterations, required due to last-minute or unforeseen events. Compactness of the routes is a key desirable feature, and it can be accomplished by minimizing the area enclosed by the routes. A common approach in the literature relies on imposing a penalty on the area of the convex hull. We propose to use new features which are well correlated with the convex hull area but are significantly easier to implement, having $O(n)$ computational complexity instead of $O(n\log n)$. By accepting only a minimal loss of quality with respect to a primary objective function, like the routes' total length, we show that area-type penalties can be effective in providing good guidance: construction methods which are based on insertion are naturally steered towards routes displaying more attractive shapes. Used in conjunction with an adaptive large neighbourhood search, our new proposed features lead to routes that exhibit similar compactness compared to using a convex hull area penalty. We also achieve good separation between routes.

Keywords Behaviour tree, Air combat, Genetic programming

6.1 Introduction

The *Vehicle routing problem* (VRP) consists of making optimal use of a fleet of vehicles, in order to serve a pool of clients for their delivery needs. Optimality is generally defined in terms of minimizing a cost function. Typically, the cost incorporates measures of variable quantities such as distance travelled, elapsed time, fuel

P. Kilby (✉) · D. C. Popescu
CSIRO Data61 and Australian National University,
Canberra, ACT 2601, Australia
e-mail: philip.kilby@data61.csiro.au

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_6

consumption. The VRP is an important and well-studied problem and has received much attention in the literature. See [1] or [2] for an overview of this research.

Minimal cost may not be the only desirable feature for the routes of a VRP solution. It is often desirable for the routes to also display several features of visual appeal: to be either non-overlapping or have small intersections; to be compact and roundish; and not self-crossing. These additional features are therefore important and in practice fleet logistics managers often prefer to slightly sacrifice cost efficiency in order to get VRP solution with such "nice looking" routes.

The area of the convex hull covering all points in the route is a common measure of route compactness [3]. The convex hull is the smallest simple polygon covering all customers in the route, excluding the depot. However, the best-known algorithms for computing the convex hull area have $O(n\log n)$ computational complexity. Classical examples include Graham's scan [4] and Jarvis's March [5] algorithms. The monograph by Preparata and Shamos [6] contains a comprehensive overview of the topic, and also shows that the related problem of finding the diameter of a set of point has similar computational complexity.

We are not aware of any measures which specifically address the "roundness" of a route. This paper does not deal with self-crossing.

Several solutions have been proposed in the literature to address the problem of shape quality for VRP. They are based on the general idea of imposing some additional penalties—so-called soft constraints—on routes, in order to favour the desired shape characteristics. Soft constraints could include linear compactness measures, such as assessing for each route how much locations deviate from the route centre, or measures to estimate the extent of the areas covered by individual routes, or some topological features like the number of self-crossings of the routes. While those features have proven useful in achieving good outcomes for properly shaping resulting routes, they can also have higher computational complexity, which can become a drawback when dealing with VRP problems of large size.

In this paper, we propose several features of low computational cost, aimed at acting as soft constraints for "nice looking" routes. Our paper is organized as follows. Section 6.2 contains a brief literature review. We present in detail our new proposed soft constraints in Sect. 6.3 and in Sect. 6.4 we outline how they are integrated into the optimization process. Section 6.5 describes our VRP solver *Indigo* used for conducting the experiments, and the experimental results are analysed in Sect. 6.6. We present our conclusions in Sect. 6.7.

6.2 Existing Methods

Soft constraints aimed at improving the visual attractiveness of routes resulting from a VRP solution are proposed in [7] by Poot et al. Several measures for compactness and low overlap for routes are derived with reference to the geometric (or gravity) centre of the routes. One measure consists of counting how many locations have another route centre closer than their own route centre; ideally this number should be as low as possible. In a similar vein, one could aim to minimize either the average distance to the centre of the route, or the average distance between all locations in a route. Another measure of overlap counts the number of locations that are contained within the convex hulls of a different route. Topological measures proposed in the same paper include the number of self-crossings within routes and the number of crossings between different routes. The measures are evaluated separately from the classical measures such as time or distance travelled, and prove to provide routes with more clustered locations.

In [8] the compactness of a route is assessed with respect to a differently defined centre, chosen from existing locations to be served by that route. This centre, termed median of the route, is defined as the location that minimizes the sum of the distances of all other locations in the route to this particular location. Using the median centre one can derive shape measures, such as the number of locations that are closer to the median of another route than to the centre of their own route, or average distance to the route centre. This paradigm has the advantage of working in a coordinate-free setup when only distances between locations are known and no coordinates are available. However, it may also raise some problems due to the possible lack of stability of the median to small variations in location positions. This could be the case, for example, if the locations on a route are evenly distributed on a circular path. These measures are incorporated into an interactive procedure, which allows an experienced operator to initialize a solution and then iteratively guide it towards an optimal one, which would satisfy both cost efficiency and visual attractiveness constraints.

The paradigm proposed in [9] is to incorporate both the standard efficiency costs of a route and the visual attractiveness costs of a route into a single scalar function. This objective function is then used in conjunction with a guided local search algorithm [10] to produce VRP routes with visual attractiveness, at the expense of only minor deviation from optimality.

6.3 Linear Complexity Soft Penalties

In this section, we describe several soft constraints of low computational cost, which can be used to influence the shapes of the routes. In the following we assume that a route $\mathscr{R}$ is a path through a set of n planar locations $\mathscr{L} : \{L_i = (x_i, y_i),\ 1 \leq i \leq n\}$, where x_i, y_i are the spatial coordinates. Let $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ denote the coordinates of the geographic centre of $\mathscr{L}$. We denote by $D(\mathscr{L})$ the Euclidean distance matrix of the set $\mathscr{L}$: $D(\mathscr{L})_{ij} = ||L_i - L_j||$, where $||.||$ denotes the usual Euclidean norm. All our proposed constraints have linear complexity as a function of the number of locations, $O(n)$. The first four measures presented are proxies of area estimation, while the last two are estimators of "roundness". Figure 6.1 encapsulates pictorial representations corresponding to the first three area features in the top row, and of the two roundness features in the bottom row.

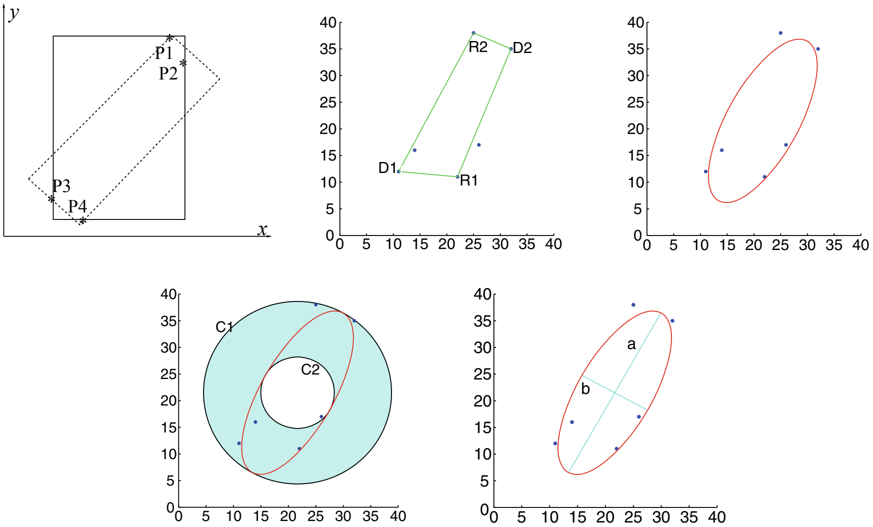


Fig. 6.1 Illustrations of soft constraint features. Top row, left to right: rotated bounding box, maximal quadrilateral area and covariance ellipse. Bottom row, left to right: differential area and eccentricity

6.3.1 *Rotated Bounding Box Area*

The bounding box of the set $\mathscr{L}$ is the rectangle $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$, where $x_{min} = \min_{i=1}^{n}(x_i), x_{max} = \max_{i=1}^{n}(x_i)$ and similarly $y_{min} = \min_{i=1}^{n}(y_i), y_{max} = \max_{i=1}^{n}$ (y_i). Because the bounding box establishes tight limits on both the x and y directions, its area gives a reasonable estimate of the area covered by the points in the set $\mathscr{L}$. Nevertheless, because this estimate is only made with respect to two pre-defined orthogonal directions, the area of the bounding box can misrepresent the area of $\mathscr{L}$. This can be easily seen from the example shown in the top row leftmost diagram of Fig. 6.1, where the rectangular bounding box, drawn in black colour, is covering considerably more area than the set of four points it bounds. This situation could be improved upon if we were allowed to gradually rotate the bounding box sides, such that they do not necessarily align with the coordinate axes. Computing a rotated bounding box by angle θ is easy, by taking the same *min* an *max* coordinate limits on the rotated set of points:$\mathscr{L}_\theta = (x_i cos\theta + y_i sin\theta, -x_i sin\theta + y_i cos\theta), 1 \leq i \leq n\}$. We remark that because the set $\mathscr{L}$ lies inside any such rotated bounding box, the area of the bounding box will always be larger than, or equal to, the area spanned by $\mathscr{L}$. Therefore, if we compute several rotated bounding boxes, the best area estimate would result from the minimum area of all estimates. One such rotated box is shown in the same diagram, drawn with dash lines. Ideally, we may wish to rotate the bounding box continuously, to find an optimal estimate. In practice, for computational efficiency reasons, we need to settle for just a few discrete rotations. We propose to use just two bounding boxes: the original box (no rotation) and one more box with 45° rotation. Our choice is motivated by the observation that, by allowing for the extra 45° rotation, the underestimation of the diameter of the set $\mathscr{L}$ is reduced from a worst case of 41 % (as exemplified by the box drawn with continuous line) to a worst case of about 8 %. The only computational penalty is the approximate doubling of the $O(n)$ time, to compute two bounding boxes instead of just one.[1]

6.3.2 Approximate Square Diameter

The diameter of the set $\mathscr{L}$ is defined as the largest distance between any pair of points in the set. As previously pointed out, an exact computation of the diameter has computational complexity $O(n\log n)$. We propose to use an "approximate diameter" of computational complexity $O(n)$ by using the previously defined approximate bounding box. This is defined as the maximum of two distances: the distance between the two points of $\mathscr{L}$ defining the limits of the x direction of the box, and the distance of the two points of $\mathscr{L}$ defining the y direction of the box. For example, with reference to the diagram in the leftmost position in the top row of Fig. 6.1 this would lead to the distance between points P1 and P3. While this is an approximation, in practice the error turns out to be quite small. Our tests over more than 10,000 simulated random sets of point, of sizes ranging from 10 to 100, indicate that on average the error in estimating the true length of the diameter is less than 2%. In fact, in more than 70% of the cases the approximate diameter method actually found the real diameter.

We could use this approximate diameter as a soft constraint feature. However, in this study we prefer to use the square of this approximate diameter, which we denote by $Diam^2$ as a soft feature, to obtain an area-like feature similar to the other soft constraints we propose.

6.3.3 Maximal Quadrilateral Area

The idea of the Maximal Quadrilateral Area soft penalty is to approximate the area covered by the set of points $\mathscr{L}$ by the area of a more suitably fitted quadrilateral than a rectangular bounding box. This is exemplified by the diagram in the middle of the top row in Fig. 6.1, for a set of six points. First, we determine two points D_1 and D_2 which define the approximate diameter of the set $\mathscr{L}$, as described in the previous section. The straight line through D_1 and D_2 splits the plane into two halves. Assuming there are points of $\mathscr{L}$ in each of those two half planes, we then choose

[1] We could of course continue this process of angle orientation refinement. However, it turns out that adding one more level of angle subdivision would require considering another four rotations, then another eight rotations for the next level of refinement and so on. We believe any such additional computational effort is not justified for achieving no more than 8% additional accuracy.

from each half plane the point that is farthest away from the line through D_1 and D_2, thus getting two new points, say R_1 and R_2. The points D_1, D_2, R_1, R_2 define the vertexes of the maximal quadrilateral. For those cases when one of the half planes does not contain any point of $\mathscr{L}$, our maximal quadrilateral reduces to a triangle.

6.3.4 Covariance Ellipse Area

The covariance ellipse is a “best fitting” ellipse to the set of locations of the route. It is a concept often used in pattern recognition problems for shape and orientation estimation, see for example [11]. The ellipse is derived from the positive definite matrix $C_{\mathscr{L}}$, the covariance matrix of $\mathscr{L}$:

$$C_{\mathscr{L}} = \frac{1}{n} \begin{bmatrix} m_{xx} & m_{xy} \\ m_{xy} & m_{yy}, \end{bmatrix} \tag{6.1}$$

where

$$m_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2 \qquad m_{yy} = \sum_{i=1}^{n} (y_i - \bar{y})^2 \qquad m_{xy} = \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$

are the second-order central moments of the coordinates. The centre of the ellipse is located at ($\bar{x}$,$\bar{y}$). The eigenvectors of the covariance matrix give the orientations of the two axes of the ellipse, and the square roots of the corresponding eigenvalues determine the sizes of the axes. The area of the covariance ellipse is given by

$$A_{\mathscr{L}} = \frac{2\pi}{n} \sqrt{m_{xx} m_{yy} - m_{xy}^2} \tag{6.2}$$

and has $O(n)$ computational complexity, which is the complexity for calculating the first- and second-order moments. We remark in passing that the covariance ellipse area can also be calculated directly from the Euclidean distance matrix $D(\mathscr{L})$, however with the penalty of increasing the computational cost to $O(n^2)$. This could be useful in situations when only distances between locations would be available, and no coordinates are known.

The covariance ellipse corresponding to a set of six locations is shown in the leftmost diagram of the top row in Fig. 6.1.

6.3.5 Differential Area

The differential area is an estimator of “roundness” which is also sensitive to scale. It is defined as the difference between the area of the smallest circle that contains the covariance ellipse of the set of location and the largest circle that can fit inside the same ellipse. The two circles defined above have as diameters the major and minor axes of the covariance ellipse and have their centres at the centre of the ellipse. The differential area is therefore the area of the circular crown bounded by the two circles. It measures roundness because low values for this feature indicate roundish shapes, while higher values indicate elongated shapes. If the geometric configuration defined by the points in $\mathscr{L}$ is spread out unevenly, there will be a significant difference between the sizes of the two axes of the covariance ellipse. In such a case, like the one pictured in left diagram in the bottom row of Fig. 6.1, the differential area will be relatively large. Conversely, if the pattern of $\mathscr{L}$ is evenly spread in all directions, the covariance ellipse will be close to a circle; its two axes will be approximately equal and therefore the differential area would be relatively small. It is nevertheless a feature that takes scale into account: if the pattern $\mathscr{L}$ scales up uniformly by a linear scaling factor s, then the differential area scales up by s^2. The differential area of $\mathscr{L}$ can be easily computed in terms of the central moments defined in the previous section as

$$dA_{\mathscr{L}} = \frac{2\pi}{n}\sqrt{(m_{xx} - m_{yy})^2 - 4m_{xy}^2} = \frac{2\pi}{n}\sqrt{\Delta}, \tag{6.3}$$

where we have used the abbreviated notation $\Delta = (m_{xx} - m_{yy})^2 - 4m_{xy}^2$.

6.3.6 Eccentricity

The eccentricity of the set $\mathscr{L}$ is another measure of roundness, also derived from the covariance ellipse. It is defined as the eccentricity of the corresponding covariance ellipse. This is a classical geometry notion defined as

$$E_{\mathscr{L}} = \sqrt{1 - \frac{b^2}{a^2}}, \tag{6.4}$$

where a and b are the lengths of the major and minor axes of the ellipse, respectively, as shown in the rightmost diagram of the bottom row in Fig. 6.1. The values of the eccentricity are in the range [0, 1], with values close to 1 indicative of an elongated shape and values close to 0 indicative of a roundish shape. As opposed to the differential area, the eccentricity value is insensitive to scale. With the notations of the previous sections, the eccentricity of the set of locations $\mathscr{L}$ can be computed from the central moments as

$$E_{\mathscr{L}} = \sqrt{\frac{2\sqrt{\Delta}}{m_{xx} + m_{yy} + \sqrt{\Delta}}}, \tag{6.5}$$

which again involves only $O(n)$ computational complexity.

The two diagrams of Fig. 6.2 illustrate a comparison of all area-based soft constraints, for both a set of 6 locations (upper diagram) and a set of 14 locations (lower

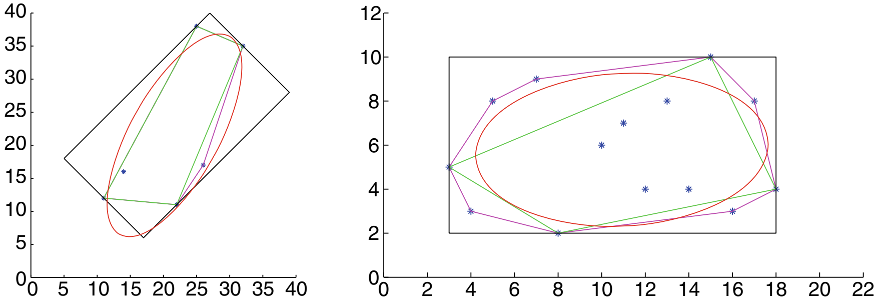


Fig. 6.2 Comparative illustration of the soft features for a 6 points set (upper diagram), and a 14 points set (lower diagram)

diagram). The convex hull area has also been included, shown with magenta colour. We notice that the rotated bounding box generally tends to overestimate the area covered by the set of locations $\mathscr{L}$. The convex hull is better at establishing tighter border limits, while the maximal quadrilateral has a tendency of underestimating the spread of $\mathscr{L}$. The covariance ellipse is quite good at estimating the spread of when the point in $\mathscr{L}$ is concentrated around the border, as in the case in the upper diagram of Fig. 6.2. It can however underestimate area when there are many interior points, as is the case in for the example in the lower diagram of Fig. 6.2, because the corresponding central moments of the covariance matrix are drawn towards lower values by the interior points.

In the end of this section, we point out that all soft features analysed so far are characterized by a non-negative value, and that the *minimization* of this value generally leads to more appealing shapes: either more compact or more roundish.

6.4 VRP Guided by Features

We use the features outlined in the previous section to guide VRP solutions towards more appealing routes. Our aims are as follows:

1 : To have a very simple and efficient computational model. Therefore, just one of the features presented in the previous section is calculated for all generated routes and used to guide the quality of a VRP. No cross measures between routes, such as area overlaps or crossing points between different routes are calculated during the search.

2 : To continue to place emphasis on the original objective function of the VRP optimization model, which is typically a cost associated with the size of the routes. However, in order to accommodate adjustment towards better values of the soft constraints, we are prepared to sacrifice the optimality of this objective function by a small amount.

Suppose a VRP solution consists of a set on m routes $\mathscr{R}_i$, $1 \leq i \leq m$. Without any soft constraints, a typical primary objective function to minimize would be

$$\mathscr{O} = \sum_{i=1}^{m} |\mathscr{R}_i|, \tag{6.6}$$

where $|\mathscr{R}_i|$ denotes the length of route i.

We propose to incorporate the soft constraints by minimizing instead a "blended" objective function $\mathscr{O}_b$ defined as

$$\mathscr{O}_b = \mathscr{O} + \alpha \mathscr{S}, \tag{6.7}$$

where $\mathscr{S}$ could be any of the soft features previously described, and α is a positive weight controlling the degree of blending. Minimizing $\mathscr{O}_b$ does not, in general, result in a minimization of the primary objective $\mathscr{O}$; however, we show that, by allowing for a small increase in the value of $\mathscr{O}$, one can achieve routes displaying significantly better visual appeal. Quantitatively, we measure the visual appeal by two factors: the area size of the mutual intersections of the routes and the total area of the convex hulls of the routes. While the optimization formulation of (6.7) does not explicitly incorporate any of those two measures, our experiments show that with proper choice of the blending weight α significant decreases for both intersections and convex hulls can be obtained. However, due to the dependence on the magnitude of the objective value, some care is required in the selection of α.

6.5 Experimental Context

The performance of these "visual appeal" measures was tested within the context of a VRP solver called *Indigo* [12]. The operation of *Indigo* is essentially identical to the Adaptive large neighbourhood search (ALNS) procedure described by Ropke and Pisinger [13]. Although *Indigo* is capable of more specialized visit selection and ordering heuristics, it was tuned in a manner that replicates the ALNS procedure of [13]. Although [13] focuses on the Pickup and Delivery problem with Time Windows, the ALNS procedure has been shown to be effective in a large number of vehicle routing contexts [15–22].

Starting with a solution created using a construction technique, ALNS then uses a "destroy and reconstruct" iteration to improve the solution. In the destroy phase, a number of visits are removed from their current position in the solution. A reconstruct technique is then used to re-insert them back into the solution. The destroy phase can use one of a number of methods to select visits for removal. For instance, one method is based on geographic neighbourhood. One visit is chosen at random as a starting point. Visits are then chosen at random with a bias inversely proportional to the distance from the originally selected visit. More details are available in [13].

One of a number of reconstruction techniques can be selected to repair the solution. A simulated annealing [14] procedure is then used to conditionally accept the new solution as incumbent. The destroy and re-create iteration is repeated until a stopping criteria is met. In the runs reported here, we use the same as [13]: 25,000 iterations. Both the original construction and the subsequent reconstructions, use a successive insert procedure. At each iteration, one unscheduled customer is selected, and inserted into the solution. The visit to be inserted is chosen by heuristics. The visit is then inserted in the position that increases the objective by the smallest amount.

A number of heuristics are available to choose the visit to be inserted. For instance, the method may choose the visit that increases the objective by the least amount (the so-called *min insert cost* selection. Ropke and Pisinger also use "regret"-based methods. A 2-regret method looks at the difference between the best and second best insertion points for a customer. The customer to be inserted is the customer with the maximum regret—i.e. the one with the most to lose if they don't get their preferred position. In 3-regret, the costs of best, second best and third best positions are taken into account. A 4-regret is defined analogously. The other insert heuristic used by Ropke and Pisinger is a random selection of customer to insert. All customer insertion procedures insert the customer in the position that increases the cost by the smallest amount.

The only difference between the standard ALNS algorithm and our "visually appeal" algorithm is that a penalty term is added to the objective. That is, when the minimum insert cost or regret feature considers the insert cost, it looks at the blended objective function defined in (6.7), rather than the usual objective defined in (6.6).

Note that in order to select the best customer to insert and the best position to insert that customer, we must calculate the change in cost for insertion of that customer at the given position. Using the standard objective (6.6), this is a simple O(1) calculation. Using objective (6.7), however, requires us to calculate the change in our area measures induced by including the new customer. This may involve determining whether the customer is within the convex hull and if not then the increase in the area of the convex hull if the point were to be included. Using a true convex hull, these are $O(n)$ and $O(n \log n)$ calculations. Using our surrogate methods, these can both be achieved in constant time.

We use the term *guidance* to describe the way that augmenting the objective with a term based on the incremental penalty defined by the features we have described can guide the (re-)construction heuristic towards a solution that exhibits the desired property.

6.6 Experimental Results

We have applied the minimization strategy of (6.7) for several VRP scenarios taken from Taillard et al. [23] and also for several time-constrained Solomon instances [24], using the features described in Sect. 6.3.

Indigo requires the number of vehicles available as input. For the Taillard instances, this number is set at the number of vehicles in the best-known solution, as reported in *CVRPLIB* [25]. Only the "Random" ("R") instances of the Solomon set were used for testing; the solutions to the clustered ("C") instances already exhibit compact, non-overlapping structure. Similarly, the random/clustered ("RC") solutions already exhibit good "visual appeal". The number of vehicles available was based on best-known values from [26]. However, due to the way the Solomon instances were created, using only the minimum number of vehicles in these instances sometimes severely limits the pool of feasible routes, which in turn limits the extent to which solutions can exhibit visual appeal. Therefore, one additional vehicle was made available in each instance.

For each scenario a range of blending weights α was used, starting from $\alpha = 0$ which corresponds to "no blending". The values of the parameters for the base case ($\alpha = 0$) have then been used to normalize all relevant parameters, in order to make the interpretation of the results corresponding to values of ($\alpha > 0$) more intuitive.

For each experiment, an upper tolerance limit, typically between 1 and 15%, was imposed on how much increase in the value of the primary objective function was acceptable. Only the VRP solutions for weight values α that met this limit were considered. In viewing these solutions, it was observed that the reduction in convex hull area was very similar. However, even though not directly encouraged, the methods also reduced the overlap between convex hulls. As this is also a highly desirable feature of visually appealing routes, we favoured routes with minimal intersection area. We use α_{opt} to designate the value of the weight giving the solution with minimal convex hull intersection. Our experiments show that the value α_{opt} depends on the type of soft feature selected. A reasonable value can be obtained in a few run of the solver. This value can be used for similar instances but must be recalculated, for instances, with different characteristics, e.g. time windows.

We ran experiments over 13 VRP problems without time windows and over 23 problems with time windows. The results in each category have followed similar trends. We generally found that when the blending feature $\mathscr{S}$ was chosen to be an "area-like" feature, we were able to obtain significantly better looking routes, both in terms of routes' intersection area and routes' total convex hull area, for modest losses in primary objective optimality. However, when the blending feature $\mathscr{S}$ was chosen to be a "roundness" feature—either eccentricity or differential area—the routes did not improve in quality, even for significant losses in primary objective optimality. The figures below show in detail two of the optimization results using the function $\mathscr{O}_b$, for the value of the weight α_{opt} for all area-like features $\mathscr{S}$. In these experiments, the upper limit increase allowed on the primary objective function $\mathscr{O}$ was 5%.

In Fig. 6.3 are shown the original, unbiased results of a VRP problem without time windows, having 150 locations to be served. The diagrams on the left-side display the corresponding routes, which start at the depot location and generally follow minimal geometric paths, because of the absence of time window constraints. For clarity purposes, the last leg of each route, which consists of the vehicle returning to the depot, has not been marked. The corresponding diagrams on the right-side display the convex hulls of the routes. A comparison of the diagrams in the left

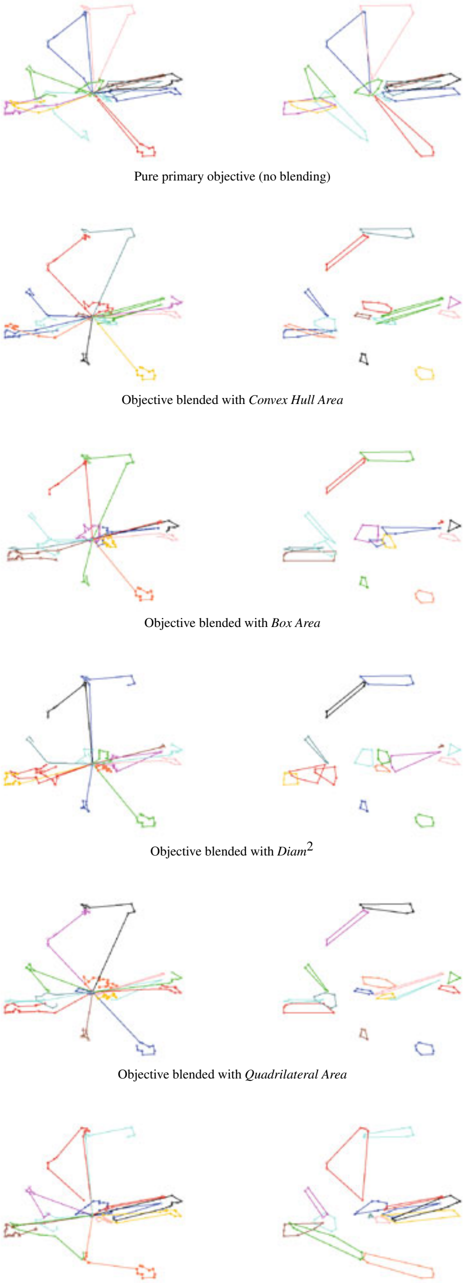


Fig. 6.3 Optimization results corresponding to the problem in file r150b

Pure primary objective (no blending)

Objective blended with *Convex Hull Area*

Objective blended with *Box Area*

Objective blended with $Diam^2$

Objective blended with *Quadrilateral Area*

Objective blended with *Ellipse Area*

Table 6.1 Parameters corresponding to the optimization results of Fig. 6.3

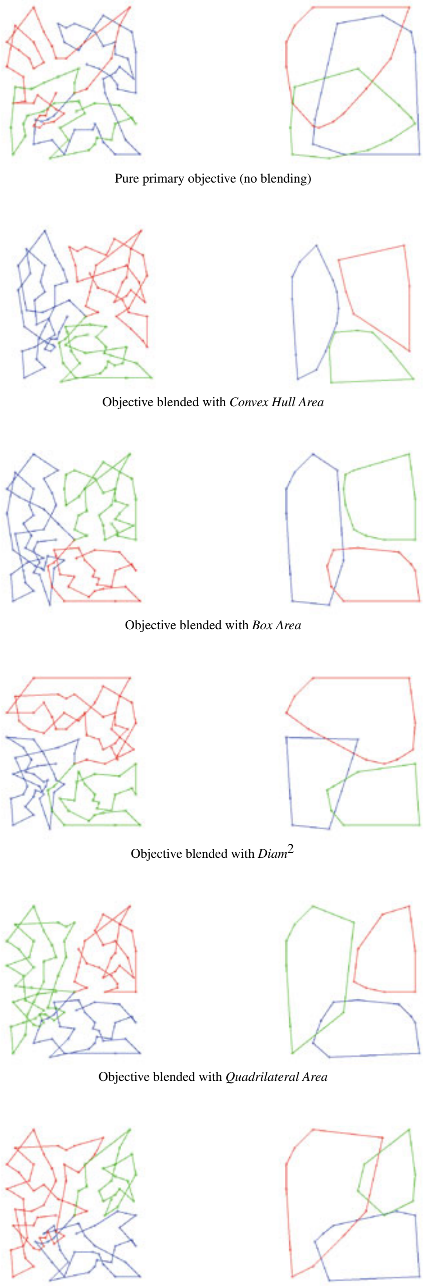
Feature	α_{opt}	Objective	Inters	Conv hull	Ecc
Conv hull	0.5	1.045	0.06	0.24	1.01
Box area	0.5	1.047	0.05	0.29	0.96
$diam^2$	0.05	1.048	0.05	0.30	0.91
Quad area	1.00	1.048	0.01	0.26	1.02
Ell area	0.0075	1.014	0.28	0.66	1.01

column reveals that indeed, as imposed, there is no significant change in the value the primary objective function—the routes' total length. However, both the routes' convex hull mutual intersections and the total area of the routes' hulls have been significantly decreased by optimizing the function $\mathscr{O}_b$, as opposed to the primary function $\mathscr{O}$. This is clearly seen by comparing the first diagram in the second column of Fig. 6.3 against the other diagrams in the second column. The results in Table 6.1 briefly quantify these results. It is most interesting to remark that all area-like features help in achieving a spectacular reduction in the routes intersection area. It may look, based on this particular example, that the ellipse area feature is less suitable than the other area-like features at driving an attractive looking solution. This, however, turns out to be just an artefact of imposing the 5% limit on the loss of primary objective optimality, which is admittedly a bit artificial. If an 8% limit is imposed instead, the results for all features, including the ellipse area turn out to line up in pretty much the same range: more than 95% drop in intersection area, more than 70% drop in total convex hulls area.

Figure 6.4 shows the optimization results for a VRP problem with time windows, on 100 locations. As expected, the time window constraints are reflected in the geometrically more intricate routes, with lots of self crossings. With respect to the visual aspect of the routes, we notice that the results are in the same vein as those of the unconstrained problem of Fig. 6.3: that is, an optimal blending of area-like features into the optimization function brings considerable improvements in both the routes' intersection areas and the routes convex hulls total area. The quantitative results of the optimization results for this VRP problem are summarized in Table 6.2. We notice that the general improvement in intersection area is again by more than 90%, while the drop in total convex hull area is still relevant, around 40%.

The last column in Tables 6.1 and 6.2 reveal that roundness, as defined by the eccentricity soft feature, does not correlate well with the decrease in area. For the optimal solutions, there is either insignificant appreciation or more often, some depreciation of the roundness factor.

Statistics on the original VRP solutions for all experiments are shown in the graphs of Figs. 6.5, 6.6, 6.7 and 6.8. Figure 6.5 shows the average decrease in the routes intersection area for the original VRP solution, for all problems without time windows constraints. The plots are drawn for several thresholds of acceptable increase in the primary objective function, ranging from 1 to 15%. We can see that for less



Pure primary objective (no blending)

Objective blended with *Convex Hull Area*

Objective blended with *Box Area*

Objective blended with *Diam*^{2}

Objective blended with *Quadrilateral Area*

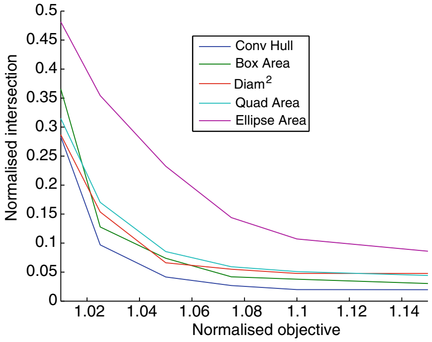
Objective blended with *Ellipse Area*

Fig. 6.4 Optimization results corresponding to the problem in file r210

Table 6.2 Parameters corresponding to the optimization results of Fig. 6.3

Feature	α_{opt}	Objective	Inters	Conv hull	Ecc
Conv hull	0.2	1.034	0.00	0.50	1.15
Box area	0.05	1.025	0.03	0.54	1.03
$diam^2$	0.02	1.039	0.04	0.59	1.07
Quad area	0.1	1.035	0.03	0.53	1.10
Ell area	0.0075	1.038	0.08	0.61	1.10

Fig. 6.5 Average normalized intersection area of original routes, over solutions of VRP without time windows



than a 5% increase in the primary objective function all area-based soft features help towards guiding towards VRP solutions on significantly smaller intersection of mutual routes. In fact, with the exception of the ellipse area, all other features help reduce the intersection, on average, by more than 90%. The slightly different performance of the ellipse area soft feature can be explained by the fact that it underestimates bounding area by incorporating interior points in the area count, which may drive the separation of adjacent regions at lower speeds. Figure 6.6 shows the corresponding average decrease in convex hull areas for the same set of VRP scenarios. As expected we notice that using the convex hull area itself as a soft feature gives the best results, but this comes at the expense of having to compute a feature having $O(n\log n)$ computational complexity. We can also remark that quite similar results are obtained by using either the box area, quadrilateral area or the $Diam^2$ feature, all of which have $O(n)$ computational complexity.

Figures 6.7 and 6.8 show the similar statistics over the 23 problems with time window constraints. Again we notice the very significant drop in the routes' intersection areas if an extra penalty of about 5% in the primary objective function can be tolerated. The bounding box area is performing almost as well as the convex hull area, followed closely by the quadrilateral area. There is also improvement in the total area of the convex hulls of the routes, as can be seen from the graphs in Fig. 6.8, although the decrease is less steep than for the unconstrained scenarios of Fig. 6.6. Overall, the statistics for both types of problems recommend the box area as the preferred choice of soft constraint with $O(n)$ computational complexity, for obtaining

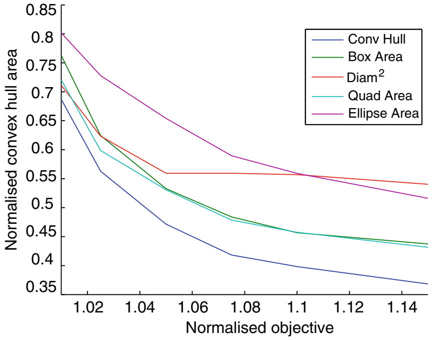


Fig. 6.6 Average normalized convex hull area of original routes, over solutions of VRP without time windows

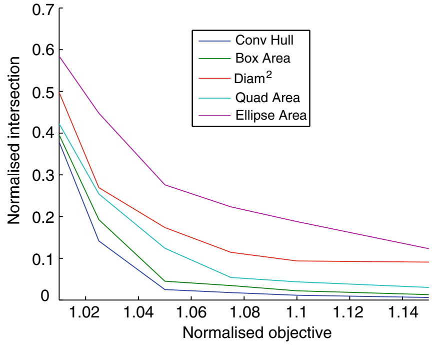


Fig. 6.7 Average normalized intersection area of original routes, over solutions of VRP with time windows

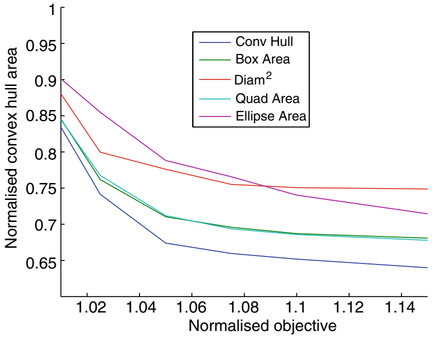


Fig. 6.8 Average normalized convex hull area of original routes, over solutions of VRP with time windows

routes of good visual appeal. It is followed closely by the quadrilateral area, while *Diam*^{2} still appears to be a reasonable choice.

When we have run experiments similar to those summarized in Figs. 6.5, 6.6, 6.7 and 6.8 using both eccentricity and differential area, for the soft penalty feature $\mathscr{S}$ of (6.7). The results have not been encouraging. By using the eccentricity soft features, hardly any improvement has been achieved both in terms of intersection area or total convex hull area, by accepting a deterioration in the primary objective function of 15%. Under the same conditions, the differential area feature has performed slightly better, however only relatively modest gains were achieved: on average, less than 25% decrease in intersection area, and less than 10% decrease in total convex hull area. This is not very surprising, as the roundness measures do not directly penalize area, and hence cannot provide guidance towards compact routes.

To summarize, we propose the following paradigm for generating VRP solutions with an increased visual appeal:

1 : Choose an acceptable threshold of depreciation in the primary objective function $\mathscr{O}$ to be minimized, and a low computational cost soft constrained feature $\mathscr{S}$ (e.g. box area).

2 : Run the VRP optimization on the primary objective to get the best solution without soft constraints.

3 : Run the VRP optimization on the blended objective function $\mathscr{O}_b$ for a discrete number of the weight α, suitably chosen to match the problem. Choose the best visually appealing VRP solution that still matches the depreciation threshold on $\mathscr{O}$.

6.7 Conclusions

We have proposed a new paradigm for generating VRP solutions having visual appeal. The visual appeal is measured primarily in terms of low intersection between the areas covered by different routes, and also by the total size of the convex hulls of the routes.

Our method requires the calculation of soft constraints, which have $O(n)$ computational complexity and can therefore easily be integrated into intensive ALNS search. The soft features are used to adjust the primary objective function, which helps guide the search towards routes with better visual appeal. Our soft features are also much easier to implement than the canonical convex hull.

We have also described how insertion-based reconstruction techniques embedded in an ALNS method can be guided by our measures of visual appeal to construct solutions that exhibit the desired properties.

Our experimental results show that by using area-type soft constraints to adjust the objective function to be optimized, very significant gains in visual appeal can be obtained for only minor losses in optimality as defined by the VRP primary objective function. The experiments have also demonstrated that the use of roundness features as soft constraints does not guide the search towards VRP solutions of increased visual appeal.

References

1. Golden B, Raghaven S, Wasil E (eds) (2008) The vehicle routing problem: latest advances and challenges. Springer, Berlin
2. Toth P, Vigo D (eds) (2014) Vehicle routing: problems, methods and applications. MOS-SIAM series on optimzation. SIAM

3. Hollis BL, Green PJ (2012) Real-life vehicle routing with time windows for visual attractiveness and operational robustness. Asia Pac J Oper Res 29(4)
4. Graham RL (1972) An efficient algorithm for determining the convex hull of a finite planar set. Inf Process Lett 1:132–133
5. Jarvis RA (1973) On the identification of the convex hull of a finite set of points in the plane. Inf Process Lett 2:18–21
6. Preparata FP, Shamos IM (1985) Computational geometry, 2nd edn. Springer, New York
7. Poot A, Kant G, Wagelmans APM (2002) A savings based method for real-life vehicle routing problems. J Oper Res Soc 53(1):57–68
8. Tang H, Miller-Hooks E (2006) Interactive heuristic for practical vehicle routing problem with solution shape constraints. Transp Res Rec: J Transp Res Board J 1964(1):9–18
9. Hollis BL (2010) Vehicle and crew routing and scheduling. PhD Thesis, School of Mathematics and Physics, The University of Queensland
10. Kilby P, Prosser P, Shaw P (1999) Guided local search for the vehicle routing problem. In: Metaheuristics: advances and trends in local search paradigms for optimization. Kluwer Academic, pp 473–486
11. Dougherty G (2013) Pattern recognition and classification. Springer, New York
12. Kilby P, Verden A (2011) Flexible routing combing constraint programming, large neighbourhood search, and feature-based insertion. In: Proceedings 2nd workshop on artificial intelligence and logistics, (AILOG-2011), pp 43–48
13. Ropke S, Pisinger D (2006) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp Sci 40(4):455–472
14. Azi N, Gendreau M, Potvin JY (2006) An adaptive large neighborhood search for a vehicle routing problem with multiple routes. Comput Oper Res 41:167–173
15. Bent R, Van HP (2006) A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. Comput Oper Res 33(4):875–893
16. Hemmelmayr VC, Cordeau JF, Crainic TG (2012) An adaptive large neighborhood search heuristic for two-Echelon vehicle routing problems arising in city logistics. Comput Oper Res 39(12):3215–3228
17. Koç Ç, Bektaş T, Jabali O, Laporte G (2016) The impact of depot location, fleet composition and routing on emissions in city logistics. Transp Res Part B: Methodol 84:81–102
18. Mancini S (2016) A real-life multi depot multi period vehicle routing problem with a heterogeneous fleet: formulation and adaptive large neighborhood search based matheuristic. Transp Res Part C: Emerg Technol 70(7):100–112
19. Pillac V, Guéret C, Medaglia AL (2013) A parallel matheuristic for the technician routing and scheduling problem. Optim Lett 7(7)
20. Maknoon Y, Laporte G (2017) Vehicle routing with cross-dock selection. Comput Oper Res 77:254–266
21. Dayarian I, Crainic TG, Gendreau M, Rei W (2016) An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem. Transp Res Part E: Logist Transp Rev 95:95–123
22. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
23. Taillard ED, Gambardella LM, Gendreau M, Potvin J-Y (2001) Adaptive memory programming: a unified view of metaheuristics. Eur J Oper Res 135(1):1–16
24. Solomon M, Desrosiers J, problems (1988) Time window constrained routing and scheduling. Transp Sci 22:1–12
25. http://vrp.atd-lab.inf.puc-rio.br/index.php/en/
26. https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/100-customers/

Chapter 7
Prioritizing Autonomous Supply—Comparing Selection by Marginal Analysis and Neural Nets

Gregory D. Sherman and Mitchell Mickan

Abstract When managing inventory or supply systems, it is important to make good choices about which stock to prioritize over others. We can improve the overall availability of the supplied systems by making optimal choices on which inventory items should be allocated to meet demands. In this paper, we will show how machine learning algorithms can be used to prioritize inventory. The developed algorithms were tested on a real data set and the improvement in inventory allocation measured. Machine learning is a powerful technique for transforming inputs to outputs in order to best achieve a set goal. It has many applications in areas where there is an abundance of data, and where the resulting decisions can be measured. As such inventory management is a suitable area of application, in particular the prioritization of supply. Such an approach is even more relevant to those inventory models that represent autonomous processes. The models we are interested in are those relating to system availability and that use item backorder calculations. These models rely on a traditional prioritization approach known as marginal analysis, otherwise known as a process of marginal allocation using a greedy algorithm. Because marginal analysis does not take into account performance over time, nor complex relationships in data sets, there may be potential for a machine learning algorithm to provide better results if it can learn to exploit both temporal and relationship data. The benefit of such an improvement is the value of availability generated and cost savings made in the supply network.

Keywords Inventory management · Machine learning

G. D. Sherman (✉)
Land Division, Defence Science and Technology Group, Edinburgh, Australia
e-mail: gregory.sherman@dst.defence.gov.au

M. Mickan
University of Adelaide, Adelaide, Australia

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_7

7.1 Introduction

This paper briefly reports on an exploration of various artificial neural network machine learning (ML) algorithms and applies it directly to the prioritization process that allocates stock in supply chains. This differs from many other approaches in the literature [1] that tend to apply machine learning to demand forecasting and then apply marginal analysis as a prioritization method. We hope to find when or under what conditions (given exemplar data sets) the machine learning approach of item selection performs better than marginal analysis at supporting system availability.

The inventory models we focus on are those system based spares provisioning models that attempt to maximize availability across a collection of items. Such models are commonly found in defence supply chains, the aerospace industry, health industry, disaster relief and large-scale maintenance sustainment practices. These models do this by trying to reduce the expected backorder (EBO) of items. A backorder is an order for a good or service that cannot be filled at the current time due to a lack of available supply. There is usually a delay (lead time) involved in getting this item to stock and ready supply.

We describe the types of modelling that together allow experimentation to test the efficacy of machine learning in the prioritization domain of supply. There are four parts to the modelling; the abstraction of the physical system to be modelled, the simulation of the queueing and backordering processes, the prioritizationn component and the ML algorithms.

7.1.1 The Modelling of a Simple Physical System

Figure 7.1 shows the flow of items between inventory and the supply line which is for our purposes the delivery system to the end user for consumption. The autonomous vehicle is used to ready supplies (items) for the supply line. We have a demand data set that the supply line requires. There are two parts to the problem: stocking the inventory—known methods exist (may begin with infinite supply for simplicity) and loading the autonomous vehicle (which items to load) set by a constraint (defined by data and requirements of the supply line). The latter is the one we focus on here

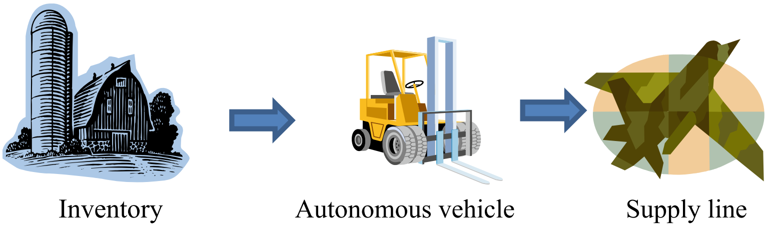


Fig. 7.1 Flow of items from inventory to supply line

but use an inventory stocking method from spares provisioning. The question should then arise about how to prioritize the selection and movement of items (in each time step) for the autonomous vehicle. The prioritization selection method could include either marginal analysis, random selection, one based upon machine learning or a combination of these.

7.1.2 Queueing and Backordering Modelling

In this section, we break down the problem to be modelled into its processing components. The models here are intended to simulate over a set discretized time period (usually 120 time steps). In each time step, there is a new set of demands and, over the entire time period, there will be a complete set of these for the time realization. However, in this model, there is expected to be at times unmet demand from previous time steps. This is because there is a constraint on how much the autonomous vehicle can move in a particular time step. Given the two sources of demand, we need to prioritize which items can be moved in the next time period by the vehicle that leads to the best performance of the system. This leads to two important calculations that need to be defined, the performance measure itself and the metric that determines selection.

An important metric here is the expected backorder which in our model is denoted EBO_i for an item i. The EBO value is calculated using the following equation:

$$EBO_i(s) = \sum_{x=s+1}^{\infty} (x - s)P\{DI = x\}, \tag{7.1}$$

where $P\{DI = x\}$ is the probability of having x orders due in. This is calculated using a Poisson probability mass function (PMF) and the value s in this case represents the number of items that should be ready for delivery [3]. To avoid summing to infinity EBO_i is approximated by summing up to the maximum order size for item *i,* with little loss of accuracy. The equation for system availability is taken from “Optimal Modelling of Inventory Systems” [2] which is given by

$$A = \prod_{i=1}^{I} \left(1 - \frac{EBO_i(s_i)}{NZ_i}\right)^{Z_i}. \tag{7.2}$$

However, due to model restrictions, we make an approximation of this availability calculation. We first assume the values of Z_i are all set to 1. This means each item has an independent failure and replacement requirements and is reasonable because we do not know how they are to be used in service. The second is by using logarithm approximation in which the equation for availability can be calculated by using a logarithmic version and converting back to avoid numerical errors (i.e.

log(1−a) = −*a*−*0.5a*^{2}$_{...}$ can be approximated by dropping terms a^2 and higher). Lastly, the value for N which represents the number of systems does not matter as it is simply a scalar multiplier. We can assume we are supporting one system. Hence, if we maximize the following simplified backorder calculation, then we are maximizing availability. An analogue for availability is the backorder equation given in the following approximation.

$$-\sum_{i=1}^{I} EBO_i(s_i). \tag{7.3}$$

We now use this expression for availability for the purposes of this paper. By using this approximation, some model accuracy is lost. However, the overall goal of the project is not to have a highly accurate model but to compare the performance of marginal analysis versus a machine learnt algorithm.

Marginal analysis is a method for decision-making that simply takes the value of an expected return and divides it by some cost to get the return per cost value ([4]—MALLOC). After applying this to all possible decisions the decision that has the highest return per cost is chosen. This is repeated until you have reached the total allowed cost. In our case, we use the availability as the value metric and substitute cost as either the weight or just the number of items (depending on the simulation being run). Reducing the dependency to only the backorder calculations focuses on the significance of any findings of improvement in a ML algorithm for this marginal analysis (that uses EBO values for items).

7.1.3 The Simulation

To accurately measure the performance of various stock allocation methods, an independent simulation had to be created. The simulation was coded in Python using additional specific packages Pandas, Numpy and Tensorflow (as well as common additional parsing modules for input and output). The simulation has three regular prioritization algorithms built into it for allocating stock.

The ordered item number algorithm was implemented first as a baseline algorithm with a simple prioritization method. This algorithm simply allocates all the unmet demand for each item in ascending order of the item numbers until all demands are met or the maximum limit is reached. This means items at the end of the list are likely to suffer starvation. The second method is to randomly select items commonly known as the "simple naïve method", however, for this paper, the results are not shown. This method and the average results of using realizations of this method are often used as a benchmark for performance against other algorithms. The third method was the one using marginal analysis for prioritizing allocation, this is important for our analysis. The last method of allocating stock was the main method used for comparison against the ML algorithms. The algorithm for calculating priority uses the following marginal benefit measure as follows:

$$\frac{EBO_i(s) - EBO_i(s+1)}{C_i * \mu_i}, \tag{7.4}$$

where for item number i, C_i represents the cost (weight) of an item, μ_i represents the mean demand of an item. There is a difference between the calculation used here and traditional marginal analysis (where more types of data are available). In our case, the marginal benefit is inversely proportional to the mean demand of an item—we had made this assumption based upon weights of items types being the same to begin with.

This algorithm was originally formulated for 'Push' supply systems however it can be adapted to work well for a 'Pull' system with a base stock assumption. In the calculations for availabilities, it was assumed that each item has a base amount of stock and when supply is reduced below this level, demand is issued. This current stock level of an item is calculated as follows:

$$Stock_i = \mu_i + EBO_i(\mu_i) - U_i, \tag{7.5}$$

where μ_i is the mean demand of item i (per time step), $EBO_i(\mu_i)$ is the expected backorders for $\mu_i = s$ in Eq. (7.1) and U_i is the current amount of unmet orders for item i. The justification for this is that it would be expected that the inventory aims to stock the mean demand amount for each item plus a safety amount. This safety amount is simply set to be the expected backorders at a mean demand stock level. Unmet orders are then subtracted as this represents the number of orders the inventory is behind for the given item. In the event where unmet orders exceed the mean plus the safety stock then a stock level of zero is assumed and any extra unmet orders beyond this is added to the $EBO_i(s)$ value.

The simulation used aggregated demand data (time step aggregated) as input to the model. The input consists off a simple CSV file that contained three columns indicating the time step, the stock index and the amount of that stock ordered. This data would be used as the demands received for the simulation. The simulation would then begin for the given time step. It would then calculate the priority of each item based on the marginal analysis benefit measure and choose to allocate the one with the highest value. This item would then be added and the state of the simulation would be updated. This would continue until either the maximum allowed load is exceeded or all current unmet demands have been met. This signals the end of the current time step and current unmet demands (aggregate unmet demands as well as individual order amounts) and met demands are logged. The availability for each item and the average across all items are calculated for these time steps and logged form part of the model output. After all the data is logged, the demands from this time step are added to the system and the process is repeated until all time steps have been completed. The whole model is based on the concept of being a 'pull' system and only meeting demands as they are received and not pre-allocating demand.

7.1.4 The Machine Learning Method

To improve upon the existing methods, machine learning was looked into to develop an alternate prioritization algorithm with the goal of achieving higher availability rates. The method of creating a model that would develop this algorithm came with many design decisions which will be detailed in this section along with their justifications.

Initially, fully connected neural networks were looked into to provide a prioritization algorithm. There were several design challenges that were necessarily addressed which included what the inputs and outputs would be and how the network would learn time dependencies.

An important consideration for the architecture was what sort of inputs would be fed into it and what outputs would be received. The output was the most important consideration as this would determine what the inputs would need to be. It was decided that the best method would be to simply output all demands to be allocated at the current time step at once. This was chosen over outputting individual priorities for each item as it was suspected that this would be more difficult for the network to learn from as the relation between this priority and the availability is more distant. In this case, the output is the demands to be met and input simply becomes the current unmet demands.

The inspiration for choosing machine learning was the hope that a model would be able to pick up relationships between demands over time that help maximize its performance. As such the architecture needs to learn according to performance over time rather than just at the moment. To address this, the initially proposed architecture was a neural network whereby the output would be the demands to be allocated at that time step. This output would then be used in calculating the new unmet demands and would feed into another neural network repetitively a variable number of times. The average of the availability at each time would be used to calculate performance.

The amount of hidden layers and their sizes as well as the number of neural networks feeding into the availability were left as hyper-parameters and tuned according to performance. This flexible architecture was not ideal as it required constructing many individual neural networks which means it would likely be quite prone to overfitting. Despite this, it served as a good method to develop an initial model to test the potential of machine learning in this problem.

To improve upon the initial model architecture, ways to reduce the need for multiple neural networks were looked into as well as methods to improve the model's ability to learn time relationships. To handle these situations, a recurrent neural network (RNN) architecture was investigated as an alternative approach (Fig. 7.2).

The architecture is similar to a neural network however it includes a component that acts as the network's memory and helps it to form time-dependent relationships. It does this by using the component to keep a record of the model's state via a variable. The recurrent neural network has both the unmet demands and the state (variable) as an input. The state is initially set to zero however over time the state changes and is repeatedly fed into the next stage of the network. The presence of a state variable

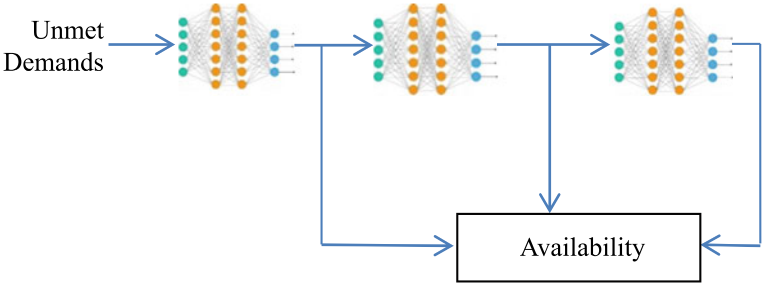


Fig. 7.2 High-Level overview of the initial neural network model architecture

greatly improves a neural networks ability to capture time dependencies which was the main inspiration between choosing to implement this type of network. To allow the model to represent more complex relationships several RNN cells were stacked on top of each other to achieve a similar effect to adding hidden layers in a traditional fully connected neural network. The actual recurrent cell in an RNN cell is what gives the network its memory and acts as the state of the network. The actual architecture of this cell can vary and there are several commonly used cells that were tested. The simplest of these cells is the basic RNN cell which simply combines the input and previous state through a sigmoid gate and propagates this through as the next state.

Regardless of architecture used a cost function would need to be decided upon that allows the network to learn. There were design decisions to consider when implementing a cost function that would best maximize performance. The most logical choice of cost function was to simply use the availability measure directly. This was the function that was used in the end, however, it has several downsides. The main one is that it is a difficult function to learn from, due to its complex calculation that includes doing many Poisson PMF calculations, as well as a large weighted sum of these values to get the EBO. Further experimentation could look into a simplified version of this function to learn from and see if it yields better results. The other downside is that due to this complexity, training speed is greatly affected. The outputs of the networks represent which demands to meet at a given time step. Therefore such outputs cannot be any given value initially and must meet certain constraints. The most obvious constraint is that they can't be negative. This is simply handled by applying a rectified linear unit to the output. Other constraints were more difficult to meet in such a way that provided optimal performance and required weighing up several design options.

There were several types of RNN's that seemed suitable for this task. These included Gated recurrent unit (GRU) and Long short-term memory (LSTM) variations with the GRU and LSTM versions being more complex and having a better ability to capture time relationships. These involved advanced cell types for which in the case of GRU enables choice in which parts of the state to update and what to reset (allowing more complex expression of the network). The LTSM cell type is

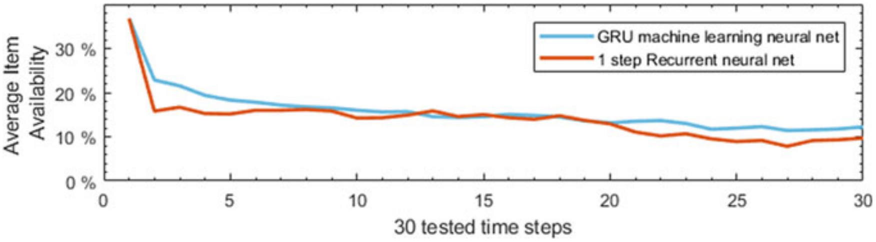


Fig. 7.3 Averaged item availability over 30 time steps: GRU versus RNN

more complex than the GRU and is designed to capture long- and short-term relationships. Experimentation found that both the GRU and LSTM were better than the traditional RNN cell which is likely due to them being more capable at capturing time series relationships. Performance between the GRU and LSTM cells was very similar with GRU slightly better and simpler to implement. Figure 7.3 shows a comparison between average item availability on two realizations over 30 tested time steps (part of the small data set in the next section) comparing the GRU cell type and the other recurrent neural net cases. Hence, it was used as the cell of choice in the network for producing results for our main experiments.

7.2 Experimentation and Results

Experiments results were calculated using the programmed simulation to allow for a consistent environment between various tests. Results were compared between those generated by various machine learning models, traditional methods and combinations of both.

Initial performance of the ML model was found to yield very poor results. After closer inspection, it was found that this was due to the algorithm only allocating approximately 40–50% of the fully allowed amount. The ML model found it difficult to learn an algorithm that allowed it to allocate large amounts of stock across all time steps while not violating the given constraints. An algorithm design decision was then made in response: the allocations suggested by the ML model were to be combined with the marginal analysis technique that it was being compared to. The ML model would indicate how to allocate stock and then based on the given constraints make a selection of the stock item types that would then employ this method in each time sample. The remaining stock items would have their allocation determined by marginal analysis.

Two different data sets were used. The smaller data set contained 100 items and 60 time steps and was split evenly between test and training sets. It totalled approximately 12,000 individual item demands across time. The larger data set contained 200 items and 261 time steps with one third being the test set and the rest being the training set. This set had approximately 106,000 individual demands.

The results for the data set are split between the performance on the training and the test data sets. The training data set is the set from which the ML algorithm learnt from which would be expected to have a better performance. The test data set is a subset of the original whole data set that was not used for training.

Generally, the difference between the performances of the ML algorithm and the marginal analysis (MA) algorithm decreased as the maximum model throughput load (this is the amount of items or weight of items that can be carried in each time step) allowed was increased. Due to the way the availability metric is calculated, a value of 100% is not achievable, because we used a pull model where we would not allocate stock until a demand is received. As such getting 100% availability was impossible as we cannot stock items in advance (throughput constraints of the problem), and also because of the influence of the adjusted Poisson probability mass function (logarithmic approximation) in determining EBO values for each item type. The maximum reachable possible availability was 33.67% for the small training set, 29.70% for the small test set, 45.68% for the larger training set and 37.09% for the larger testing set. We also display performance as a percentage of these maximum achievable availability metrics.

Performance improvements on the small data set shown in Tables 7.1 and 7.2 were very minor and the data set itself involved very sporadic demands. The lack of data in this set would likely have been a large contributor to its performance as without sufficient data ML algorithms struggle to learn anything meaningful.

The large data set had slightly better results as shown in Tables 7.3 and 7.4 compared to those resulting from the smaller data set. This supports the idea that more data is beneficial meaning similar improvements may be seen if even more data is gathered.

Table 7.1 Training data set results—small data set

Max load	Marginal analysis (MA) availability (%)	Machine learning (ML) availability (%)	Difference (ML – MA) availability (%)	Ratio of (ML/MA) availability
6000	33.19 [90.51]	33.25 [90.67]	0.06 [0.16]	1.0018
12,000	34.11 [93.02]	34.12 [93.05]	0.01 [0.03]	1.0002
18,000	34.54 [94.19]	34.54 [94.19]	0.00 [0.00]	1.0000
24,000	34.86 [95.53]	35.03 [95.53]	0.17 [0.47]	1.0049
30,000	35.06 [95.61]	35.06 [95.61]	0.00 [0.00]	1.0000

*Values in [] represent as a percentage proportion of maximum availability that is achievable based on the experimental demand

Table 7.2 Testing data set results—small data set

Max load	Marginal analysis (MA) availability (%)	Machine learning (ML) availability (%)	Difference (ML – MA) availability (%)	Ratio of (ML/MA) availability
6000	26.21 [88.25]	26.08 [87.81]	−0.13 [−0.44]	0.9950
12,000	27.46 [92.46]	27.53 [92.69]	0.07 [0.23]	1.0025
18,000	28.04 [94.41]	28.02 [94.34]	−0.02 [-0.07]	0.9993
24,000	28.47 [95.86]	28.54 [96.09]	0.07 [0.23]	1.0025
30,000	28.80 [96.67]	28.81 [97.00]	0.01 [0.03]	1.0003

Table 7.3 Training data set results—large data set

Max load	Marginal analysis (MA) availability (%)	Machine learning (ML) availability (%)	Difference (ML – MA) availability (%)	Ratio of (ML/MA) availability
6000	24.41 [53.44]	24.94 [54.6]	0.53 [1.16]	1.0217
12,000	33.00 [72.24]	33.20 [72.68]	0.20 [0.44]	1.0061
18,000	36.92 [80.82]	37.11 [81.24]	0.19 [0.42]	1.0051
24,000	39.00 [85.38]	39.04 [85.46]	0.04 [0.08]	1.0010
30,000	40.36 [88.35]	40.43 [88.51]	0.07 [0.16]	1.0017

Table 7.4 Testing data set results—large data set

Max load	Marginal analysis (MA) availability (%)	Machine learning (ML) availability (%)	Difference (ML – MA) availability (%)	Ratio of (ML/MA) availability
6000	11.90 [32.08]	11.53 [31.08]	−0.37 [−1.00]	0.9689
12,000	19.82 [53.44]	20.28 [54.68]	0.46 [1.24]	1.0232
18,000	24.44 [65.89]	24.78 [66.81]	0.34 [0.92]	1.0139
24,000	26.59 [71.69]	26.48 [71.39]	−0.11 [−0.30]	0.9959
30,000	27.93 [75.30]	28.19 [76.00]	0.26 [0.70]	1.0093

The machine learnt algorithm tended to have better performance on the training set (as shown in Tables 7.1 and 7.3) rather than the test set as expected. Test performance showed at least similar results to the marginal analysis indicating that the ML algorithm was not doing anything overly detrimental to performance. Comparing the differences between the ML algorithm and the MA algorithm, there is a very small net gain for the case of the test sets.

Generalization of the algorithm would likely improve more with larger data sets as the ML algorithm would have to learn to be more versatile to a large variety of data meaning it will adapt better to the test set.

7.3 Conclusion and Future Work

This investigation found that a minor improvement of availability could be achieved under certain circumstances by combining the ML algorithm with the traditional approach. The results showed minor improvements in performance on the training set, and on average minor improvements in the test set. This is a useful result because it shows that the ML algorithm is able to learn appropriate stock to choose given the availability function. However, this algorithm struggles to generalize its performance beyond the data it was trained on. The fact that better performance is achieved on the training data means that there was some success, as the model is able to capture complex relationships in the data and exploit them. More investigation using larger data sets or changes to the model could solve the generalization issue and allow the algorithm to perform well beyond the training data set.

The availability metric for performance was adapted to be suitable for our use case and as such may not be perfect. However, for the real-world case, it does still work as a suitable metric to determine if a machine learning approach is beneficial to stock allocation as it is directly related to expected backorders. Sherbrooke's text [2] further explains why this is a suitable measure.

Further research into how to get a ML model to deal with constraints and rewards for filling capacity could also provide gains in performance hence verify if ML is more suitable for this type of problem. Looking at the effects of larger data sets would be beneficial to continue this research and different techniques such as reinforcement learning. This may give insight into fixing bottlenecks or whether the ML performance continues to improve.

References

1. Rosienkiewicz M (2013) Artificial intelligence methods in spare parts demand forecasting. Logistics and Transport
2. Sherbrooke CC (2004) Optimal inventory modeling of systems: multi-echelon techniques (Intl Series In Operations Research & Management Science) Kluwer Academic Publishers, Norwell, MA, USA

3. Svanberg K (2009a) On spare parts optimization. KTH, Stockholm Sweden
4. Svanberg K (2009b) “On marginal allocation” (MALLOC). KTH, Stockholm Sweden

Chapter 8
Capacity Alignment Planning for a Coal Chain: A Case Study

Saman Eskandarzadeh, Thomas Kalinowski, and Hamish Waterer

Abstract We study a capacity alignment planning problem for a coal chain. Given a set of train operators, a set of train paths and a terminal comprising of a dump station and a set of routes from the dump station to the stockyard, we seek a feasible assignment of train operators to train paths, to time slots at the dump station, and to routes. The assignment must maximize the number of system paths in the resulting schedule and the schedule should perform well with respect to various performance criteria. We model the problem as a mixed-integer conic program (MICP) with multiple objectives which we solve using a hierarchical optimization procedure. In each stage of this procedure, we solve a single objective MICP. Depending upon whether we evaluate the associated performance criteria under a 2- or 1-norm, we reformulate the MICP as either a mixed-integer second-order cone program or as a mixed-integer linear program, respectively, and can streamline the hierarchical optimization procedure by exploiting properties of the model or observed behaviour on practical instances. We compare the performance of the procedure under the different norms on a real instance of the problem and find that the quality of the solutions found by the faster 1-norm procedure compares well to the solution found under the 2-norm.

Keywords Capacity planning · Coal supply chains · Mixed-integer conic programming

This research was supported by the ARC Linkage Grant LP140101000.

S. Eskandarzadeh (✉) · H. Waterer
School of Mathematical and Physical Sciences, University of Newcastle,
Callaghan, NSW 2308, Australia
e-mail: saman.eskandarzadeh@newcastle.edu.au

H. Waterer
e-mail: hamish.waterer@newcastle.edu.au

T. Kalinowski
School of Science and Technology, University of New England,
Armidale, NSW 2351, Australia
e-mail: tkalinow@une.edu.au

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_8

8.1 Introduction

Aurizon is a large rail freight operator that owns, operates and manages, the Central Queensland Coal Network, Australia's largest coal rail network. A central planning problem that they face is the alignment of the available capacity of the various components of coal chain infrastructure so as to maximize the opportunity for trains to haul coal from the mines to the ports where the coal is unloaded and stockpiled for export. The available capacity of the rail network is measured in terms of train paths, but a train path is only usable if it can be linked to loading and unloading slots at the mine and port, and the unloaded coal can then be stacked onto a stockpile. These so-called system paths are the true measure of the available capacity of a coal chain. We consider the capacity alignment planning problem in which a schedule of feasible system paths is sought that perform well with respect to various performance criteria.

Related literature includes a review by Abril et al. [1] on maximizing the number of trains which can be scheduled on a single track rail network and the work of Caprara et al. [2] on the train scheduling problem. Liu and Kozan [3] and Masoud et al. [4] also consider the optimization of the capacity of coal rail networks in the Australian setting.

Before formally defining our capacity alignment planning problem, we introduce some terminology. The Central Queensland Coal Network is comprised of four main rail corridors, each of which forms the backbone of what is referred to as a coal *system*. Each system has one or more terminals. A *terminal* is a facility where the railed coal is unloaded and stockpiled. A terminal is located at a port and is connected by rail to the rail network.

In our setting, we consider a single terminal that serves two systems. Adjacent to the terminal is a rail yard. The *rail yard* is a facility through which the trains loaded with coal must pass in order to reach the terminal from either rail corridor. Upon arriving at the yard, some trains must undergo *provisioning* operations such as servicing and refuelling before continuing to the terminal.

We define a *train path* along a rail corridor to be the 3-tuple (*system*, *departure time*, *arrival time*). There are two types of train paths: *loaded* and *empty*. Loaded train paths permit travel along the rail corridor in the direction of the terminal. The train path originates at the far end of the rail corridor at *departure time* and terminates at the terminal at *arrival time*. Empty train paths serve the opposite direction.

Loaded train paths are typically used by loaded trains hauling coal from a mine to the terminal. Upon arriving at the terminal, the loaded train enters a *dump station* and the coal is unloaded into a *pit*, where it is then transported by conveyor belt to a pad in the terminal stockyard where it is stockpiled. The path taken by the coal from the pit to the pad is termed a *route*. Typically, a dump station has several pits, not all routes are accessible from every pit, and there are restrictions on using some route combinations simultaneously. An *unload slot* is a 3-tuple (*pit*, *start time*, *finish time*). If an unload slot is assigned to a train, the train must arrive at the pit and begin unloading at *start time* and depart the pit no later than *finish time*.

We define a *system path* to be the 5-tuple (*train path*, *train operator*, *provisioning state*, *unload slot*, *route*). If a system path is used by a train, then the train must follow *train path*, be operated by *train operator*, be provisioned if *provisioning state* is true, use *unload slot*, and the unloaded coal will be transported to a pad via *route*. For a system path, *waiting time* is defined as the time difference between the end of provisioning and start time of the unload slot.

A *schedule* is a set of system paths. The problem is to find one or more schedules that are feasible with respect to a set of constraints and perform well against a set of performance criteria. In the remainder of this section, we define the problem input, the constraints and the performance criteria.

8.1.1 Problem Input

The problem input is the following:

1. A planning horizon of four weeks or equivalently 28 days.
2. A set of train operators.
3. A set of systems.
4. The provisioning times for each train operator.
5. The maximum waiting time of a train following provisioning.
6. The set of routes in the terminal. There are five routes A, B, E, G, J.
7. The set of pits at the dump station. The dump station has three pits 1, 2 and 3.
8. The pit-route access mappings which determine the routes each pit has access to. Pits 1 and 2 have access to routes A, B, E and G. Pit 3 has access to routes B, E, G and J.
9. A set of available loaded train paths.
10. The unloading times at the dump station. These are fixed at 2 h and 25 min.
11. The total demand for each week in the planning horizon for the system paths corresponding to each (*system*, *operator*) combination. A system path corresponds to the combination (*system*, *operator*) if its train path belongs to *system* and its nominated operator is *operator*.
12. The target route utilizations. These are the desired fraction of system paths corresponding to each (*system*, *operator*, *route*) combination in a schedule.

8.1.2 Constraints

The constraints or business rules of the problem are the following:

1. For each system path:

   (a) the start time of its unload slot is no earlier than the finish time of its provisioning; and

   (b) the associated pit of its unload slot has access to its route.

2. For each week in the planning horizon, the number of system paths in the schedule for each (*system*, *operator*) combination is at least equal to the loaded train path demand for that system, operator, and week. A system path is counted as a system path of a given week if the start time of its unload slot occurs in that week.
3. The schedule must respect the pit-route access mappings.
4. Each train path and unload slot can only be used in at most one system path in the schedule.
5. Since there is only single-track access to the dump stations at the terminal there must be at least 45 min separation between two consecutive arrivals. Therefore, for every two system paths in the schedule, the start times of their unload slots must be at least 45 min apart.
6. Two system paths with unload slots that overlap in time cannot have the same pit or the same route.
7. The waiting time of any system path in the schedule cannot be greater than the maximum waiting time.
8. Routing rules:

   (a) Each route can be served by at most one pit at any time.
   (b) Each pit can serve at most one route at any time.
   (c) Pits 1 and 2 cannot serve routes E and G at the same time.

8.1.3 Performance Criteria

The performance criteria for the problem are the following:

1. The number of system paths is maximized.
2. For each week of the planning horizon, the fraction of system paths for each (*system*, *operator*) combination is close to the associated contract share. The contract share corresponding to (*week*, *system*, *operator*) is defined as the ratio of the demand for system paths which belong to *week* and correspond to the (*system, operator*) combination over the total demand. Furthermore, the deviations over (*system*, *operator*) combinations should be balanced, that is, they should be as close to each other as possible.
3. The system paths for each (*system*, *operator*) combination are evenly distributed over each week of the planning horizon. The deviations over (*system*, *operator*) combinations should be balanced.
4. For each week of the planning horizon, the fraction of system paths for each (*system*, *operator*, *route*) combination is close to the desired route utilization. Furthermore, the deviations over (*system*, *operator*, *route*) combinations should be balanced.
5. The total waiting time at the yard is minimized.

6. The system paths corresponding to each (*system*, *operator*) combination allow provisioning with the given frequency. For example, with respect to departure times, every second system path from each system, for each train operator, allows for provisioning. The schedule should be balanced regarding this performance criterion in the sense that the deviations from the ideal frequency over (*system*, *operator*) combinations and all intervals are balanced. Continuing the example, the schedule should not contain two disjoint intervals, each with four system paths, in which all the system paths of one interval allow for provisioning, while all of the system paths of the other interval do not.

8.2 Problem Formulation

Without loss of generality, the time horizon is $H = [1, h] = [1, 40{,}320]$, where each time period corresponds to a minute. A system path belongs to the time horizon H if its unload start time is in H. The departure time of some of the system paths is in the day before the planning horizon starts. We represent the day before the planning horizon starts by the interval $[-1439, 0]$. The planning horizon is for scheduling all system paths with unload start times in the planning horizon. Note that all data time periods are in the range $[-1439, 40{,}320]$.

8.2.1 Parameters and Variables

We define the following parameters for the given planning horizon based on the input data:

- Y: Set of systems.
- P: Set of available loaded train paths. Each train path $p \in P$ is described by a departure time $t^d \in H$, a system, an arrival time t^a. The destination of all train paths is the yard and there exists a unique origin per each system.
- O: Set of train operators.
- R: Set of routes.
- D: Set of pits.
- $B = \{0, 1\}$: Set of provisioning states in which the `True` state is indicated by 0 and the `False` state is indicated by 1.
- U: Set of all available unload slots. Each unload slot is described by a pit, and a start time and an end time. The end time depends just on the start time. Therefore, each unload slot can be identified by its pit and its start time. For example, unload slot `(1,100)` refers to the unload slot which starts at pit 1 at time 100, and finishes 2 hours and 25 min later at time 245.
- S: Set of all valid system paths which their unload start time is in H. According to the definition, each system path, i, is described by 5-tuple $(p, o, e, u, r) \in S$.

Let H^d, H^a, H^s denote the sets of departure times, arrival times and unload start times, respectively. More formally,

$$\begin{aligned} H^d &= \{t' : \exists y \in Y \text{ and } t^a \in [-1439, 40{,}320]\ (y, t', t^a) \in P\}, \\ H^a &= \{t' : \exists y \in Y \text{ and } t^d \in [-1439, 40{,}320]\ (y, t^d, t') \in P\}, \\ H^s &= \{t' \in H : \exists \ell \in D\ (\ell, t') \in U\}. \end{aligned}$$

Let $S_1 = Y$, $S_2 = H^d$, $S_3 = H^a$, $S_4 = O$, $S_5 = B$, $S_6 = D$, $S_7 = H^s$ and $S_8 = R$. Note that $S \subseteq P \times O \times B \times U \times R$, $P \subseteq Y \times H^d \times H^a$, and $U \subseteq D \times H^s$. For a subset $J \subseteq \{1, 2, \ldots, 8\}$, we define map $S_J : \prod_{i \in J} S_i \to 2^S$ as follows:

$$S_J((s_i)_{i \in J}) = \{((s'_1, s'_2, s'_3), s'_4, s'_5, (s'_6, s'_7), s'_8) \in S : s'_j = s_j \text{ for all } j \in J\}$$

Furthermore, for $Q \subseteq \prod_{j \in J} S_j$, let $S_J(Q) = \bigcup_{s \in Q} S_J(s)$. We say that system path $i \in S$ has *property* $s \in \prod_{j \in J} S_j$ if $i \in S_J(s)$. Analogously, for $Q \subseteq \prod_{j \in J} S_j$ we say that system path i has property Q, if $i \in S_J(Q)$. For ease of exposition and readability, we define new subscripts for maps S_J. More specifically, let $S_{YOH^s} = S_{\{1,4,7\}}$, $S_P = S_{\{1,2,3\}}$, $S_U = S_{\{6,7\}}$, $S_{H^s} = S_{\{7\}}$, $S_R = S_{\{8\}}$, $S_{YOH^s} = S_{\{1,4,7\}}$, $S_{YOH^sR} = S_{\{1,4,7,8\}}$, $S_{YH^dO} = S_{\{1,2,4\}}$ and $S_{YH^dOB} = S_{\{1,2,4,5\}}$.

We denote the decision of choosing or not choosing system path $i \in S$ in a schedule by binary variable x_i, that is, $x_i = 1$ indicates that system path i is chosen and $x_i = 0$ indicates otherwise. Let $x_J(Q) = \sum_{i \in S_J(Q)} x_i$ denote the summation of the variables corresponding to system paths that have property Q. We similarly define new subscripts for maps X_J.

8.2.2 Constraints

We formulate the constraints as follows:

Constraint sets 1 *and* 7: The constraint sets 1 and 7 are implicit in the construction of the set of system paths S.

Constraint set 2: Let the demand for the system paths with property $(y, o) \in Y \times O$ in week j of the planning horizon be denoted by d_{qj}. The system path $i = (p, o, e, u = (\ell, t^s), r) \in S$ belongs to week $j \in W = \{1, 2, 3, 4\}$ if the start time of its unload slot, i.e. t^s, is in the jth week, i.e. $t^s \in I_j = [10{,}080(j-1)+1, 10{,}080j]$, where 10,080 is the number of minutes in a week and discrete interval I_j describes week j. Then

$$x_{YOH^s}(y, o, I_j) \geq d_{yoj} \qquad \text{for all } y \in Y,\ o \in O \text{ and } j \in W. \tag{8.1}$$

Variable $x_{YOW}(y, o, I_j)$ which denotes $x_{\{1,4,7\}}(\{y\} \times \{o\} \times I_j)$ is the number of system paths in the given schedule $x = (x_i)_{i \in S}$ with property (y, o), where their unload slots $u = (\ell, t^s) \in U$ belong to week j, i.e. $t^s \in I_j$.

Constraint sets 3 *and* 6: Let $A_t(r), r \in R, t \in H$ be the set of all system paths $i \in S_R(r)$ such that their unload slots overlap time $t \in H$. For each $r \in R$, we denote the collection of all distinct sets $A_t(r)$ over all $t \in H$ by $\mathcal{S}_0(r)$. To define it more formally, we have:

$$A_t(r) = \{(p', o', e', u', r') \in S_R(r) : u' = ((\ell, p), t'), t \in [t', t' + 144]\},$$
$$\mathcal{S}_0(r) = \{A_t(r) : t \in H, \forall t' \in H\ A_t(r) \not\subset A_{t'}(r), |A_t(r)| > 1\}.$$

Similarly, let $D_t(\ell), \ell \in D$ be the set of all system paths $i \in S_D(\ell)$ such that their unload slots overlap time $t \in H$. For each $\ell \in D$, we denote the collection of all distinct sets $D_t(\ell)$ over all $t \in H$ by $S_1(l)$. Let $B_t, t \in H$ be the set of all system paths $i \in S$ such that

(a) their pits are either pit `1` or `2`;
(b) their routes are either route `E` or `G`; and
(c) their unload slots overlap time $t \in H$.

We denote the collection of all distinct such sets B_t over interval H by $\mathcal{S}_2$. Or more formally,

$$B_t = \{(p', o', e', u', r') \in S : r' \in \{\mathtt{E}, \mathtt{G}\}, u' = ((\ell', p), t'), (\ell', p) \in \{\mathtt{1}, \mathtt{2}\}, t \in [t', t' + 144]\},$$
$$\mathcal{S}_2 = \{B_t : t \in H, \forall t' \in H\ B_t \not\subset B_{t'}, |B_t| > 1\}.$$

Then

$$\sum_{i \in Q} x_i \leq 1 \qquad \text{for all } r \in R, Q \in \mathcal{S}_0(r), \tag{8.2}$$

$$\sum_{i \in Q} x_i \leq 1 \qquad \text{for all } l \in D, Q \in S_1(l), \tag{8.3}$$

$$\sum_{i \in Q} x_i \leq 1 \qquad \text{for all } Q \in \mathcal{S}_2. \tag{8.4}$$

Constraint set 4:

$$\sum_{i \in S_P(p)} x_i \leq 1 \qquad \text{for all } p \in P, \tag{8.5}$$

$$\sum_{i \in S_U(u)} x_i \leq 1 \qquad \text{for all } u \in U, \tag{8.6}$$

Constraint set 5: Constraint 5 is equivalent to this constraint that in a feasible schedule in every 45 min interval in the planning horizon, at most one unload slot starts. There are 40,276 45 min intervals in a planning horizon and the last interval starts at time 40,276. More formally, for $t \in [1, 40{,}276]$, let $S^{>45}(t) = S_W([t, t+44])$ and let $\mathcal{S}^{>45} = \{S^{>45}(t) : t \in [1, 40{,}276]\}$, then

$$\sum_{i \in Q} x_i \leq 1 \qquad \text{for all } Q \in \mathcal{S}^{>45}. \tag{8.7}$$

8.2.3 Performance Criteria

We formulate the performance criteria as follows:
Criterion 1: The number of system paths is equal to $\sum_{i \in S} x_i$. Let $z^{(1)} = -\sum_{i \in S} x_i$. The goal is to maximize this measure.
Criterion 2: Let d_j denote the total demand over week $j \in W$. The associated contract share for (*system*, *operator*) combination $(y, o) \in Y \times O$ in week $j \in W$ is equal to d_{yoj}/d_j. We want the share of system paths with property (y, o) in week j, i.e. $x_{YOH^s}(y, o, I_j)/x_{H^s}(I_j)$ is as close as possible to the associated contract share or equivalently $x_{YOH^s}(y, o, I_j)$ is as close as possible to $x_{H^s}(I_j)d_{qj}/d_j$. Therefore, the associated deviation vector is $z^{(2)} = (z^{(2)}_{yoj})_{(y,o,j) \in Y \times O \times W}$ where

$$z^{(2)}_{yoj} = x_{YOH^s}(y, o, I_j) - d_{qj}/d_j x_{H^s}(I_j) \tag{8.8}$$

for all $(y, o, j) \in Y \times O \times W$.
Criterion 3: Assume we have n system paths in week j. One plausible interpretation of performance criteria 3 is that the number of system paths in each day to be as close as possible to $n/7$. The deviation vector is $z^{(3)} = (z^{(3)}_{yoji})_{(y,o,j,i) \in Y \times O \times W \times [1,7]}$ where

$$z^{(3)}_{yoji} = x_{YOH^s}(y, o, b_{ji}) - x_{YOH^s}(y, o, I_j)/7 \tag{8.9}$$

for all $(y, o, b_{ji}) \in Y \times O \times W \times [1, 7]$, and $b_{ji} = [10{,}080(j-1) + 1440(i-1) + 1, 10{,}080(j-1) + 1440i]$ describes day $i \in [1, 7]$ of week j.
Criterion 4: Let $f_{yoj}(r)$ be the desired route $r \in R$ utilization by system paths with property $(y, o) \in Y \times O$ (i.e. system paths which are from system y and are operated by operator o) in week $j \in W$, where $f_{yoj}(r)$ is a real number between zero and one. In other word, $100 \times f_{yoj}(r)$ per cent of total number of system paths in week j from system y which are operated by operator o, are assigned to route r. The deviation vector is $z^{(4)} = (z^{(4)}_{yojr})_{(y,o,j,r) \in Y \times O \times W \times R}$, where

$$z^{(4)}_{yojr} = x_{YOH^sR}\left(y, o, I_j, r\right) - f_{yoj}(r) x_{YOH^s}(y, o, I_j) \tag{8.10}$$

for all $(y, o, j, r) \in Y \times O \times W \times R$.
Criterion 5: Let w_i denote the waiting time for system path $i \in S$, then the total waiting time is equal to $z^{(5)} = \sum_{i \in S} w_i x_i$. The goal is to minimize this measure.

Criterion 6: Let $g(y, o, I)$ be the desired fraction of system paths with property (y, o) which their departure times are in the interval $I \subseteq H_1 = [-1439, 40{,}320]$ and they are allowed to be provisioned. One suitable choice for $g(y, o, I)$ is to be defined as equal to $1/k$. Let $H^d_y = \{t' \in H_1 : \exists t^a \in [-1439, 40{,}320] p = (y, t', t^a) \in P\}$ includes all departure times of system paths from system y. Let $H'^d_y = \{[t_1, t_2] : t_1, t_2 \in H^d_y, t_1 \leq t_2\}$. We define the deviation vector as $z^{(6)} = (z^{(6)}_{yoI})_{(y,o,I)\in Y\times O\times H'^d_y}$, where

$$z^{(6)}_{yoI} = x_{YH^dOB}(y, I, o, 1) - g(q, I)x_{YH^dO}(y, I, o) \tag{8.11}$$

for all $(y, o, I) \in Y \times O \times H'^d_y$. One of the main improvements one can make to make the above measure less computationally expensive is to reduce the number of values which set I can take. If we know which intervals contain exactly k system paths in the optimal solution, then we just need to consider those intervals. However, since we do not know the optimal schedule before solving the model, we need to consider intervals of all lengths which can contain k consecutive system paths in any possible optimal schedule.

The deviations associated with Criteria 2–4 and 6 are vectors. In order to measure these criteria, we need to formalize notions of *deviation* and *unbalancedness*. Let $z = (z_i)_{i\in n}$ denote such a deviation vector. Then each element z_i is the deviation of component i of this criterion from some target value. We define the deviation of z to be $D(z) = \|z\|_p$ and the unbalancedness of z to be $B(z) = \min_{z_0\in\mathbb{R}}\{\|z - z_0e\|_p\}$ where $p \in \{1, 2\}$ and $e = (1, \dots, 1)^T$. Thus, the length of the vector z is the measure of deviation and the shortest distance between z and a point on the line $z_1 = z_2 = \cdots o = z_n$ is the measure of unbalancedness. For each of these criteria, we wish to minimize both the deviation and the unbalancedness.

8.2.4 Model Formulation

The aforementioned constraints and performance criteria gives rise to the following mixed-integer conic program (MICP) with multiple objectives:

$$\min \left(z^{(1)}, z^{(2D)}, z^{(2B)}, z^{(3D)}, z^{(3B)}, z^{(4D)}, z^{(4B)}, z^{(5)}, z^{(6D)}, z^{(6B)}\right)$$
$$= \begin{pmatrix} z^{(1)}, D(z^{(2)}), B(z^{(2)}), D(z^{(3)}), B(z^{(3)}), \\ D(z^{(4)}), B(z^{(4)}), z^{(5)}, D(z^{(6)}), B(z^{(6)}) \end{pmatrix}$$
$$\text{subject to (8.1)–(8.11)}, x_i \in \{0, 1\} \text{ for all } i \in S.$$

We now make several observations about this model that we will use to our advantage when solving it. For Criteria 2–4 and 6, we note that for any deviation vector z, the optimal value for z_0 is the average of the components of z under the 2-norm, the median of the components of z under the 1-norm, and that $B(z) \leq D(z)$ for either norm. Furthermore, for a feasible deviation vector, the sum of the components of z is equal to zero. Thus, under the 2-norm, $B(z) = D(z)$ and so a minimum deviation solution also minimizes unbalancedness. Consequently, we can omit the objective functions $z^{(\cdot B)}$ under the 2-norm.

8.3 Solution Methodology

To solve the multi-objective MICP described in the previous section, we employ a hierarchical optimization procedure in which we solve successive single-objective MICPs. The order in which the criteria were presented reflects their relative importance to Aurizon and minimizing deviation is more important than minimizing unbalancedness.

To begin the hierarchical optimization, we optimize the MICP with respect to the first objective function $z^{(1)}$. Let $z_*^{(1)}$ denote the value of the best integer solution found and suppose that in a solution to the multiple objective MICP we require that the value of $z^{(1)}$ degrades by a factor of at most $a^{(1)}$, where $a^{(1)} \geq 0$. We refer to $a^{(1)}$ as the degradation factor and add the threshold constraint $z^{(1)} \leq z_*^{(1)}(1 + a^{(1)})$ to the current MICP. In the next stage, we optimize the current MICP with respect to the second objective function $z^{(2D)}$ and then add the corresponding threshold constraint $z^{(2)} \leq z_*^{(2)}(1 + a^{(2)})$ to the current MICP. The above process is repeated until all objective functions $z^{(i)}$ for $i \in \{1, 2D, 2B, 3D, 3B, 4D, 4B, 5, 6D, 6B\}$ have been considered.

Each single-objective MICP can be reformulated as a mixed-integer second-order cone program under the 2-norm, and it can be reformulated as a mixed-integer linear program under the 1-norm.

8.4 Computational Investigation

In this section, we investigate the performance of the hierarchical optimization procedure to solving the multi-objective MICP on an instance of realistic size. The investigation is carried out on a machine with dual oct core 3.33 GHz Intel Xeon E5-2667 v2 processors and 256 GB of RAM. The number of threads used is 13 out of available 16 threads. We use Gurobi v8.0.0 via the Python API and Python v3.6.4. The time limit for solving each single objective MICP is 600 s. The instance has the following characteristics:

- The maximum waiting time is 60 min.
- There are two train operators named `op1` and `op2` and two systems named `s1` and `s2`.
- The demand for each week is given in Table 8.1.
- The unload slots are generated with consideration of constraint set 5 and the implication of constraint set 6 for unload slots. The maximum number of unload

Table 8.1 Weekly demand

System	Train operator	Demand
s1	op1	98
s1	op2	21
s2	op1	21
s2	op2	0

slots which can be used in each day is 27. Note that the constraint set 5 is implicit in the construction of unload slots and is therefore not coded.

- The departure times of train paths from systems s1 and s2 are 20 min and 90 min apart, respectively. Some of these train paths are cancelled due to maintenance activities.
- The desired provisioning frequency for system paths from system s2 is one per two and for system paths from system s1 is one per one (i.e. we prefer all system paths from system s1 to allow provisioning).
- The route utilization ratios are shown in Table 8.2. The route utilization ratios for system s2 and train operator op2 are zero.

The performance statistics under each norm are shown in Tables 8.3 and 8.4. Each row in the table corresponds to a single-objective MICP and we use the following notation:

- Relax: the objective function value of the initial MICP relaxation.
- Root: the objective function value of the final MICP relaxation solved at the root node of the branch-and-bound tree.
- BestBnd: the objective function value of the best MICP relaxation found during the branch-and-bound search.
- BestFeas: the objective function value of the best feasible solution found during the branch-and-bound search.
- Gap: the optimality gap of the best feasible solution which we define to be $|\text{BestFeas} - \text{BestBnd}|/\text{BestFeas}$.

Under the 2-norm, Gurobi finds an optimal solution to each single objective MICP associated with Criteria 1 and 2, and finds a solution within 1% of the optimal value for 5. However, it cannot find good solutions for the other criteria within 600 s. It is not that surprising that the 1-norm leads to much better performance than what was observed under the 2-norm. Gurobi solves the single-objective MICPs associated with Criteria 1–3 and 5 to optimality. The quality of the solutions is much better for Criteria 4 and 6, much worse for Criterion 5, and about the same for the other criteria. The reason that the solution for Criterion 5 under the 1-norm is much worse than the solution under the 2-norm is that the threshold constraint associated with Criterion 4 under the 2-norm is not restrictive compared to that under the 1-norm. In Table 8.3, the value of the best integer solution found under Criterion 4 is 10,090 which is likely to be far from optimal.

Table 8.2 Weekly route utilization

System	Train operator	Route	Route utilization
s1	op1	A	19/98
		B	22/98
		E	15/98
		G	34/98
		J	8/98
s1	op2	A	3/21
		B	2/21
		E	0
		G	2/21
		J	14/21
s2	op1	A	13/21
		B	5/21
		E	0
		G	0
		J	3/21

Table 8.3 Performance statistics under the 2-norm

Objective	Degrad (%)	Relax	Root	BestBnd	BestFeas	Gap (%)	Time (s)
1	0	756	–	756	756	0	<1
2D	10	0	0	2.5	2.5	0	49
3D	10	0	0	0	3.4	100	600
4D	10	–	–	0	10,090	100	600
5	10	5406	5618	5640	5698	1	600
6D	–	–	–	0	4204	100	600

One interesting observation is that the best integer solution values of the deviation and unbalancedness objectives associated with Criteria 2–4 and 6 are equal. This implies that the median of the components of each deviation vector is zero and that the minimum deviation solution also minimizes unbalancedness. We believe that this is an artefact of the instance rather than the model and so, in general, we cannot omit the unbalancedness objectives as we did under the 2-norm. However, we have reason to believe that this could be a common occurrence when solving practical instances and so revised the hierarchical optimization procedure to skip minimizing unbalancedness if the median of the deviation solution is zero.

Table 8.4 Performance statistics under the 1-norm

Objective	Degrad (%)	Relax	Root	BestBnd	BestFeas	Gap (%)	Time (s)
1	0	756	–	756	756	0	<1
2D	10	0	0	5.2	5.2	0	26
2B	10	0	0	5.2	5.2	0	30
3D	10	0	–	0	0	0	4
3B	10	0	–	0	0	0	3
4D	10	0	0	12	12	0	162
4B	10	0	0.3	9.8	12	18	600
5	10	5986	6025	6085	6088	0.05	600
6D	10	1153	1303	1399	1516	8	600
6B	–	1153	1165	1170	1516	22	600

Table 8.5 Revised performance statistics under the 1-norm

Objective	Degrad (%)	Relax	Root	BestBnd	BestFeas	Gap (%)	Time (s)
1	0	756	–	756	756	0	<1
2D	10	0	0	5.2	5.2	0	18
3D	10	0	0	0	0	0	4
4D	10	0	0	12	12	0	108
5	10	5987	6025	6085	6088	0.05	220
6D	–	1218	1391	1483	1512	2	600

Table 8.6 Comparison under the 2-norm of the deviations of the final multiple objective MICP solutions

Norm	1	2	3	4	5	6
2-norm	756	2.7	3.4	6441	5698	4204
1-norm	756	2.9	0	4.5	6688	1591
Revised 1-norm	756	2.9	0	5.5	6688	1610

The performance statistics of the revised procedure are shown in Tables 8.5 and 8.6 compares for each criterion, the deviation of the final multiple objective MICP solutions when evaluated under the 2-norm. Overall the solutions found under the 1-norm are better quality and can be found comparatively quickly using the revised hierarchical optimization procedure.

8.5 Conclusion

We consider a capacity alignment planning problem for a coal chain that is faced by our industry partner Aurizon in which a schedule of system paths are sought that perform well with respect to various performance criteria. For many of these criteria, the schedule should not only minimize the deviation from some prescribed targets but also the deviations should be well balanced. We model the problem as a mixed-integer conic program (MICP) with multiple objectives which we then solve using a hierarchical optimization procedure. In each stage of this procedure, a single-objective MICP must be solved. Depending upon whether we evaluate the associated performance criteria under a 2- or 1-norm, we reformulate the problem as either a mixed-integer second-order cone program or as a mixed-integer linear program, respectively.

A property of the model is that a minimum deviation solution for a given criterion measured under the 2-norm also minimizes the unbalancedness for that criteria. While this is not a property of the model under the 1-norm, we believe that it will frequently be the case when solving practical instances. Consequently, we revised the hierarchical optimization procedure to omit the unbalancedness objectives under the 2-norm, and check for their redundancy under the 1-norm.

A computational investigation on a real instance of the problem reveals, not unsurprisingly, that the hierarchical optimization procedure under the 1-norm finds good solutions much faster than under the 2-norm. Moreover, the quality of the solutions found by the procedure under the 1-norm compares well to the solution found under the 2-norm when the 1-norm solutions are evaluated for each criterion using the 2-norm.

Future work will include improved modelling of bottleneck performance criteria such as Criterion 6, improved solution procedures such as customized branching for the single-objective MICPs within the hierarchical optimization procedure, and extending the problem considered to include additional practical considerations such as dynamic start times for unload slots.

References

1. Abril M, Barber F, Ingolotti L, Salido MA, Tormos P, Lova A (2008) An assessment of railway capacity. Transp Res Part E: Logist Transp Rev 44(5):774–806
2. Caprara A, Fischetti M, Toth P (2002) Modeling and solving the train timetabling problem. Oper Res 50(5):851–861
3. Liu SQ, Kozan E (2011) Optimising a coal rail network under capacity constraints. Flex Serv Manuf J 23(2):90–110
4. Masoud M, Kozan E, Kent G, Liu SQ (2017) A new constraint programming approach for optimising a coal rail system. Optim Lett 11(4):725–738

Chapter 9
Situational Awareness for Industrial Operations

Peter Baumgartner and Patrik Haslum

Abstract The smooth operation of industrial or business enterprises rests on constantly monitoring, evaluating and projecting their current state into the near future. Such *situational awareness* problems are not well supported by today's software solutions, which often lack higher level analytic capabilities. To address these issues, we propose a modular and re-usable system architecture for monitoring systems in terms of their state evolution. As a main novelty, states are represented explicitly and are amenable to external analysis. Moreover, different state trajectories can be derived and analysed simultaneously, for dealing with incomplete or noisy input data. In the paper, we describe the system architecture and our implementation of a core component, the state inference engine, through a shallow embedding in Scala. The implementation of our modelling language as an embedded domain-specific language grants the modeller expressive power and flexibility, yet allows us to abstract a significant part of the complexity of the model's execution into the common inference engine core.

Keywords Industrial operations modelling · Situational awareness · Supply chains

9.1 Introduction

The smooth operation of industrial or business enterprises like supply chains, assembly lines or warehouses critically depends on maintaining situational awareness. In our context, *situational awareness* is the problem of gathering changes in the opera-

This research is supported by the science and industry endowment fund.

P. Baumgartner (✉) · P. Haslum
CSIRO/Data61, Australian National University, Canberra, Australia
e-mail: Peter.Baumgartner@data61.csiro.au

P. Haslum
e-mail: Patrik.Haslum@data61.csiro.au

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_9

tion's environment, aggregating them for deriving the current state of the operation at a semantically high level, projecting the current state into the near future, alerting the user of potential problems and proposing corrective action if required.

Situational awareness in this sense currently lacks software tool support. While database systems and ERPs can help human operators gain situational awareness, they do not offer a complete solution with integrated analytic capabilities. In this paper, we address this issue by proposing an architecture for a novel situational awareness software platform.

This architecture has its origin in a situational awareness and scheduling system for the factory floor of an industrial client, which we generalize to be applicable in a variety of domains, ranging from local operations, such as factory floor assembly lines or warehouse management, to large-scale and distributed operations, such as national or international supply chains or the Array of Things [2]. We also describe a concrete realization of one core component, the state inference engine.

The state inference engine maintains a current state which is updated whenever an external event comes in. In the current implementation, state update is described in terms of rules that are triggered by such events. It offers a modelling paradigm that assigns rule sets to (parallel) processes, one process for each entity of interest, and message channels connecting these processes. The execution of the processes' rules enables inference about aspects of the state that are not directly observable, including disjunctive reasoning about alternative states and execution histories.

While the model is domain-dependent, the interpreter is universal. It is, in essence, a forward-chaining rule engine and is realized via a shallow embedding in Scala. The main advantage of this design is that it gives access to the full power of the Scala language to represent process states and express conditions, functions and transformations on them. It provides expressive power and flexibility, at little implementation cost. It is this modelling language and inference engine implementation that we put forward as the main contribution of this paper. Due to modularity, our architecture allows for transparently substituting another method of state inference, e.g. conflict-directed diagnosis [10], with its associated modelling formalism.

We emphasize that we do not intend to replace existing systems, e.g. ERPs or tailored scheduling systems. Instead, we want to capitalize on their functionality by integrating and augmenting them with inference capabilities for the purpose of gaining a more holistic view. An important feature of our system is the explicit and externally analysable state representation, enabling it to simultaneously represent and explore different state trajectories, for example, to assess their plausibilities and what-if reasoning. This ability is essential to deal with noisy or unreliable data sources.

9.2 Related Work

Supply chains have become complex networks with automated transitions to manage the manufacturing and flow of goods. Techniques like service-oriented architectures (SOAs) and business process modelling (BPM) [24] are instrumental for automating and optimizing supply chains with respect to life cycle management, procurement, logistics and order management. BPM comprises methods for making business operations explicit, such as BPMN [20], which is intended mainly for manual use. Such explicit models are typically not designed from the ground up for situational awareness, but they may well inform models for situational awareness. Software designs like SOAs or model-driven architectures do not lead to situational awareness by themselves but may integrate situational awareness capabilities.

The basic principle of Industry 4.0 is to connect machines, workpieces and business management systems in intelligent networks for controlling each other autonomously. Examples are machines that can predict failures and trigger maintenance processes autonomously, or self-organized logistics which react to unexpected changes in production [15]. Companies like Oracle and SAP have proposed to go further and leverage the new technologies for *situational awareness*. The underlying observation is that automation is not enough, the overall system should be conscious and knowledgeable of its surroundings [19]. Indeed, some commercial "business intelligence" information systems [21] offer event management functionality that allows users to monitor and measure supply chain activities. However, existing systems lack the capability to perform deeper inference, for instance, about unobserved events or completion of missing data.

Derigent and Thomas [9] expressed the need for situational awareness in the context of the Internet of Things (IoT). They propose a corresponding architecture and identify key functionalities. Similar proposals were made by Lee, Ardakani, Yang and Bagheri [17], and by Singh and Tripathi [23]. They all remain somewhat inconcrete regarding the realization of their proposals. Ghimire, Luis-Ferreira, Nodehi and Jardim-Goncalves [11] go further and mention a situational awareness module that makes use of formal knowledge representation and semantic web technologies such as OWL (Ontology Web Language). From their description, however, it seems that the module is currently under implementation and no instantiation of their architecture is yet available.

One of the key uses of formalized business models is *monitoring* their execution for conformance to the model. Cook and Wolf [7] developed techniques for uncovering and measuring the discrepancies between models and executions. They utilize rather high-level metrics for that. In contrast, and closer to our work, Chesani, Mello, Montali, Riguzzi, Sebastianis and Storari [5] propose a framework for performing conformance checking of process execution traces w.r.t. expressive reactive business rules. Rules are mapped to Logic Programming, using Prolog to classify execution traces as compliant/non-compliant. Our approach is rule-based as well but additionally offers processes and channels as a modelling paradigm. DeGiacomo, Maggi, Marella and Sardina [8] also address conformance checking, by modelling process rules in the declarative Planning domain definition language (PDDL) and applying an off-the-shelf automated planner. They also attempt to identify the missing or superfluous activities in non-conformant traces, which is a special case of the general problem of diagnosis of discrete dynamical systems [6, 18].

Formal modelling languages, such as Petri nets or the Promela language embodied in the Spin model checker [13], are used for modelling business processes or work-flows as well as concurrent processes in (embedded) computer systems. However, these languages do not support modelling processes whose internal state or messages are made up of complex data types.

In the runtime verification area, Kauffman, Havelund and Joshi [16] propose a more specialized Scala DSL for monitoring event streams over Allen's temporal interval logic [1]. For a different application, Havelund and Joshi [12] propose a Scala DSL based on hierarchical state machines. These approaches overlap with our approach in terms of implementation (Scala DSL) but differ conceptually. In particular, our approach supports explicit candidate model computation via disjunctive rules and backtracking.

Outline of the paper. The rest of this paper is structured as follows: in Sect. 9.3 we present an overview of our system architecture. This is needed to provide context for our process modelling language and how it is executed. This is explained in Sect. 9.4. We conclude in Sect. 9.5 with a brief summary and future work.

9.3 System Architecture

The system architecture is depicted in Fig. 9.1. Its components may run in parallel, and may be distributed across sites. We use the term "message" for the information flowing between the components.

Reports, sensor output and message handling The situational awareness system is driven by and reacts to external events. These may come from sensors, such as RFID and video (e.g. on a factory floor) or they may be reports filed by humans (e.g. orders and dockets in a sales business).

We do not assume that the environment provides complete information, e.g. sensors can fail and workers forget to report or report incorrectly. However, we assume an abstraction mechanism, the "Message Handling" which preprocesses input mes-

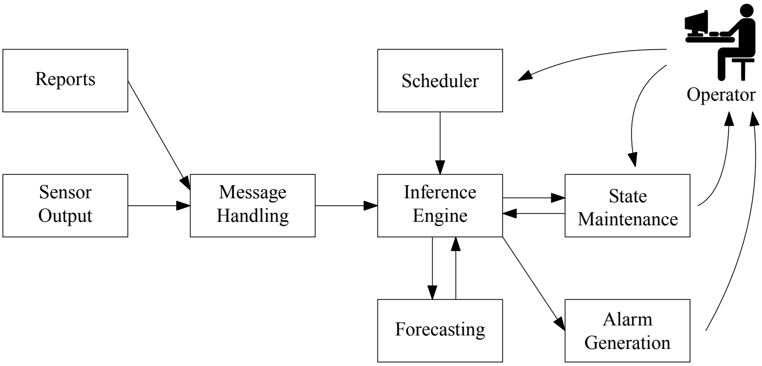


Fig. 9.1 System architecture

sages into a uniform message format. It may also carry out other operations, such as adding a timestamp, sorting or filtering.

Inference engine In addition to sensor and report input being unreliable, one cannot expect a complete model of the world under consideration. It is the task of the inference engine to use the information at a given time to derive a more complete description of the current overall state. This may involve inferring missing information, what-if reasoning and auto-correction.

The inference engine is general and needs to be equipped with a concrete model. In our current realization, the model is given by processes connected via message channels. Execution is stateful, with user-supplied rules governing state transitions. (See Sect. 9.4 for more details.) The inference engine receives a windowed sequence of message, and, as a result, derives one or more plausible current states. Analysing this state or states in context with the state history is the responsibility of the state maintenance module. Figure 9.2 gives an overview of these dynamic aspects of the inference engine and the state maintenance module.

State maintenance The state maintenance module manages a set of *execution paths*, or just *paths*. A path is a time-finite sequence of states obtained by successive runs of the inference engine. The set of execution paths naturally forms an *execution tree*, cf. again Fig. 9.2. The execution tree is meant to represent possible executions of the modelled system. The state maintenance constructs the execution tree based on new states coming in from the inference engine, and it informs the inference engine by telling it the next state to continue with. We emphasize that this next state does not have to be among the input states, it could be any state derived from the execution tree so far.

The full generality of execution trees may not be needed in all applications. Some immediate special cases and variations come to mind:

- *Markovian*: no histories; every branch is a singleton, the most recent state.
- *Deterministic*: the tree has only one branch.
- *Focused*: only some branches, e.g. the most plausible states, are kept.

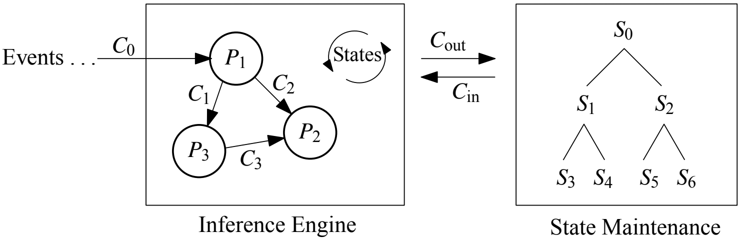


Fig. 9.2 Inference engine and state maintenance. Events are coming in from the channel C_0. The inference engine has processes P_1, P_2, P_3 and internal channels C_1, C_2, C_3. The channels C_{in} and C_{out} connect the two components for exchanging states. The state maintenance module maintains possible execution paths in terms of states derived by the inference engine

- *Probabilistic*: assign each state a probability distribution over its children.

While state maintenance could in principle be done by the inference engine, there are arguments to keep it separate. Because the nodes along a path are temporally ordered, execution trees can be analysed with temporal logic. One would express (un)desirable properties with logics such as linear temporal logic (LTL), computational tree logic (CTL) and apply verification methods to them. (See [3] for a textbook on temporal verification.) Particularly relevant are runtime verification methods capable of handling structured data like those developed by Havelund and Peled [14]. For instance, one could formulate a temporal LTL safety constraint for monitoring travel times between waypoints for fresh goods and raise an alarm if violated. Also, techniques for log analysis seem applicable. For instance, Brandt et. al. [4] propose an expressive ontology-based framework on top of metric temporal logic.

Scheduler The scheduler provides the timeline for future events so that a mission can be accomplished. In a supply chain domain, it could be route planning, among others (we consider the term "Scheduler" loosely).

Forecasting The components described so far are backward-looking in time. The forecasting module is concerned with projecting the current state into the near or medium future, and thus requires additional information, e.g. the current schedule stored in a database. Even a simple approximate prediction can be useful in practice. In a supply chain domain, for instance, forecasting may combine current vehicle locations with the routes yet to be travelled.

Alarm generation Generating alarms or notifying the user of deviations of the expected state is a core functionality of situational awareness systems. In our current implementation, this functionality is provided by the inference engine, which is poised for doing this as it has at its disposal (a) the current state, (b) the current schedule and (c) the forecast. Incoherencies between (a), (b) and (c) are thus detectable. Conceptually, however, alarm generation is a different activity from state inference, as it involves a judgement about whether the state merits human attention, and not only what the state is.

Operator The operator receives notifications from the system, in particular alarms, and interacts with the state maintenance system to achieve faithful state representation. The operator invokes scheduling when needed, e.g. in case of alarms.

9.4 The Process Modelling Language

The main rationale behind our modelling language is to provide a natural framework for mapping actors in the real world—physical or conceptual ones, like schedules—to corresponding modelling entities. Our modelling approach supports object-oriented design principles like abstraction, polymorphism, encapsulation and inheritance. The main entities are *processes*, which run (conceptually) in parallel and exchange information through *messages*. The message-passing paradigm was inspired in part by the

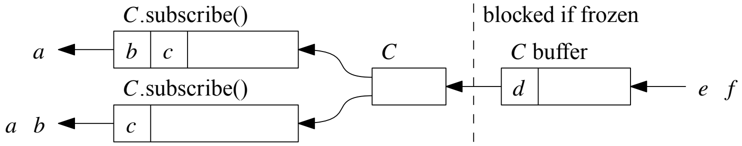


Fig. 9.3 The channel subscription model. In this example, an external channel C has already received data items a, b and c. Then C was frozen and incoming data d was put into the buffer for now, while e and f have not arrived yet. The data a, b and c already received is actually copied and forwarded to all subscribing channels. In the example, the upper subscribing channel has a being read from, and the lower subscribing channel has a and b being read from

Spin model checker and its modelling language Promela [13]. Spin is geared towards model checking, i.e. the problem of proving or disproving that all possible runs of the system satisfy some property. Unlike Spin, our system is tailored towards situational awareness, which analyses *one* run (the “reality”) and it supports object-oriented design principles.

Processes *Processes* are models of the entities of the system at hand. The entities could be physical entities, such as machines producing goods and workers scheduled for work, or abstract entities, such as schedules and rosters. Processes can be created and scheduled for execution dynamically, and they can be stopped and destroyed dynamically. Processes are stateful, where a process’ *state* is given by a user-declarable set of local variables and their current values. It is the collection of the states of all processes that is sent to the state maintenance component at certain times.

The computation inside a process is described by a set of *rules*. A rule is an if-then statement whose condition may involve reading a channel. If a message is available at the time and the rule’s condition is met then the rule is *executable*. Otherwise, the rule *fails*. The conclusion of a rule typically modifies local variables and/or sends data to other processes. The conclusion can be disjunctive, which allows for branching into alternative possible states that may be consistent with the current, incomplete state.

Messages and channels *Channels* are used for sending *messages* into and out of processes. Messages can be of any type as long as they can be serialized. We distinguish between *internal* and *external* channels. Internal channels connect processes, while external channels connect processes with the outside world.[1] External channels can be *frozen*, which means that incoming data is deferred into a buffer until the channel is unfrozen again. Channels are lossless and can be used for m-to-n communication. Any number of processes can *subscribe* to a channel. This has the effect that incoming messages are duplicated to all subscribers so that multiple processes can receive and deplete the channel individually without interfering with each other, cf. Fig. 9.3.

[1] External channels must be equipped with deserialization for their message type.

Inference engine The inference engine executes the models written in our language. This is done in rounds of scheduling and running its processes. At the beginning of each round, all external channels are frozen, so that incoming messages are temporarily buffered, cf. Fig. 9.3. A scheduler then selects in a fair way the next process for execution among the scheduled processes. This process is executed by trying its rules, once, in the given order. A failed rule application has no effect, i.e. it never modifies a state or consumes channel messages. The conclusion of the first executable rule is executed. If the conclusion is a disjunction, the first alternative is executed and the second alternative is put away together with the current state, and later restored for execution. That is, disjunctions require maintaining a tree of execution sequences over saved states and channel data. The execution of a conclusion (case) may also explicitly fail.[2] In this case, process and channel states are restored to what they were before.

In each round, this selection of processes is repeated until all channels are depleted or a user-defined cutoff number of rule executions has been reached. This results in one or more derived states. These are sent, via the dedicated external channel, to the *state maintenance* module, whose task is to derive from them some state that is sent back to the inference engine as the new current state for the next round. Finally, the external channels are unfrozen so that messages arrived in the meantime become available, and the next round starts.

9.4.1 Illustrative Example

As an example, we consider a highly simplified food supply chain, which we model the supply as a set of interacting processes. A process may correspond to an actor in the supply chain (such as a distributor), or an actor may be modelled as several interacting processes (for example, a distributor may be decomposed into inventory, transportation and contracting processes). It is also possible for processes to model interfaces between actors (such as a shipment). Processes act according to their internal logic, which is codified in the model, and interact via message channels. The channel mechanism is necessary to model the synchronization of processes (e.g. between a delivery and an inventory process).

Processes There are *goods* (apples and oranges) of specific *origin* (Riverina and Batlow) which are transported between *warehouses* (Sydney, Goulburn, Canberra) by *trucks* (TruckA, TruckB and TruckC). “Truck” and “Warehouse” are process *classes* in our model, and each truck and each warehouse is an instance of its class.[3]

Messages and channels Occasionally, *waypoints* for the trucks are available (say by GPS) in terms of time and location. There are *dockets* for goods and their origins, and the goods are loaded on trucks at a warehouse. These events are sent as messages

[2] Similar to Prolog’s `fail` statement.

[3] As an object-oriented language, Scala gives us the class/instance paradigm for free.

into an external input channel and then dispatched into internal channels typed for “waypoint” notifications and “loading” activities.

Process rules The process logic supports that waypoints are not noticed and that occasionally the origin of goods is not recorded. That is, not all events find their way into the system, or some remain partially specified. In general, the system should try its best to complete missing information or deal with it in another reasonable way. This is obviously a domain-dependent task. For the sake of illustration, we use the following rules for trucks:

1. If a current waypoint message specifies a location L for truck T then L is recorded as T's current location.
2. If a current loading message specifies that goods have been loaded on truck T at location L and the current location recorded with T is different to L then a waypoint message is broadcast, specifying that T is at L.
3. If a current loading message specifies that goods have been loaded on truck T with “unknown” origin that loading message with “unknown” is replaced by “Riverina” *or* that loading message with “unknown” is replaced by “Batlow” through broadcasts.

Rule (1) is the normal way of recording locations of trucks. Rule (2) infers a missing waypoint message. Notice that Rule (2) does not simply set the current location of T to L but sends a message instead. This allows other processes listening for waypoint messages (the warehouse processes, in our example) to also be informed about the inferred waypoint. Rule (3) branches into alternatives for resolving the missing information by making it concrete. Each case will be investigated in a separate strand of computation and may trigger further consequences (“what-if” reasoning).

9.4.2 Shallow Embedding in Scala

Our approach to process modelling is implemented in Scala [22]. Scala is a modern high-level programming language that combines object-oriented and functional programming styles. Scala comes with a comprehensive data structure library and runs on the Java virtual machine, allowing Scala code to use existing Java libraries in a straightforward way.

Scala has functions as first-class objects and supports user-definable pre-, post- and infix syntax. With these features, Scala is suitable as a host language for embedding domain-specific languages (DSLs). (See, e.g. [12] for a Scala DSL for runtime verification.) In our case, we embed a DSL for modelling the processes and channels from above. The embedding is shallow, i.e., the source constructs of the DSL are actual Scala code with DSL-specific classes and methods, which then needs to be compiled by the Scala compiler to be executable. Scala’s (optional) call-by-name parameter passing style enables us to take statements as data and, hence, to explicitly invoke their execution, or not. This feature was instrumental for implementing rules as partial functions, where rule applicability reduces to partial function definedness and rule execution reduces to statement execution.

Listings 1, 2 and 3 show concrete excerpts of our food supply chain example written in our DSL. A shallow embedding, it is comprised mostly of standard Scala. The DSL-specific language constructs are underlined.

Listing 9.1 Channels and message types. **package** and `include` declarations are not shown. Here and below, the dots indicate "glue code" for serialization and deserialization.

```
// Waypoint: observed time and location of a specific truck
case class Waypoint(time: LocalDateTime, truck: String, location: String)
object WaypointChannel extends Channel[Waypoint]("Waypoint") { ... }

// Loading: time and location of goods from a specific origin loaded on a truck
case class Loading(time: LocalDateTime, truck: String, location: String,
                   goods: String, origin: String)
object LoadingChannel extends Channel[Loading]("Loading") { ... }

// Input: sole external channel for receiving messages in Json
object Input extends Channel[JsObject]("Input", withInputPort = 5554, window = 1)
```

Listing 1 defines the main message types—`Waypoint` and `Loading`—and corresponding internal channels. The message types are ordinary Scala case classes. `Input` is an external channel whose messages are deserialized and dispatched into the other two channels. We use JSON as the format of external messages, but any other format can be used in its place. The declaration `withInputPort=5554` specifies that the channel's messages are received over TCP. The `window` size says how many messages are taken from the input queue for the next round of processing.

Listing 9.2 The Truck processes, one for each truck.

```
class Truck(Id: String) extends Process("Truck") {
  // Keeps track of the current location and load of this truck
  var location = "unknown"
  var load = Set.empty[(String, String)] // Items on this truck, as (goods, origin)

  stateVar("location", ..., ...) // The Process state is comprised of location and load
  stateVar("load", ..., ...)

  val waypointChannel = WaypointChannel.subscribe() // Channel subscriptions
  val loadingChannel = LoadingChannel.subscribe()

  rules (
    waypointChannel −−> {
      case Waypoint(_, Id, loc) if location != loc =>
        location = loc // Update current location
      case _ => () // All other cases ignored
17      },
18      loadingChannel −−> {
19        case p @ Loading(time, Id, loc, _, _) if location != loc =>
20          // Infer waypoint from this Loading message and inform all processes
21          WaypointChannel <−− Waypoint(time, Id, loc)
22          LoadingChannel <−− p // Send Loading message again in order not to loose it
23        case Loading(time, Id, loc, goods, origin) if origin != "unknown" =>
24          // Fully specified loading
25          load += ((goods, origin)) // Add to current load
26        case Loading(time, Id, loc, goods, origin) if origin == "unknown" =>
27          // Partially specified loading
28          // Branch out into two cases, replacing unknown origin by concrete places
29          or ( LoadingChannel <−− Loading(time, Id, location, goods, "Riverina"),
30               LoadingChannel <−− Loading(time, Id, location, goods, "Batlow") )
31        case _ => ()
32      }
33    )
34  }
```

Listing 2 defines the `Truck` process class. The main program (not shown here) schedules instances by statements like `Scheduler.schedule(new Truck ("TruckA"))`. A truck's state is given by its location and current load. Lines 3 and 4 are the local variables, lines 6 and 7 declare them as the process' externally visible state. Lines 9 and 10 subscribe to the two channels of interest for the process. The rules-statement in line 12 defines two rules.

The first rule reads the `WaypointChannel` via the `->` method. A `case` statement defines a partial function by pattern matching. The first case sets the current `location` to the location given by the `Waypoint` message. The second case is a catch-all to make sure that the channel will not be blocked if the first case does not apply.

The second rule deals with `Loading` messages. The first case infers a (possibly missing) `Waypoint` message from the `Loading` message and sends it to the `Waypoint` channel. It will be picked up later by the first rule and processed as described above. The second case applies to complete `Loading` messages, and updates the `load` variable. The third case applies when the origin of the goods is "`unknown`". By its disjunctive conclusion, the `or`-statement, state generation branches out by replacing "`unknown`" with concrete alternatives.

Listing 9.3 The Warehouse processes, one for each warehouse.

```
class Warehouse(Location: String) extends Process("Warehouse") {
  // Keeps track of the set of trucks currently at this warehouse
  var trucks = Set.empty[String]
  stateVar(“trucks”, ..., ...)

  val waypointChannel = WaypointChannel.subscribe()
  rules (
    waypointChannel --> {
      case Waypoint(time, truck, Location) if
          (! (trucks contains truck)) => trucks += truck
      case Waypoint(time, truck, loc) if loc != Location &&
          (trucks contains truck) => trucks -= truck
      case _ => ()
    }
  )
}
```

Finally, Listing 3 shows a second process class that subscribes to the `Waypoint` channel. A `Warehouse` process instance reads `Waypoint` messages to track which trucks are currently at the warehouse. Note that this process will also see `Waypoint` messages inferred by a `Truck` process.

9.5 Conclusions

We introduced a novel architecture for situational awareness for industrial operations. As our main contribution in this paper, we presented a design and implementation of one core component, the inference engine and its associated process modelling language, via a domain-specific embedding into Scala. We illustrated our approach with a small example from the food supply chain domain. Our implemented system runs this example as described in the main part of the paper, but we did not include a log here for space reasons. We have also been gathering experience with our system on more elaborate food supply chain, factory floor and data cleaning applications. In each of these, we have found our modelling approach of processes, rules and channels confirmed to be viable in practice. Notwithstanding this promising experience, we need to further mature our system as a prerequisite for wider impact. We envisage a number of things: enriching the modelling language by an ontological component, e.g. a description logic, for added declarative domain modelling and reasoning; employing a declarative, temporal-logic-based system for state maintenance as indicated in Sect. 9.3; model-checking the process models (this will be possibe only for controlled subsets of Scala); and probabilistic reasoning based on distributions for conclusions of disjunctive rules.

References

1. Allen JF (1983) Maintaining knowledge about temporal intervals. Commun ACM 26(11):832–843
2. The Array of Things. https://arrayofthings.github.io/
3. Baier C, Katoen J (2008) Principles of model checking. MIT Press
4. Brandt S, Kalaycl EG, Ryzhikov V, Xiao G, Zakharyaschev M (2018) Querying log data with metric temporal logic. J Artif Intell Res 62(5):829–877

5. Chesani F, Mello P, Montali M, Riguzzi F, Sebastianis M, Storari S (2008) Checking compliance of execution traces to business rules. In: International conference on business process management. Springer, Berlin
6. Cordier MO, Thiébaux S (1994) Event-based diagnosis for evolutive systems. In: Proceedings 5th international workshop on principles of diagnosis
7. Cook JE, Wolf AL (1999) Software process validation: quantitatively measuring the correspondence of a process to a model. ACM Trans Softw Eng Methodol 8(2)
8. De Giacomo G, Maggi FM, Marrella A, Sardina S (2016) Computing trace alignment against declarative process models through planning. In: Proceedings ICAPS
9. Derigent W, Thomas A (2017) Situation awareness in product lifecycle information systems. In: Service orientation in holonic and multi-agent manufacturing—proceedings of SOHOMA 2017, Springer, Berlin
10. Grastien A, Haslum P, Thiébaux S (2012) Conflict-based diagnosis of discrete event systems: theory and practice. In: Proceedings KR 2012
11. Ghimire S, Luis-Ferreira F, Nodehi T, Jardim-Goncalves R (2017) Iot based situational awareness framework for real-time project management. Int J Comput Integr Manuf 30(1)
12. Havelund K, Joshi R (2017) Modeling rover communication using hierarchical state machines with scala. In: Computer safety, reliability, and security—SAFECOMP 2017 workshops, LNCS 10489, Springer, Berlin
13. Holzmann Gerard J (1997) The model checker spin. IEEE Trans Softw Eng 23(5):279–295
14. Havelund K, Peled D (2018) Efficient runtime verification of first-order temporal properties. In: Model checking software—25th international symposium, SPIN 2018, LNCS 10869, Springer, Berlin
15. Industry 4.0. https://en.wikipedia.org/wiki/Industry_4.0
16. Kauffman S, Havelund K, Joshi R (2016) nfer—A notation and system for inferring event stream abstractions. In: Runtime verification—16th international conference, RV 2016. LNCS 10012, Springer, Berlin
17. Lee, J Davari Ardakani H, Yang S, Bagheri B (2015) Industrial big data analytics and cyber-physical systems for future maintenance and service innovation. Procedia CIRP 38:3–7
18. McIlraith S (1994) Toward a theory of diagnosis, testing and repair. In: Proceedings 5th international workshop on principles of diagnosis
19. Oberoi S (2017) Situational awareness for the factory floor. https://blogs.oracle.com/iot/situational-awareness-for-the-fac-tory-floor
20. OMG (2013) BPMN 2.0: Specification. www.omg.org/spec/BPMN/
21. SAP Event Management 9.2, 1.8 edition (2017)
22. The Scala Programming Language. https://www.scala-lang.org
23. Singh D, Tripathi G, Jara AJ (2014) A survey of internet-of-things: Future vision, architecture, challenges and services. In: 2014 IEEE World Forum on Internet of Things, WF-IoT 2014. IEEE
24. Wil MP (2013) van der Aalst. A comprehensive survey. ISRN Software Engineering, Business process management

Part III
Case Studies and Novel Applications in Non-specific Operational Scenarios

Chapter 10
Dynamic Relocation of Aerial Firefighting Resources to Reduce Expected Wildfire Damage

Nicholas Davey, Simon Dunstall, and Saman Halgamuge

Abstract Aerial firefighting resources are an integral part of modern wildfire suppression strategies. In many locations around the world where wildfires pose a serious threat, firefighting authorities have access to fleets of different aircraft. These can be used to provide support to land-based resources during the extended attack of existing fires or to quickly suppress recent spark events during the initial attack phase. As the amount of time that a fire has been burning is a predictor of the amount of damage it causes, fast aerial response times are critical. Therefore, there is significant value in dynamically repositioning aircraft to airbases and fires over the course of a fire day or fire season. In this paper, we devise one such approach based on model-predictive control to make relocation decisions at various times over a single day. These relocation decisions are based on solving an underlying Mixed-integer linear program (MILP) so as to minimize expected damage over a lookahead horizon. The inputs to this program are updated at each of these decision times based on prevailing stochastic weather conditions, the current state of fires in the region, and the current assignment of aircraft to bases and fires. The expected fire damage profiles used in this model are based on empirical data that is pre-computed for the region of interest. We apply our model to a scenario in Central Chile and show that with careful parameter selections it is possible to make improved relocation decisions to reduce the expected fire damage in a region using this approach.

Keywords Mixed-integer linear programming · Model-predictive control · Optimal relocation · Wildfires

N. Davey (✉) · S. Halgamuge
The University of Melbourne, Parkville, VIC, Australia
e-mail: ndavey@student.unimelb.edu.au

S. Halgamuge
e-mail: saman@unimelb.edu.au

S. Dunstall
Data61 CSIRO, Docklands, VIC, Australia
e-mail: Simon.Dunstall@data61.csiro.au

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_10

10.1 Introduction

The demand for firefighting resources to combat wildfires has steadily increased over the past century [1, 2]. This can be attributed to both climate change [2], which is increasing the frequency and severity of outbreaks, and land use patterns that place communities close to fire-prone areas [3]. This is evident from the destruction caused by many recent high-profile disasters [4]. Unfortunately, many fire-prone locations throughout the world are experiencing wetter winters and hotter and drier summers—a potentially disastrous combination that will literally "add fuel to the fire" of future ignition events and increase the damage they cause [5].

Aerial resources such as tankers and helicopters play a pivotal role in mitigating this damage and can be used in both *extended* and *initial* attack phases when suppressing wildfires. In *extended attack*, air tankers and helicopters provide support to ground crews to keep existing fires under control. Their capabilities include laying firebreaks ahead of fires or providing direct suppression of portions of the blaze to try and extinguish them [6]. During *initial attack*, aerial resources can suppress fires before they grow too large. They are particularly useful in areas that are inaccessible to ground crews or that are too far away for ground crews to reach in an acceptable amount of time [7]. As only a small percentage of established fires are responsible for the vast majority of total damage [8, 9], prompt aerial responses (typically within 20–30 min) that extinguish fires early can dramatically reduce the total damage across a region.

However, deciding how to assign aircraft to bases and fires is a difficult question that depends on many factors such as expected fire weather, existing fire severities, expected suppression success and current resource availability and location throughout the region [10]. This has important implications on both the strategic level of how many aircraft to charter and of what type, as well as tactical level decisions of where and when to deploy these aircraft [1]. These two tasks are typically performed by a centralized command [11].

According to Calkin [9], initial attack is generally quite effective, with most fires being extinguished before they reach a critical size. However, the fires that do progress beyond this initial phase may also require aerial support, thus creating a competing need for a finite number of resources. Unfortunately, regions to be covered are large, and the fire danger can vary dramatically both spatially and temporally throughout a fire season. Therefore, the optimal dynamic allocation of aerial resources to airbases and fires throughout a fire season can have a large effect on the total fire damage incurred.

Firefighting agencies are well aware of the need for optimal management of resources. In fact, they have used Operations Research (OR) techniques to manage resources as early as the 1960s [12]. The earliest known application of OR specifically related to air tankers was in 1984 [13]. In this study, the authors used a simulation model to make aircraft stationing decisions.

Later works have regularly applied linear programming techniques. For example, in [14], the authors used a mixed-integer linear program (MILP) to make strategic home base assignment decisions for aircraft. A number of more recent studies have investigated ways to deploy fire resources to bases to specifically improve initial attack success. Donovan and Rideout [15] developed a multi-period MILP model to assign fire trucks to locations so as to minimize the growth of a single fire over a certain period of time. This technique was subsequently applied to multiple fires in [16, 17]. The aforementioned models all sought to minimize the contained area of fires and therefore included detailed models of the fire line over time.

Recently, MILP models have been extended to account for the inherent uncertainty in the relocation decision process. Haight and Fried [18] developed a stochastic model that assigned a fixed number of resources to bases, which they solved with a two-stage MILP. This model sought to minimize the number of fires that progressed beyond the initial attack phase through the optimal stationing of aircraft to these bases. Wei et al. [19] developed a comprehensive two-stage model that also sought to minimize the number of fires that are not contained over a given timeframe. They explicitly modelled fire containment, similar to earlier studies, but also used a two-stage, chance-constrained MILP, similar to Haight and Fried [18] in order to account for uncertainty.

While the above models are able to account for uncertainty, they mainly focus on terrestrial vehicles that have fixed home bases at the start of a season. However, aerial resources have a greater opportunity to relocate between bases than land-based resources due to their ability to cover large distances in a short period of time. This is an opportunity for firefighting authorities to pre-emptively relocate aerial resources to be based in areas with greater fire danger. However, realizing the value of relocation presents a number of computational challenges. In fact, a full analysis of its advantages would require the use of stochastic dynamic programming approaches that can be computationally intractable [12]. Two studies that addressed this dynamic relocation problem were [20, 21]. In the former, the authors used a queueing model to manage redeployment decisions and a heuristic to deal with computational tractability. In the latter, the authors made static relocation decisions at daily intervals using a "chance-constrained" p-median relocation linear programming model. The parameter inputs of [21] that reflected the fire danger throughout the region were updated using a rolling horizon and managed using model-predictive control (MPC). The overall objective in their model was to minimize the expected costs of relocation as opposed to reducing the expected damage caused by fires.

In this paper, we develop an approach to dynamically relocate aircraft over time so as to minimize expected fire damage over the course of a single fire season. We use a model predictive control approach, similar to [21] to dynamically resolve an MILP. However, we also integrate the concepts of expected damage, which is similar to other studies such as [18, 19]. Therefore, the parameters of the program presented in this paper are updated at each future time period based on stochastic fire danger and expected fire damage growth probabilities.

The paper is organized as follows. Section 10.2 outlines the formulation of the optimization problem and the model-predictive framework. Section 10.3 then describes the embedded MILP that is updated at each time step, while Sect. 10.4 applies the model to a case study. Finally, Sect. 10.5 concludes the paper and presents an agenda for future work.

10.2 Problem Description

The problem investigated in this paper consists of optimally updating the assignments of a fixed number of helicopters and tankers to bases and active fires over the course of a single fire day. These relocations are performed so as to minimize expected fire damage, D. The aircraft must also be available at short notice to suppress new outbreaks if they occur. Finally, there is an additional operational requirement that no aircraft can exceed its maximum flying hours for the day.

Let $\boldsymbol{\theta}^t = \left[\boldsymbol{\phi}^t, \boldsymbol{\beta}^t, \mathbf{G}^t\right]$ represent the state of the system at time t, where $\boldsymbol{\phi}^t$ is the vector of severities of currently burning fires, $\boldsymbol{\beta}^t$ is the vector of the fire danger index in each patch and $\mathbf{G}^t$ is the cumulative flying hours recorded for each aircraft. In addition, let vectors $\boldsymbol{X}^t$ and $\boldsymbol{Y}^t$ represent the binary assignments of aircraft to bases and fires, respectively. Their respective components are denoted $X_{r,b}$ (aircraft r is assigned to base b), and $Y_{r,m}$ (aircraft r is assigned to fire m). The minimum expected damage for the remainder of the day, $D\left(t, \boldsymbol{\theta}\right)$, given the current time step and state, is found by solving for $\boldsymbol{X}^t, \boldsymbol{Y}^t$ in the Bellman equation:

$$D\left(t, \boldsymbol{\theta}\right) = \min_{\boldsymbol{X}^t, \boldsymbol{Y}^t} \left\{ f\left(t, \boldsymbol{\theta}^t; \boldsymbol{X}^t, \boldsymbol{Y}^t\right) + \Phi\left(t, \boldsymbol{\theta}^t; \boldsymbol{X}^t, \boldsymbol{Y}^t\right) \right\}, \tag{10.1}$$

where

$$\Phi\left(t, \boldsymbol{\theta}^t; \boldsymbol{X}^t, \boldsymbol{Y}^t\right) = \mathbb{E}\left\{ \sum_{u=t}^{T} f\left(u, \boldsymbol{\theta}^u; \hat{\boldsymbol{X}}^u, \hat{\boldsymbol{Y}}^u\right) \middle| \boldsymbol{\theta}^t, \boldsymbol{X}^t, \boldsymbol{Y}^t \right\} \tag{10.2}$$

is the conditional expectation of future fire damage until the end of the day assuming that optimal assignment decisions ($\hat{\boldsymbol{X}}^u, \hat{\boldsymbol{Y}}^u$) are made at all future time periods. Additionally, $f\left(u, \boldsymbol{\theta}^u; \hat{\boldsymbol{X}}^u, \hat{\boldsymbol{Y}}^u\right)$ represents the expected single period fire damage given the prevailing state and chosen assignments.

Solving Eq. 10.1 optimally requires the use of stochastic dynamic programming over all possible states and assignments, which would be computationally intractable [12]. As a compromise, we adopt a model predictive approach with a rolling horizon, in a manner similar to [21]. Within this framework, at each time step, an MILP is solved to find the best aircraft assignments to make given the fire danger forecast over a lookahead horizon and the current state of the system. This MILP uses the expectations of fire damage over the lookahead window to make relocation decisions. We denote the expected damage at time t of an existing fire, m by $\delta_{m,\kappa}^{E,t}$ and the expected damage for potential fires at location n by $\delta_{n,\kappa}^{P,t}$. These expectations are conditional upon the forecast fire danger index over the lookahead horizon ($(\beta_m)_{\mathcal{T}}$ and $(\beta_n)_{\mathcal{T}}$), and the aircraft configuration applied to the fire/patch at time t, κ. They are computed at each time step using the following equations, which can be obtained by analysing empirical data for the study region:

$$\delta^{E,t}_{m,\kappa} : \qquad \delta^{E,t}_{m,\kappa}\left(\phi^t_m, (\beta_m)_{\mathcal{T}}\right) \tag{10.3}$$

$$\delta^{P,t}_{n,\kappa} : \qquad \delta^{P,t}_{n,\kappa}\left((\beta_n)_{\mathcal{T}}\right). \tag{10.4}$$

The parameter κ is an index corresponding to a single aircraft configuration from a finite set of possible configurations. A valid configuration is defined by the number of helicopters and tankers that are within a 20 min travel time to the fire/location (*early arrivals*), and the numbers outside a 20 min travel time (*late arrivals*). Therefore, a configuration can be represented by an encoding of the form $TE(X)HE(X)TL(X)HL(X)$, where TE, HE, TL and HL refer to early tankers, early helicopters, late tankers and late helicopters, respectively. The X's are the corresponding numbers of aircraft of each component in the configuration.

10.3 MILP Relocation Model

The embedded MILP updates aircraft assignments at time t so as to minimize the expected fire damage over the lookahead horizon, $\mathcal{T}$. It does this by explicitly accounting for expected damage given the burning index throughout the region ($\boldsymbol{\beta}^t$), the severity of existing fires ($\boldsymbol{\phi}^t$), and the current assignments of aircraft to bases ($\boldsymbol{X}^t$) and existing fires ($\boldsymbol{Y}^t$).

We first define the following sets that are used in the MILP:

R	Set of all aerial resources	$r \in \{1, \ldots, r_{max}\}$
B	Set of bases	$b \in \{1, \ldots, b_{max}\}$
M	Set of active fires at time t	$m \in \left\{1, \ldots, m^t_{max}\right\}$
N	Set of patches	$n \in \{1, \ldots, n_{max}\}$
$\mathcal{T}$	Set of time periods over lookahead horizon	$\tau \in \{t, \ldots, t+T\}$
K	Set of possible attack configuration indices	$\kappa \in \{1, \ldots, \kappa_{max}\}$
C	Set of indices of the components of configurations	$c \in \{1, 2, 3, 4\}$

In addition to the parameters introduced in Sect. 10.2, we also define the following extra parameters that are used in the MILP:

- λ is a user input between 0 and 1 that weights the cost of relocation versus keeping aircraft in their current configuration.
- $d^1_{r,b}$ is the distance between aircraft r and base b.
- $d^2_{r,m}$ is the distance between resource r and fire m.
- $d^3_{b,n}$ is the distance between base b and patch n.
- $\Delta^E_{m,\kappa}$ is a binary variable that indicates whether fire m is assigned configuration κ.

- $\Delta^P_{n,\kappa}$ is a continuous variable between 0 and 1 that indicates the rate at which to apply configuration κ to patch n over the lookahead.
- $A_{r,b}$ is a binary variable that indicates whether resource r is available to fight fires from base b.
- Q^κ_c is the number of aircraft in component c of κ required to satisfy configuration κ.
- η^c_b is the expected number of fires within a 20 min radius of base b that will break out over the lookahead horizon. If the expected number is less than 1, η^c_b is set to 1.
- $\mathbb{1}_c\left(d\right)$ is an indicator function that equals 1 if the travel distance d satisfies component c of the configuration encodings.
- G^t_r is the number of flying hours at time t logged by aircraft r. The maximum number of daily hours for the same aircraft is given by $G^{\max}_r$.

Using the parameters defined in Sect. 10.2 and those defined above, we can now state the MILP. All t superscripts except for those relating to accumulated flying hours have been suppressed for improved legibility.

Objective Function

$$D\left(t,\boldsymbol{\theta}\right)=\min_{\mathbf{X},\mathbf{Y}}\left\{\lambda\left[\sum_K\sum_N\delta^P_{n,\kappa}\Delta^P_{n,\kappa}+\sum_K\sum_M\delta^E_{m,\kappa}\Delta^E_{m,\kappa}\right]+\left(1-\lambda\right)\sum_B\sum_R d^1_{r,b}X_{r,b}\right\} \tag{10.5}$$

Constraints

$$Q^\kappa_c\Delta^E_{m,\kappa}\leq\sum_R\mathbb{1}_c\left(d^2_{r,m}\right)Y_{r,m} \qquad \forall\kappa,c,m \tag{10.6a}$$

$$Q^\kappa_c\Delta^P_{n,\kappa}\leq\sum_B\sum_R\mathbb{1}_c\left(d^3_{b,n}\right)A_{r,b}\Big/\eta^c_b \qquad \forall\kappa,c,n \tag{10.6b}$$

$$A_{r,b}\leq X_{r,b} \qquad \forall r,b \tag{10.6c}$$

$$\sum_B A_{r,b}+\sum_M Y_{r,m}=1 \qquad \forall r \tag{10.6d}$$

$$\sum_B X_{r,b}=1 \qquad \forall r \tag{10.6e}$$

$$\sum_M Y_{r,m}\leq 1 \qquad \forall r \tag{10.6f}$$

$$\sum_\kappa \Delta^E_{m,\kappa}=1 \qquad \forall m \tag{10.6g}$$

$$\sum_{\kappa} \Delta^{P}_{n,\kappa} = 1 \qquad \forall n \qquad (10.6\text{h})$$

$$\sum_{M} d^{2}_{r,m} Y_{r,m} + \sum_{B} d^{1}_{r,b} X_{r,b} \leq G^{max}_{r} - G^{t-1}_{r} \qquad \forall r \qquad (10.6\text{i})$$

$$X_{r,b}, A_{r,b} \in \{0, 1\} \quad \forall r, b; \quad Y_{r,m} \in \{0, 1\} \quad \forall r, m; \qquad (10.6\text{j})$$

$$\Delta^{E}_{m,\kappa} \in \{0, 1\} \quad \forall m, \kappa; \quad \Delta^{P}_{n,\kappa} \in [0, 1] \quad \forall n, \kappa \qquad (10.6\text{k})$$

Constraint set (10.6a) specifies that a particular aircraft configuration at fire m can only be met if the correct number of aircraft to satisfy each of the components, c, in configuration κ are available. Constraint set (10.6b) is the same constraints for potential fires at patch n. It also considers the expected number of aircraft that each base will see, which is captured by η^{c}_{b}. This accounts for the fact that aircraft may be needed to fight multiple potential fires over the lookahead horizon. Constraints (10.6c) ensure that aircraft r can only be available from base b if it is actually stationed at b. Constraint set (10.6d) ensures that each aircraft is either available at a base or is assigned to a fire. Constraint set (10.6e) ensures that aircraft r is assigned to one and only one base. Constraints (10.6f) ensure that an aircraft can only be allocated to at most one fire. Constraints (10.6g) and (10.6h) ensure that each patch and fire must only be assigned one configuration. Constraint (10.6i) makes sure that the total number of flying hours for each aircraft is below the maximum permissible hours for the day. Finally, (10.6j) and (10.6k) are binary and bounds constraints on all decision variables.

10.4 Numerical Example

We apply our model to a case study in Central Chile based on the 2017 megafire season. This fire season was the worst on record for the country, with over 518,000 Ha of land burned, 10 deaths, over 3000 homes destroyed and over \$370m spent to fight the fires [3].

The inputs used in this example are the fire danger parameters and the conditional probabilities that are dependent upon them. The growth rates in the absence of any firefighting efforts were estimated using AMICUS [24]. The values for different configurations of early and late aircraft were assumed to reduce with the number of aircraft in the configuration.

Our numerical example consists of 17 identical fixed-wing tankers (cruising speed of 356 km/hr) that are required to provide fire coverage to the 16 provinces within the regions of Maule, Bío Bío, Araucanía, Los Ríos, and Los Lagos, as shown in Fig. 10.1. For each of these provinces, one air strip was selected for basing the aircraft. Our example uses 16 possible attack configurations, consisting of all possible combinations of (0/1/2/3+) aircraft arriving within 20 min and (0/1/2/3+) aircraft arriving after 20 min, and are applied to both initial and extended attacks. No helicopters are used in this example. In addition, the embedded relocation model uses

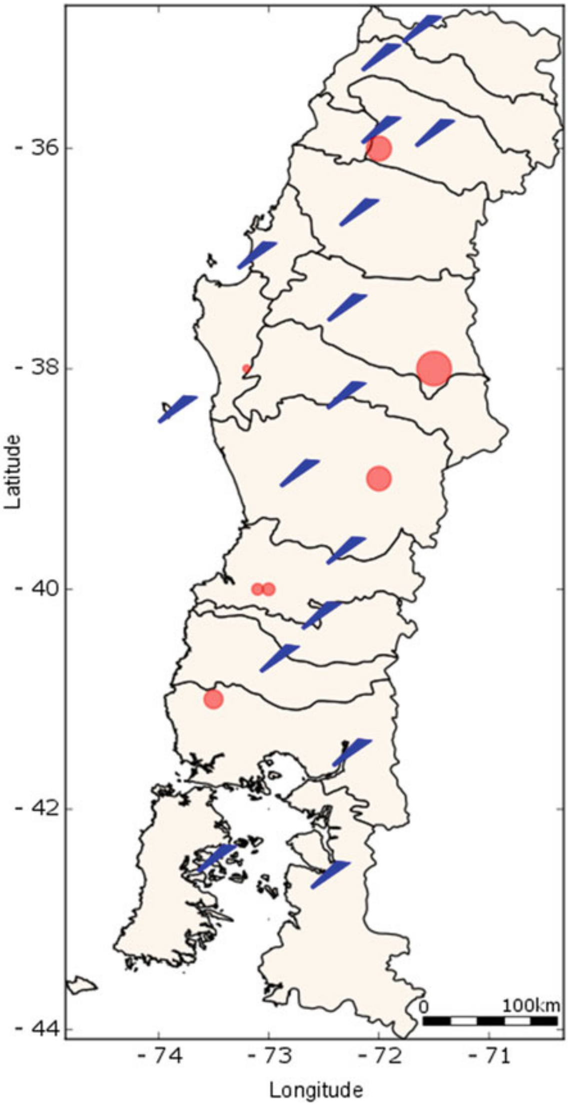


Fig. 10.1 Region of interest showing province boundaries and the location of air strips used in the example

a forward horizon of 5 hours. Finally, each sample run begins with the same seven fires, as represented by the circles in Fig. 10.1.

Input Parameters and Initial Fires The main predictors of expected damage are the McArthur Forest Fire Danger Index (FFDI, β) and the configurations applied to each patch and fire. The FFDI values were computed from raw weather data for the three sample days tested (17/12/2017, 19/12/2017 and 21/12/2017). Due to the short time horizon of a single day, the FFDIs at all time steps of the model are assumed to be

Table 10.1 Conditional probabilities for growth rates and initial sizes used in the numerical example

		Forecast Burning Index ($\boldsymbol{\beta}_{\mathcal{T}}$)					
		5	10	15	20	25	30
Ignitions /10^3 km^2/h	Poisson Param.	0.006	0.012	0.018	0.024	0.030	0.036
	Config.						
Expected Damage, Initial Attack (δ^E, Ha/hr) Mean (Std. Dev.)	TE0TL0	1.9 (0.4)	20.1 (4.0)	29.2 (5.8)	41.2 (8.2)	48.1 (9.6)	60.9 (12.2)
	TE1TL0	0.1 (0.0)	1.9 (0.4)	4.2 (0.8)	7.7 (1.5)	11.1 (2.2)	13.8 (2.8)
	TE2TL0	0.0 (0.0)	0.6 (0.1)	1.4 (0.3)	2.6 (0.5)	3.7 (0.7)	4.6 (0.9)
	TE3TL0	0.0 (0.0)	0.2 (0.0)	0.3 (0.1)	0.6 (0.1)	0.9 (0.2)	1.2 (0.2)
	TE0TL1	0.9 (0.2)	9.4 (1.9)	13.7 (2.7)	19.3 (3.9)	22.5 (4.5)	28.6 (5.7)
	TE1TL1	0.0 (0.0)	0.9 (0.2)	2.0 (0.4)	3.6 (0.7)	5.2 (1.0)	6.5 (1.3)
	TE2TL1	0.0 (0.0)	0.3 (0.1)	0.7 (0.1)	1.2 (0.2)	1.7 (0.3)	2.2 (0.4)
	TE3TL1	0.0 (0.0)	0.1 (0.0)	0.2 (0.0)	0.3 (0.1)	0.4 (0.1)	0.5 (0.1)
	TE0TL2	0.4 (0.1)	4.4 (0.9)	6.4 (1.3)	9.0 (1.8)	10.5 (2.1)	13.3 (2.7)
	TE1TL2	0.0 (0.0)	0.4 (0.1)	0.9 (0.2)	1.7 (0.3)	2.4 (0.5)	3.0 (0.6)
	TE2TL2	0.0 (0.0)	0.1 (0.0)	0.3 (0.1)	0.6 (0.1)	0.8 (0.2)	1.0 (0.2)
	TE3TL2	0.0 (0.0)	0.0 (0.0)	0.1 (0.0)	0.1 (0.0)	0.2 (0.0)	0.3 (0.1)
	TE0TL3	0.2 (0.0)	2.0 (0.4)	3.0 (0.6)	4.2 (0.8)	4.9 (1.0)	6.2 (1.2)
	TE1TL3	0.0 (0.0)	0.2 (0.0)	0.4 (0.1)	0.8 (0.2)	1.1 (0.2)	1.4 (0.3)
	TE2TL3	0.0 (0.0)	0.1 (0.0)	0.1 (0.0)	0.3 (0.1)	0.4 (0.1)	0.5 (0.1)
	TE3TL3	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.1 (0.0)	0.1 (0.0)	0.1 (0.0)
Growth Rate, Extended Attack (δ^E, km/hr)	TE0TL0	0.3 (0.1)	0.9 (0.2)	1.0 (0.2)	1.2 (0.2)	1.3 (0.3)	1.5 (0.3)
	TE1TL0	0.1 (0.1)	0.3 (0.2)	0.3 (0.2)	0.4 (0.2)	0.5 (0.3)	0.6 (0.3)
	TE2TL0	0.0 (0.1)	0.0 (0.2)	0.1 (0.2)	0.1 (0.2)	0.2 (0.3)	0.2 (0.3)
	TE3TL0	−0.1 (0.1)	−0.2 (0.2)	−0.2 (0.2)	−0.2 (0.2)	−0.2 (0.3)	−0.1 (0.3)
	TE0TL1	0.2 (0.1)	0.8 (0.2)	1.0 (0.2)	1.1 (0.2)	1.2 (0.3)	1.4 (0.3)
	TE1TL1	0.1 (0.1)	0.2 (0.2)	0.3 (0.2)	0.3 (0.2)	0.4 (0.3)	0.5 (0.3)
	TE2TL1	0.0 (0.1)	0.0 (0.2)	0.0 (0.2)	0.0 (0.2)	0.1 (0.3)	0.1 (0.3)
	TE3TL1	−0.1 (0.1)	−0.2 (0.2)	−0.2 (0.2)	−0.3 (0.2)	−0.2 (0.3)	−0.2 (0.3)
	TE0TL2	0.2 (0.1)	0.8 (0.2)	0.9 (0.2)	1.1 (0.2)	1.1 (0.3)	1.3 (0.3)
	TE1TL2	0.0 (0.1)	0.2 (0.2)	0.2 (0.2)	0.3 (0.2)	0.3 (0.3)	0.4 (0.3)
	TE2TL2	0.0 (0.1)	−0.1 (0.2)	−0.1 (0.2)	0.0 (0.2)	0.0 (0.3)	0.0 (0.3)
	TE3TL2	−0.1 (0.1)	−0.3 (0.2)	−0.3 (0.2)	−0.3 (0.2)	−0.3 (0.3)	−0.3 (0.3)
	TE0TL3	0.2 (0.1)	0.7 (0.2)	0.8 (0.2)	1.0 (0.2)	1.1 (0.3)	1.2 (0.3)
	TE1TL3	0.0 (0.1)	0.1 (0.2)	0.1 (0.2)	0.2 (0.2)	0.2 (0.3)	0.3 (0.3)
	TE2TL3	0.0 (0.1)	−0.1 (0.2)	−0.1 (0.2)	−0.1 (0.2)	−0.1 (0.3)	−0.1 (0.3)
	TE3TL3	−0.1 (0.1)	−0.3 (0.2)	−0.4 (0.2)	−0.4 (0.2)	−0.4 (0.3)	−0.4 (0.3)

known with certainty at the start of the day. Therefore, the uncertainties that remain are fire occurrences, growth rates and success rates under different combinations of aircraft configuration and FFDI. These are shown in Table 10.1. The occurrence rates are assumed to follow a Poisson distribution, while the remaining conditional probabilities are all assumed to follow truncated normal distributions. While the probabilities used are synthetic, they reflect the fact that earlier arrivals, more aircraft and lower FFDI values result in higher success rates and lower expected damage.

Table 10.2 Expected fire damage (1,000 Ha) and standard deviations for various values of λ as well as for the fixed base assignment case

Test	Mean (Std. Dev.) Damage, 1000 Ha		
	17/12/2017	19/12/2017	21/12/2017
Initial	14.3 (10.3)	35.6 (19.1)	**32.9 (21.3)**
$\lambda = 0.1$	**5.8 (3.9)**	**27.6 (16.8)**	39.6 (19.5)
$\lambda = 0.3$	9.7 (8.1)	37.6 (18.0)	55.5 (22.6)
$\lambda = 0.5$	14.7 (12.2)	70.0 (25.5)	46.9 (22.5)
$\lambda = 0.7$	13.1 (9.0)	43.2 (25.0)	64.2 (26.7)
$\lambda = 0.9$	19.1 (8.3)	57.9 (22.9)	70.7 (26.0)

10.4.1 Results and Discussion

In total, six different tests were performed for each sample weather day. The first of these is a semi-static location model where aircraft base assignments are fixed at the first time period. These assignments are made based on solving the embedded MILP using expected weather over the entire day. The remaining tests are for different values of λ. For each test, 20 runs were performed and the average fire damages and standard deviations were computed. These are summarized in Table 10.2.

The six cases with different λ values were chosen to test different ratios for weighting the relocation component of the objective function relative to the expected damage component in the MILP. The most effective model among all of the cases tested is $\lambda = 0.1$. This is closely followed by $\lambda = 0.3$ and the case where aircraft bases are fixed at time 0. The cases with larger λ values performed poorly in comparison. This suggests that relocations in response to short-term spatial variations in fire weather are undesirable, particularly if they may differ from expected fire weather further into the future. This is supported by the fact that the fixed location model takes into account the fire weather over the entire day, whereas the other three models only consider a 5 hours lookahead. Together with the fact that all models are able to assign aircraft to nearby fires as they occur, this puts the fixed base assignment model at an unrealistic advantage.

It was also observed that the model with $\lambda = 1$ performed noticeably better in the first Scenario (17/12/2017) but had limited to no benefit in the other two scenarios tested. This suggests that the parameter setting is dependent on the specific fire day encountered. The underlying weather for Scenario 1 was not as severe as that for the other two scenarios but it also varied less throughout the region over the course of the day. Future tests should therefore investigate the effect of spatial variation in FFDI on the best λ setting to use and consider ways to dynamically alter this parameter as well.

This example has a number of simplifying assumptions. First, the patches used are at the province level and are fairly large. The next smallest level (comunes) consists of 150 patches. A more fine-grained approach such as this may provide a better spatial representation of both fire weather and fire damage. Second, only one air strip was used for each patch. Once again, there are over 150 possible bases in Central Chile, which may provide better relocation options for the embedded MIP. Finally, synthetic data was used for conditional probabilities. Future work aims to use probability distributions derived from analysing actual historical fire data from the region.

Finally, one drawback of the embedded MILP is that it seeks to minimize the sum of expected fire damage over the horizon *plus* the sum of relocation costs, weighted by the parameter λ. As the latter of these two components does not measure expected damage (which is what the model ultimately seeks to minimize), it may dominate in the objective function, preventing relocations that may in fact reduce overall expected damage. Future work should address this drawback.

10.5 Conclusion

In this paper, we developed an approach for dynamically repositioning aerial firefighting resources to reduce expected fire damage over a fire season. In it, we implemented a MPC framework using a rolling horizon that we used to update assignments of aircraft to bases and fires. Our model explicitly accounts for success probabilities of extinguishing fires once these relocation decisions are made. Using a properly tuned scaling parameter, it is possible to reduce the expected damage over that of a static model where assignments are made at the start of the horizon. Preliminary results suggest that the expected damage is dependent on this scaling parameter. Future work will look at performing a sensitivity analysis across a range of scaling parameters for a larger sample of test scenarios as well as alternative formulations of the embedded MILP.

References

1. Minas JP, Hearne JW, Handmer JW (2012) A review of operations research methods applicable to wildfire management. Int J Wild Fire 21:189–96
2. Westerling AL, Hidalgo H, Cayan DR, Wetnam TS (2006) Warming and earlier spring increases western U.S. forest wildfire activity. Science 313:940–943

3. de la Barrera F, Barraza F, Favier P, Ruiz V, Quense J (2018) Megafires in Chile 2017: monitoring multiscale environmental impacts of burned ecosystems. Sci Total Environ 637–638:1536–1536
4. San-Miguel-Ayanz J, Moreno JM, Camia A (2013) Analysis of large fires in European Mediterranean landscapes: lessons learned and perspectives. For Ecol Manag 294:11–22
5. Attiwill P, Binkley D (2013) Exploring the mega-fire reality: a "Forest Ecology and Management" conference. For Ecol Manag 294:1–3
6. Plucinski MP, McCarthy GJ, Hollis JJ, Gould JS (2012) The effect of aerial suppression on the containment time of Australian wildfires estimated by fire management personnel. Int J Wild Fire 21:219–229. https://doi.org/10.1071/WF11063
7. McCarthy GJ, Plucinski MP, Gould JS (2012) Analysis of the resourcing and containment of multiple remote fires: the Great Divide Complex of fires, Victoria, December 2006. Aust For 75:54–63. https://doi.org/10.1080/00049158.2012.10676385
8. Strauss D, Bednar L, Mees R (1989) Do one percent of forest fires cause ninety-nine percent of the damage? For Sci 35:319–328
9. Calkin DE, Gebert KM, Jones JG, Neilson RP (2005) Forest service large fire area burned and suppression expenditure trends 1970–2002. J For 103(4):179–183
10. Thompson MP, Calkin DE (2011) Uncertainty and risk in wildland fire management: a review. J Environ Manage 92:1895–1909
11. Stonesifer CS, Calkin DE, Thompson MP, Stockman KD (2016) Fighting fire in the heat of the day: an analysis of operational and environmental conditions of use for large airtankers in United States fire suppression. Int J Wildl Fire 25(5):520–533
12. Martell DL (2015) A review of recent forest and wildland fire management decision support systems research. Curr For Rep 1(2):128–137
13. Martell DL, Drysdale RJ, Doan GE, Boychuk D (1984) An evaluation of forest fire initial attack resources. Interfaces 14:20–32
14. Maclellan JI, Martell DL (1996) Basing airtankers for forest fire control in Ontario. Oper Res 44(5):677–686
15. Donovan GH, Rideout DB (2003) An integer programming model to optimize resource allocation for wildfire containment. For Sci 49(2):331–335
16. Kirsch AG, Rideout DB (2005) Optimizing initial attack effectiveness by using performance measures. In: Proceedings of the 2003 Symposium on systems analysis in forest resources. Stevenson, WA. USDA For Serv, Gen Tech Rep. PNW-GTR-656, Pacific Northwest Research Station, Portland, OR, pp. 183–187
17. Rideout D, Wei Y, Kirsch A (2011) Optimal allocation of initial attack resources to multiple wildfire events. Int J Saf Secur Eng 1:312–25
18. Haight RG, Fried JS (2007) Deploying wildland fire suppression resources with a scenario-based standard response model. INFOR Inf Syst Oper Res 45:31–9
19. Wei Y, Bevers M, Belval E, Bird B (2015) A chance-constrained programming model to allocate wildfire initial attack resources for a fire season. For Sci 61(2):278–288
20. Islam KS, Martell DL, Posner MJ (2009) A time-dependent spatial queueing model for the daily deployment of airtankers for forest fire control. Infor 47:319–33
21. Chow JYJ, Regan AC (2011) Resource location and relocation models with rolling horizon forecasting for wildland fire planning. INFOR Inf Syst Oper Res 49(1):31–43
22. Wilks DS (2009) A gridded multisite weather generator and synchronization to observed weather data. Water Resour Res 45(10):1–11

23. Dowdy AJ, Mills GA, Finkele K, de Groot W (2009) Australian fire weather as represented by the McArthur Forest Fire Danger Index and the Canadian Forest Fire Weather Index. Technical Report CAWCR Technical Report No. 10, The Centre for Australian Weather and Climate Research
24. Plucinski M, Sullivan A, Rucinski C, Prakash M (2017) Improving the reliability and utility of operational bushfire behaviour predictions in Australian vegetation. Environ Model Softw 91:1–12

Chapter 11
The Operating Room Scheduling Problem Based on Patient Priority

Omolbanin Mashkani, F. J. Hwang, and Amir Salehipour

Abstract An efficient operating theatre schedule contributes significantly to enhancing the efficiency of hospital operation management and plays a critical financial role in most hospital settings. In this paper, an operating room scheduling problem based on patient priority is investigated at tactical and operational levels subject to specific strategic decisions. At the tactical level, the main goal is to generate a cyclic time table, known as the master surgical schedule (MSS) and can be repeated over the planning horizon of several months to years. Operational level concerns about allocating patients to operating rooms and determining the day of surgeries, which is called the surgical case assignment problem (SCAP). To handle the problems at both decision levels simultaneously, known as the MSS-SCAP problem, an integer linear programming (ILP) model, called MSS-SCAP model, and a heuristic approach are proposed. The objective function is to maximize the total priority scores of the patients assigned to the surgical scheduling blocks over a given planning horizon. An adaptive ILP model is also proposed to solve the SCAP, taking into consideration the dynamics of the waiting list. The computational experiments are conducted using a set of random data to evaluate the performance of the proposed MSS-SCAP model and heuristic algorithm, in terms of solution quality and computation time. Our numerical results indicate that the proposed ILP is capable of yielded optimal solutions for the small-scale instances and near-optimal solutions for medium-size instances within 3,600 seconds. The proposed heuristic algorithm can generate quality solutions within 2 seconds for large-scale instances.

Keywords Operating room scheduling · Patient priority · Master surgical schedule · Surgical case assignment problem

O. Mashkani (✉) · F. J. Hwang · A. Salehipour
School of Mathematical and Physical Sciences, University of Technology Sydney,
Ultimo, NSW, Australia
e-mail: Omolbanin.Mashkani@student.uts.edu.au

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_11

11.1 Introduction

In recent years, the efficiency of the Australian health care system has become an important area of interest, largely due to the growing costs in health care. Taking into account ageing population, increasing burden of chronic conditions, and growing patients' expectations of health services, it is anticipated that the Australian Government expenditure on health care alone will increase from 4.2% of gross domestic product (GDP) in 2014–2015 to 5.7% in 2054–2055, or 260 billion in current dollars [3]. Although the Australian health care system generally has satisfactory outcomes by international standards, recent studies have revealed that the efficiency of the health sector could be improved by 20% through making the best use of the available resources [4].

In a health care system, different stockholders have different interests and priorities. Although health care administrations' goal is to decrease costs, patients expect to receive high-quality services as well as low charges and short waiting times. As a result, to address the efficiency gap and simultaneously satisfy all stockholder expectations, one solution is to employ systematic and evidence-based approaches, which facilitate significant improvements in quality, efficiency, safety and other aspects of operations [5]. Despite the fact that the focus of improving health care systems has largely been on policy, managing hospital operations plays an important role in enhancing the efficiency. Hospital managers are directly involved in actual care rather than the context [6]. Among all hospital departments, the operating theatre (OT) department, which usually consists of several operating rooms (ORs), is the most crucial and costliest due to its operational complexity and expensive resources. It is estimated that 60–70% of all hospital admissions are surgical and the OT department accounts for more than 40% of the total expenses of a hospital [7]. Therefore, any improvement in OT efficiency contributes to having an efficient health care delivery system as a whole. Since the current surgical scheduling and planning approaches do not live up to the hospital management expectations, the OT planning and scheduling problems have attracted the attention of many researchers recently [8]. In general, the OT planning and scheduling problem is a highly complex problem that entails the assignment of OT resources, such as rooms, equipment, nurses and surgeons to patients with the aim of improving efficiency and reducing patients' waiting time in a way that balances all stockholder's expectations. The problem is getting even more complex due to different patient characteristics, restricted capacities of upstream and downstream departments and inherent uncertainty of the surgical procedures. In the literature, surgeons are classified into the surgeon groups or surgical specialties if they are homogeneous and have the same medical and procedural requirements [11]. On the other hand, although elective surgeries can be planned in advance, non-elective surgeries are unexpected.

11.1.1 Literature Review

The characteristics of the different hospitals under study and the different national realities contribute to the diversity of the literature in OT planning and scheduling problems. From the structural point of view, the OT planning and scheduling decisions are made in three hierarchical levels including strategic, tactical and operational. These decision levels depend on each other, as the outcome of each level can affect the decisions on other levels in addition to being the input for the next level [7]. in the following, the main characteristics of these hierarchical levels are introduced.

At the strategic level, the main focus is to find out how many time blocks/slots should be assigned to each surgical specialty to find the ideal composition and volume of patients in a hospital [7, 9]. A time block is defined as the smallest time unit for which an operating room can be assigned to a specialty [20]. To this end, strategic planning is primarily a resource allocation problem and known as the case-mix planning problem (CMPP). For a detailed survey of literature on the CMPP, we refer the interested reader to Hof et al. [19].

At the tactical level, the capacity of ORs per day is shared among a variety of surgical specialties in order to provide a cyclic timetable, which is called the master surgical schedule (MSS) [14]. The main goal of an MSS is to assign surgical specialties, and not individual patients/surgeons, to time blocks [12]. To develop an MSS, historical data and actual/forecast patients' demand, in the form of waiting lists, are utilized as critical inputs. In the literature, to construct an MSS three main strategies have been used, including block scheduling, open scheduling and modified block scheduling. In the block scheduling strategy, time blocks are assigned to surgical specialties, which can arrange their surgical cases only in their own blocks. In the open scheduling strategy, surgeons from different specialties can perform surgeries in the same time block. Moreover, modified block scheduling strategy is a mix of block and open scheduling strategies, which reserves some of the time blocks and assigns others to patients or specialties using an open scheduling strategy [11]. Although the open scheduling strategy is more flexible and provides a better assignment of the surgical cases in comparison with block scheduling, it is an uncommon strategy and rarely used in the health care industry. Hence, despite potential inefficiencies as a result of unbalanced block schedules, block scheduling is widely accepted to generate the MSS, due to its simplicity for both surgeons and managers [18].

In the literature, developing the MSS has been investigated as a combinatorial optimization problem. The main objective of this optimization problem is to provide an OT plan that optimizes the OT and surgical resource allocation and minimize the patients' waiting time. A variety of constraints affect the development of MSS such as availability restrictions of medical staff and equipment, capacity limitation of resources, e.g. regular opening hours, number of upstream and downstream resources, and the uncertainty of surgical procedures as well as a restriction on the number of time blocks assigned to each specialty as the result of CMPP [2, 7, 12].

At the operational level, the assignment of an operating room and operating time to each patient, as well as sequencing surgeries in each operating room are determined over a short-term planning horizon [9]. To develop an operational plan a wide range of constraints should be taken into account. For example, structural constraints, which ensure a non-overlapping of surgeries in the same room or surgeons in different rooms at the same time, and resource constraints such as daily capacity restriction in each operating room are among the most important ones [1]. In the literature, generally, the OT planning and scheduling problems at the operational level have been solved as two main subproblems including the advance scheduling and the allocation scheduling. The advance scheduling, which is also called the surgical case assignment problem (SCAP) seeks to assign each patient to an operating room and a particular day for surgery over a planning horizon of 1–2 weeks. The allocation scheduling, also referred to as the surgical case sequencing problem (SCSP), concentrates on the timing aspects and sequencing of the assigned surgeries within each OR [15]. To address the subproblems at the operational level, a variety of solution approaches have been developed in the literature utilizing mathematical programming techniques, simulation and scenario-based analysis and analytical procedures. They mainly have been solved as a combinatorial optimization problem with the aim of achieving a trade-off among different stockholders' interests [15, 17, 21].

11.1.2 Contribution of this Research

Much literature has investigated just one decision level of OT planning and scheduling problems. In other words, they solved the problem using multi-stage approaches, with each stage dealing with just one decision level [11]. The main reason is that solving the overall problem of all decision levels as multi-stage problems decreases the complexity of the problem. However, the three hierarchical decision levels depend on each other and the outcome of each level can be utilized as input to the other levels. Therefore, solving the problems on different decision levels concurrently provides more effective procedures and solutions. In some studies, all decision levels were investigated at the same time [10]. Other studies coped with the problems at tactical and operational levels simultaneously [2, 11, 16]. Despite the fact that recent studies have focused on the integrated MSS and SCAP scheduling problems at both tactical and operational levels, the proposed exact methods could not live up to medium- or large-size instances. This study investigates the integrated MSS-SCAP based on patient priority, which is the indicator of a patient's surgery urgency. The main novelty of the integrated MSS-SCAP problem in this paper is the consideration of three fundamental factors altogether including consideration of patients' priorities, strategic decisions and solving both MSS and SCAP problems concurrently. An ILP model, called MSS-SCAP model, is proposed to produce robust surgical scheduling. The MSS-SCAP model can optimally assign time blocks to specialties and determine the OR and surgery date for each patient over the planning horizon of medium term. To cope with the large-size instances, a heuristic approach is developed to provide high-quality solutions for the MSS-SCAP. Then to adapt operation-related dynamics and any unpredicted changes in the waiting list, given an MSS, an adaptive ILP is proposed to solve the SCAP as the multiple knapsack problem (MKP) over the planning horizon of one to several weeks.

The remainder of this paper is organized as follows. Section 11.2 introduces the problem statement, the assumptions as well as the mathematical models. In Sect. 11.3, the details of heuristic approach are provided. Computational experiments, which indicate the performance of the MSS-SCAP model and the heuristic approach, are reported in Sect. 11.4. Finally, the conclusion of the study and further research directions are presented in Sect. 11.5.

11.2 Problem Statement and Mathematical Models

In this section, the problem definition and assumptions are given. The mathematical models are then provided to solve the MSS-SCAP and SCAP. It should be noted that for the sake of integrity and simplicity, the symbols and definitions similar to [2] are used.

11.2.1 Problem Definition

The goal of the integrated MSS-SCAP is to provide a cyclic time table that allocates the time blocks of each OR and each day to the specialties, in addition to the determination of the OR and the surgery date per patient over the planning horizon. In this study, the block scheduling strategy is utilized to allocate specialties to time blocks. Although the duration of time blocks can vary, it is determined in advance. Pre-emption is not allowed, which means that, once a surgery starts, it cannot be interrupted. It is assumed that all surgeries of each specialty can be performed by any surgeon of that specialty and all ORs can be used by all specialties. The number of nurses, upstream and downstream resources such as number of ICU and ward beds are enough and do not force any bottleneck or restriction on the planning process. It is also assumed that the minimum and maximum OR times to be assigned to each specialty, as the output of CMPP at the strategic level, are given. Moreover, the surgeries can be scheduled during the working days from Monday to Friday (we only consider elective surgeries), which means that each week of the planning horizon includes 5 days. Table 11.1 summarizes the notations used in the paper.

Table 11.1 Nomenclature

Notation	Definition
$B_{\max}$	Number of time blocks in a working day
$D_{\max}$	Number of working days in the planning horizon
$P_{\max}$	Number of patients on the waiting list at the start of the planning horizon
$S_{\max}$	Number of surgical specialties
$T_{\max}$	Number of operating rooms
B	The set of time blocks in a working day, indexed by $b \in B = \{1, \ldots, B_{\max}\}$
D	The set of working days in the time horizon, indexed by $d \in D = \{1, \ldots, D_{\max}\}$
P	The set of patients on the waiting list at the start of the planning horizon, indexed by $p \in P = \{1, \ldots, P_{\max}\}$
S	The set of surgical specialties, indexed by $s \in S = \{1, \ldots, S_{\max}\}$
T	The set of operating rooms, indexed by $t \in T = \{1, \ldots, T_{\max}\}$
L_p	The surgery duration of patient p in minutes, $\forall p \in P$
τ_b	The length of time block b in hours, $\forall b \in B$
N_s^-	The minimum number of OR hours assigned to specialty s over the planning horizon, $\forall s \in S$
N_s^+	The maximum number of OR hours assigned to specialty s over the planning horizon, $\forall s \in S$
I_{ps}	Binary parameter, which is 1 if surgery of patient p can be performed by specialty s, and 0 otherwise, $\forall p \in P, s \in S$
ρ_p	The priority score of patient p, $\forall p \in P$
P_s	The set of patients in the current waiting list of the surgical specialty s
Q_{tsdb}	Binary parameter, which is 1 if block b of operating room t is assigned to specialty s on day d, and 0 otherwise, $\forall t \in T, s \in S, d \in D, b \in B$
X_{tsdb}	Binary decision variable, which is 1 if block b of operating room t is assigned to specialty s on day d, and 0 otherwise, $\forall t \in T, s \in S, d \in D, b \in B$
Y_{tpdb}	Binary decision variable, which is 1 if block b of operating room t is assigned to operate surgery of patient p on day d, and 0 otherwise, $\forall t \in T, p \in P, d \in D, b \in B$

11.2.2 *The MSS-SCAP Model*

In this section, the MSS-SCAP model is proposed to allocate the time blocks to specialties and assign patients to ORs as well as days over the planning horizon. Despite the fact that hospital managers assign time blocks to the specialties based on the equity and fairness criteria [10], the priority of a patient contributes to the urgency and importance of performing his/her surgery. In other words, in assignment of patients to ORs and dates, the patient with higher priority have precedence to be operated, which means that the larger priority score, the higher priority of surgery. Thus, to generate a distribution of time blocks among the specialties, patient priority should be taken into account. Nevertheless, only a few researches have taken patient prioritization into consideration [18]. To this end, the objective function of this study is to maximize the summation of assigned patient priority scores to ORs. The MSS-SCAP model is formalized as below:

$$\text{Maximize} \quad \sum_{t \in T} \sum_{p \in P} \sum_{d \in D} \sum_{b \in B} \rho_p Y_{tpdb}$$

s.t.

$$\sum_{s \in S} X_{tsdb} \leq 1 \qquad \forall t \in T, d \in D, b \in B \tag{11.1}$$

$$I_{ps} Y_{tpdb} \leq X_{tsdb} \qquad \forall t \in T, p \in P, s \in S, d \in D, b \in B \tag{11.2}$$

$$\sum_{t \in T} \sum_{d \in D} \sum_{b \in B} Y_{tpdb} \leq 1 \qquad \forall p \in P \tag{11.3}$$

$$\sum_{t \in T} \sum_{d \in D} \sum_{b \in B} \tau_b X_{tsdb} \leq N_s^+ \qquad \forall s \in S \tag{11.4}$$

$$\sum_{t \in T} \sum_{d \in D} \sum_{b \in B} \tau_b X_{tsdb} \geq N_s^- \qquad \forall s \in S \tag{11.5}$$

$$\sum_{p \in P} L_p Y_{tpdb} \leq 60\tau_b \qquad \forall t \in T, d \in D, b \in B \tag{11.6}$$

$$X_{txdb}, Y_{tpdb} \in \{0, 1\} \qquad \forall t \in T, p \in P, d \in D, b \in B \tag{11.7}$$

The objective function is to maximize the summation of priority scores of assigned patients to ORs over the planning horizon. Using constraint (11.1), it is not possible to share a block between different specialties as per the block scheduling strategy. Based on constraint (11.2), patient p can be operated in block b of an operating room t during day d only if that time block is assigned to its specialty. Constraint (11.3) determines that a patient can be operated at most once during the planning horizon. Constraints (11.4) and (11.5) enforce the restrictions on the maximum and minimum numbers of hours that can be assigned to each specialty as the result of strategic decisions. Using Constraint (11.6), the total processing time of all assigned patients to a time block must not be greater than the duration of that time block. Constraint (11.7) is related to the definition of binary decision variables.

11.2.3 The Adaptive SCAP Model

An MSS usually is constructed to cover a planning horizon of one to several months. Then, considering this MSS, hospital administrations determine the staff rostering and equip ORs with required instruments. Making these decisions and providing the equipment entail spending lots of time and negotiating with different surgical specialties as well as going to great expense. Hence, hospital administrations do not tend to change the MSS over the planning horizon of medium term. On the other hand, in the real world, even excluding the uncertainty factors, the waiting list is dynamic due to the arrival of other elective patients with high priorities. One of the good approaches to handling this dynamic process is to generate a new SCAP solution, for each planning horizon of one to several weeks. Therefore, the MSS solution, which is provided by MSS-SCAP model, is kept constant but the SCAP solution will be updated. In other words, given the MSS, the SCAP solution takes into account new elective patients and will be updated whenever it is necessary.

In summary, at the first stage, the MSS-SCAP is solved and the assignment of time blocks to surgical specialties is considered as the MSS solution. Then, given this MSS solution, it is assumed that the SCAP solution can be updated to adapt the real-world dynamic conditions. Therefore, the overall SCAP is decomposed into several subproblems similar to the MKP, one for each surgical specialty, in which the patients correspond to the items and the blocks to the knapsacks.

The adaptive SCAP model per specialty s is illustrated as follows. Note that the parameter Q_{tsdb} is determined by the MSS solution, which can be the result of the MSS-SCAP model.

$$\text{Maximize} \quad \sum_{t\in T}\sum_{p\in P}\sum_{d\in D}\sum_{b\in B}\rho_p Y_{tpdb}$$

s.t.

$$\sum_{t\in T}\sum_{d\in D}\sum_{b\in B} Y_{tpdb} Q_{tsdb} \leq 1 \qquad \forall p \in P_s \tag{11.8}$$

$$\sum_{p\in P} L_p Y_{tpdb} I_{ps} \leq 60\tau_b Q_{tsdb} \qquad \forall t \in T, d \in D, b \in B \tag{11.9}$$

$$Y_{tpdb} \in \{0, 1\} \qquad \forall t \in T, p \in P, d \in D, b \in B \tag{11.10}$$

The objective function is to maximize the total priority scores of assigned patients to ORs and dates over the planning horizon of one to several weeks. Constraint (11.8) indicates that a patient can be operated at most once during the planning horizon. Constraint (11.9) restricts the processing time of all assigned patients, which belong to specialty s, to a time block that must not be greater than that time block duration. Based on constraint (11.10), all decision variables are binary.

Both of the proposed mathematical models focused on maximizing the total priority scores of assigned patients. However, in the real world, patients with higher priorities should be scheduled as soon as possible.

11.3 The Proposed Heuristic Algorithm

In order to generate high-quality solutions for MSS-SCAP in a reasonable amount of time and for large-scale instances, this study proposes a heuristic algorithm, which consists of the following steps:

Step 1: Sort all patients in a non-increasing order of their priority scores.
Step 2: Select patient p among unscheduled patients with the highest priority score. If there are some patients with the same priority scores, select the one with longer processing time. If their processing times are equal, select one of them randomly. Schedule the selected patient considering the following rules:

- Using the first-fit strategy, assign the patient to the first available time block b, which belongs to the specialty of patient p. If the patient p is the first patient who is assigning to time block b, then by assigning this patient to time block b, its specialty is also assigned to that time block.
- In each time block, only the patients with the same specialty s can be assigned.
- By assigning patient p to a time block b, the total processing time of all patients assigned to that block must not exceed the capacity of that block.
- The total amount of assigned hours to specialty s must not exceed the maximum number of OR hours that can be assigned to that specialty.
- The time blocks containing assigned patients are considered prior to the unoccupied ones. This criterion seeks to reduce idle time and minimize the number of open ORs.

Step 3: Repeat Step 2 until all patients are scheduled or there is no available time in blocks over the planning horizon.

The numbers of operations in Steps 1 and 2 are $O(P_{\max} \log P_{\max})$ and $O(B_{\max} D_{\max} P_{\max})$, respectively. Thus, the run time of the proposed heuristic is $O(P_{\max} \log P_{\max} + B_{\max} D_{\max} P_{\max})$.

11.4 Computational Results

To evaluate the performance of the proposed mathematical models and the heuristic algorithm, a numerical study is designed. The models were implemented via Gurobi 8.0.0 and the heuristic algorithm was coded by using Python Anaconda 3.6. The numerical study is carried out on a PC equipped with Intel Core i5 3.2 GHz CPU and 8 GB of RAM under Linux Ubuntu operating system. To generate the instances, some of the data are adopted from a data set provided by Spratt and Kozan [2]. The number of patients in the data set is 2802. Thus, in the present study the number of patients

Table 11.2 Performance comparisons between the MSS-SCAP model and heuristic algorithm for $T_{\max} = 2$

$P_{\max}$	$S_{\max}$	$Time_H$	$Time_{opt}$	APE	Gap
20	1	0	0	0	0
50	3	0	226	0.01	0
100	4	0.01	1800	0.03	0
150	5	0.01	2700	0.05	0
200	5	0.02	>3600	0.1	0.02
300	5	0.03	>3600	0.16	0.02
400	5	0.05	>3600	0.21	0.02
500	7	0.06	>3600	0.23	0.02
1000	8	0.17	>3600	0.36	0.02
1500	8	0.3	>3600	0.42	0.02
2000	10	0.49	>3600	0.44	0.03
2802	12	0.85	>3600	0.52	0.04

is selected from the set {20, 50, 100, 150, 200, 300, 400, 500, 1000, 1500, 2000, 2802}. In the data set of [2], the ratio of patients to specialties, i.e. I_{ps}, is given. As a result, the number of specialties is selected from the set {3, 5, 8, 10, 12}. The number of operating rooms is selected from {2, 4, 5, 10, 15, 20}. The number of days to plan the surgeries is selected from {5, 10, 15, 20}, meaning that the planning horizon ranges between 1 and 4 weeks. Each day of planning horizon consists of two time blocks and each block is 5 hours. Furthermore, surgery durations per each specialty are generated randomly from the lognormal distribution (the mean and variance are given in the data set [2]). Since the duration of each time block is 5hours, the lognormal distribution is truncated at 5hours to ensure that each surgery fits in a time block. The integer priority scores were also generated from a discrete uniform distribution of [1,100].

To evaluate the capability of the mathematical models and the heuristic algorithm, two types of experiments were performed. The first experiment includes only two operating rooms. Table 11.2 shows the results of this experiment. The average percentage of error (APE) for the heuristic is calculated as $\frac{F_{Opt}-F_H}{F_{Opt}}$, where F_{Opt} is the objective value of the model as reported by Gurobi, and F_H is the objective value of the heuristic algorithm. The solution gap, which is provided by Gurobi, is presented as Gap. In addition, the average computation time of the heuristic and the model are presented as $Time_H$ and $Time_{Opt}$ (the maximum runtime for Gurobi is set to 3,600 seconds; in column $Time_{Opt}$, − means that the time limit has been reached).

For the second experiment, the computational efficiency of the heuristic was investigated by increasing the number of patients and ORs. Table 11.3 shows the average times of the heuristic algorithm for different numbers of patients and ORs. As it is indicated, the average execution time of the heuristic algorithm is less than two seconds even for $P_{\max} = 2802$ and $T_{\max} = 20$. Therefore, the heuristic algorithm is quite efficient to solve the large-scale instances of the MSS-SCAP.

Table 11.3 Average times of the heuristic algorithm for different numbers of $P_{\max}$ and $T_{\max}$

$P_{\max}$	$T_{\max} = 2$	$T_{\max} = 4$	$T_{\max} = 5$	$T_{\max} = 10$	$T_{\max} = 15$	$T_{\max} = 20$
20	0	0	0	0	0	0
50	0	0	0	0	0	0
100	0.01	0.01	0.01	0.01	0.01	0.01
150	0.01	0.01	0.01	0.01	0.01	0.01
200	0.02	0.02	0.02	0.02	0.02	0.02
300	0.03	0.04	0.04	0.04	0.05	0.05
400	0.04	0.06	0.06	0.07	0.08	0.08
500	0.06	0.09	0.09	0.11	0.11	0.11
1000	0.16	0.23	0.25	0.32	0.35	0.36
1500	0.3	0.41	0.44	0.58	0.46	0.67
2000	0.49	0.36	0.68	0.87	0.97	1.04
2802	0.84	1.06	1.11	1.44	1.66	1.68

11.5 Conclusion and Future Work

This paper investigates the OR scheduling problem at tactical and operational levels concurrently, called MSS-SCAP, for maximizing the total priorities of assigned patients. The block scheduling strategy is used to allocate time blocks to specialties. To solve the MSS-SCAP, an ILP model and a heuristic algorithm have been proposed. In addition, an adaptive ILP model is suggested to solve the SCAP and cope with the dynamics of the real-world waiting list, given an MSS. To evaluate the proposed MSS-SCAP model and the heuristic algorithm, a numerical study has been developed. The computational results have shown that the heuristic algorithm has the capability to solve large-scale instances efficiently. Further research works will include the uncertainty of surgery durations and the restriction on the downstream facilities like the number of beds in the ICU Department. In addition, the problem can be extended to include a restriction on the availability of surgical teams on the days of surgery.

References

1. Addis B, Carello G, Tànfani E (2014) A robust optimization approach for the Advanced Scheduling Problem with uncertain surgery duration in Operating Room Planning- an extended analysis, working paper or preprint
2. Agnetis A, Coppi A, Corsini M, Dellino G, Meloni C, Pranzo M (2012) Long term evaluation of operating theater planning policies. Oper Res Health Care 1:95–104
3. Agnetis A, Coppi A, Corsini M, Dellino G, Meloni C, Pranzo M (2014) A decomposition approach for the combined master surgical schedule and surgical case assignment problems. Health Care Manag Sci 17:49–59

4. Aringhieri R, Landa P, Soriano P, Tnfani E, Testi A (2015) A two level metaheuristic for the operating room scheduling and assignment problem. Comput Oper Res 54:21–34
5. Australian Government, Productivity Commission (2006) Potential Benefits of the National Reform Agenda. Research Paper, Canberra, Australia
6. Australian Government, Productivity Commission (2015) Efficiency in Health. Research Paper, Canberra, Australia
7. Erdogan SA, Denton BT (2011) Surgery planning and scheduling. American Cancer Society, Wiley Encyclopedia of Operations Research and Management Science
8. Fei H, Chu C, Meskens N (2008) Solving a tactical operating room planning problem by a column-generation-based heuristic procedure with four criteria. Ann Oper Res 166:91
9. Guerriero F, Guido R (2011) Operational research in the management of the operating theatre: a survey. Health Care Manag Sci 14:89–114
10. Guido R, Conforti D (2017) A hybrid genetic approach for solving an integrated multi-objective operating room planning and scheduling problem. Comput Oper Res 87:270–282
11. Hof S, Fügener A, Schoenfelder J, JO B (2017) Case mix planning in hospitals: a review and future agenda. Health Care Manag Sci 20:207–220
12. Hussung T (2016) The role of hospital management in transforming healthcare. Husson University, Bangor, Canada
13. Jebali A, Diabat A (2017) A Chance-constrained operating room planning with elective and emergency cases under downstream capacity constraints. Comput Ind Eng 114:329–344
14. Kaplan G, Bo-Linn G, Carayon P, Pronovost P, Rouse W, Reid P, Saunders R (2013) Bringing a Systems Approach to Health. National Academy of Medicine, Discussion Paper, Washington, America
15. Koppka L, Wiesche L, Schacht M, Werners B (2018) Optimal distribution of operating hours over operating rooms using probabilities. Eur J Oper Res 267:1156–1171
16. Ma G, Beliën G, Demeulemeester E, Wang L (2009) Solving the strategic case mix problem optimally by using branch-and-price algorithms, In: proceeding paper
17. Marques I, Captivo ME (2015) Bicriteria elective surgery scheduling using an evolutionary algorithm, operations research for health care, ORAHS 2014. In: The 40th international conference of the EURO working group on operational research applied to health services, vol 7, pp 14–26
18. Marques I, Captivo ME, Vaz Pato M (2015) A bicriteria heuristic for an elective surgery scheduling problem. Health Care Manag Sci 18:251–266
19. Sieber T, Leibundgut D (2002) Operating room management and strategies in Switzerland: results of a survey. Eur J Anaesthesiology 19:415–423
20. Spratt B, Kozan E (2016) Waiting list management through master surgical schedules: a case study. Oper Res Health Care 10:49–64
21. Testi A, Tanfani E, Torre G (2007) A three-phase approach for operating theatre schedules. Health Care Manag Sci 10:163–172

Chapter 12
Analyzing Fantasy Sport Competitions with Mixed Integer Programming

Steven J. Edwards

Abstract A fantasy sport competition is an online competition in which participants act as the coach and selector of their own fantasy team of real players. These competitions are remarkably popular with currently over 5.5 million teams in the major competition of the English Premier League. A fantasy team scores points based on the statistical performances of the team's players in their real-world sporting matches. The objective for each coach is to finish the season with the highest total number of points. During the season, coaches must manage a budget as well as trade players in and out of the team subject to a number of constraints. Due to their well-defined nature, as well as the simple objective function, these competitions lend themselves very naturally to analysis by Mixed Integer Programming (MIP). In this paper, we consider three different problems for the 2018 season of the AFL SuperCoach competition, modelling and solving each with MIP. The aim of each problem is to highlight the gap between what was achieved by real players and what was theoretically possible. The first problem is to determine all the decisions that a coach should have made to obtain the highest score possible. The second problem is to determine the lowest starting budget from which it would have been possible to win the competition. The third problem is to determine whether it would have been possible for a team that was set up at the start of the competition and completely forgotten about to win the competition.

Keywords Mixed integer programming · Mathematical programming · Fantasy Football · Decision Support

S. J. Edwards (✉)
Monash University, Clayton, CA, Australia
e-mail: steven.edwards@monash.edu

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_12

12.1 Introduction

A fantasy sport competition is a competition in which participants act as the coach and selector of their own fantasy sporting team. These fantasy sporting teams consist of real-world players from a professional sporting league such as the English Premier League (EPL), National Basketball Association (NBA), National Football League (NFL), or the Australian Football League (AFL). Players obtain scores based on the statistics of their performance in the real-world league games. Fantasy teams then obtain scores based on the scores of the individual players in their team.

There are many different formats of fantasy sport competitions. In this paper, we consider the format that is typically referred to as the *Classic* format. In this format, a fantasy competition is run in parallel to a real professional league. Each of the fantasy competitions in the Classic format have their own unique set of rules; however, in general tend to have the following in common. Coaches are given a starting budget with which they select a side. Players have different prices depending on their previous performances and this price changes throughout the season. To help manage injuries and poor form, coaches are allowed to make a certain number of trades throughout the season as long as the trades respect their remaining budget.

The Classic format contrasts to the two other most popular formats: the *Draft* format and *Daily* format. The key difference between the Classic and Draft formats is that for the latter at the start of the season all the coaches in the same fantasy league perform a draft. Teams take it in turns to select players. Hence no two teams in the same league can have ownership over the same player. Furthermore, players are traded between teams only at the approval of both coaches in the trade. As the choices of coaches are influenced by the choices of other coaches in the same league, the analysis completed in this paper is not appropriate for competitions in the Draft format.

Daily sport competitions are an accelerated version of traditional fantasy competitions. Competitions are conducted over a short time period such as a week or a single day. Within daily competitions, there are many different formats such as a daily version of both the Classic and Draft formats. Due to the short time period these, competitions do not need to consider aspects of the Classic format such as trades, price changes and so on.

Due to their well-defined nature, fantasy sport competitions lend themselves very naturally to techniques used in Operations Research. Hunter et al. [5] consider the problem of selecting a portfolio of entries of fixed cardinality for a winner take all contest such that the probability of one of the entries winning is maximized and develop a greedy integer programming formulation, which they apply to daily fantasy sport contests. They show that this approach works well in practice and even comes first place in a number of competitions with thousands of entries. Newell [6] develops a stochastic integer program that optimizes the expected payout in a tiered daily fantasy sport contest. This model is based on assigning a normal distribution of the score of each player and aims to determine the team of players with the highest mean.

In this paper, we use Mixed Integer Programming (MIP) to analyze a fantasy sport competition in retrospect to highlight the gap between what was achieved by players and what was theoretically possible. We consider the fantasy competition of the AFL, SuperCoach (SC). In Australia this competition is remarkably popular with just under 200k (199, 243) teams in the 2018 competition. This paper considers the following three problems:

1. What was the highest score that was theoretically possible to obtain?
2. What was the lowest starting budget from which it was possible to win the competition?
3. Was it possible that a team that was set up at the start of the competition and completely forgotten about wins the competition? These teams are commonly referred to as 'ghost' teams.

Due to their popularity, these competitions provide a great opportunity to demonstrate the sort of problems that MIP and Operations Research, in general, can solve to a general audience. This research has already resulted in a number of articles written in newspapers and on popular websites [1–4]. These articles provide a more detailed analysis of the results of the problems. This paper focusses on the MIP models that were developed to solve these problems.

12.2 Problem Description

The rules of SC are as follows: Each coach is in charge of selecting and managing their own team of players. A team consists of 30 players spread across four lines of positions: defence (DEF), midfield (MID), ruck (RUC) and forward (FWD). These lines of positions are an intuitive generalization of the positions in AFL. In each line, the position is divided into scoring positions and substitute positions to make a total of eight different positions. Each of the eight positions have a certain number of spots that need to be filled by players. The setup of these positions is shown in Fig. 12.1.

Coaches select their 30-player team from any of the current professional AFL players. In 2018, there were 806 players to choose from. Players are only eligible to be placed in certain positions. The eligible positions try to replicate the position that a player plays in real life. For example, a player who plays only in the ruck in real life will only be eligible to be placed in the ruck slots in the fantasy competition. Likewise, a player who plays sometimes in the forward line and sometimes in the midfield in real life, may be eligible to be placed in both the midfield and forward lines in the fantasy competition. Note that no player is eligible to be placed in more than two positions, with the vast majority of players only being eligible to play in a single position. These position eligibilities do not change throughout the season.

A price is assigned to each of the players at the start of the season. This price is based on how well the players have scored in previous seasons as well as a number of other factors. The players who are expected to score the most are assigned a high

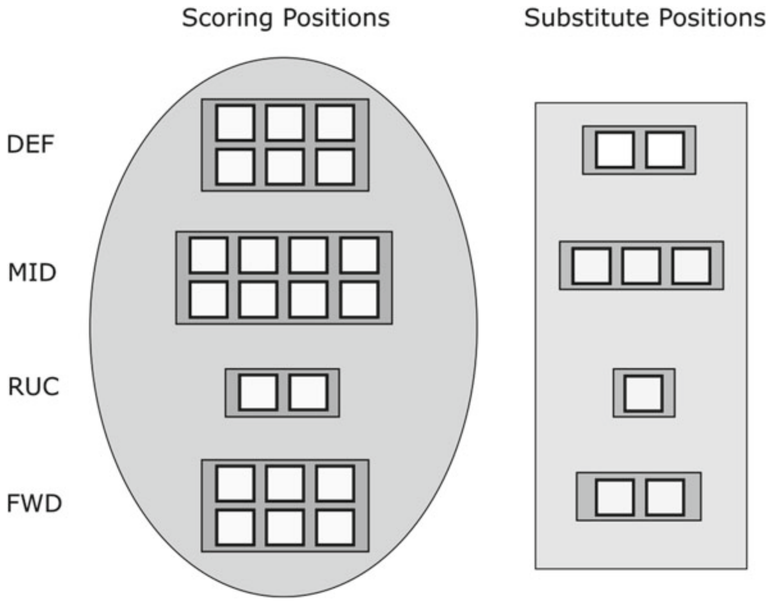


Fig. 12.1 Summary of the different positions of a SC team. Positions are separated into four lines and further divided into scoring and substitute positions

price, up to ≈$700 k, whereas players who are yet to play their first AFL game are assigned a low price, ≈$100 k. At the start of the season, each coach is allocated a starting budget of $10 m. This budget forces the coaches to select a range of premium, mid-priced and rookie players.

Each round players score points. These points are calculated based on the on-field statistics of the players in the real-world games, which act as a proxy to quantify how well an individual player performed. If a player does not play then they obtain 0 points. The very best players average around 130 points per game, with scores over 100 typically considered good performances.

Each round the fantasy team obtains a score based on the scores of the individual players in the team. In general the score of the team is determined based on the scores of the players in the scoring positions. There are a number of exceptions to this that will be discussed in the following sections.

Before each round, the coach will assign one of the players in the team to be the captain, and one player to be the vice captain. Both of these players must be in a scoring position. The points scored by the captain are counted twice in the team's score. In the situation where the captain scores zero points, the points scored by the vice captain are counted twice. If the captain scores above zero points then the vice captain is no different to any other on-field player.

Before each round the coach can select up to four players in substitute positions to be emergency. The points scored by an emergency can be counted if a player in a scoring position on the same line as the emergency scores zero points. In the situation where multiple emergencies are selected on the same line and fewer players in the scoring positions score zero points, the points of the lower scoring emergencies will be counted. For example, if two emergencies are selected in the midfield, where one scores 100 points the other scoring zero, if only one player in the scoring midfield position scores zero points then the points from the emergency who scored zero will count and not the emergency who scored 100.

During the season, the price of a player changes depending on their recent scores. The pricing formula is based on the average of each player's previous three non-zero scores. This average is then compared to the score expected by someone with the current price of the player. The player's price changes proportionate to the difference between the player's three round average and this indicator score. A common strategy used by coaches is to trade in players at a low price and trade them out at a high price and then use the profit to invest in other players.

To account for injuries and bad form, coaches can make up to 30 trades throughout the season. A trade means that a player in the fantasy team is traded out and replaced by a player who was not previously in the team. When a player is traded out, the remaining budget is increased by the current price of the player. Whereas when a player is traded in, the remaining budget is decreased by the current price of the player. Thus due to the different prices of the players, the ability to perform a trade depends on whether the coach has enough remaining budget. In general, coaches can only complete at most two trades per round.

Throughout the real AFL season each team has a bye round where the team does not play. These bye rounds occur across three rounds in the middle of the season with the aim of giving the players a rest. The rules of the fantasy competition are slightly different during these bye rounds. Before each bye round, coaches are allowed to make up to three trades. These trades are still counted towards the season limit of 30 trades. Furthermore, during the bye rounds only the points scored from 18 players will be counted—instead of the typical limit of 22 (the scoring positions). In the situation where a team has more than 18 players whose points would normally be counted the highest 18 scores are counted.

The objective of the coaches is to finish the season with the most number of points.

12.3 Mathematical Models

First, we introduce the notation used by all the mathematical models. Table 12.1 defines all of the required sets. Table 12.2 defines the notation used for the parameters.

Table 12.1 The definition of the different set notation

Index	Set	Description
p	P	Set of all players
q	Q	Set of all positions, $Q = Q^{\text{sub}} \cup Q^{\text{score}}$
	Q^{score}	Set of scoring positions $Q^{\text{score}} = \{DEF, MID, RUC, FWD\}$
	Q^{sub}	Set of sub positions $Q^{\text{sub}} = \{DEF^{\text{sub}}, MID^{\text{sub}}, RUC^{\text{sub}}, FWD^{\text{sub}}\}$
p	P_q	Set of all players $p \in P$ who are allowed to play in position $q \in Q$
q	Q_p	Set of all positions $q \in Q$, where player $p \in P$ is allowed to play. Note $Q_p^{\text{sub}} = Q_p \cap Q^{\text{sub}}$ and $Q_p^{\text{score}} = Q_p \cap Q^{\text{score}}$
r	R	Set of all rounds $R = \{1, \ldots, 23\}$

Table 12.2 The definition of the different parameter notations

Parameter	Description
$\psi_{p,r}$	The points scored by player $p \in P$ in round $r \in R$
$v_{p,r}$	The price (value) of player $p \in P$ in round $r \in R$
B	The starting budget for the season
C_q	The number of available spots (capacity) of position $q \in Q$
X_r	The number of players whose scores count towards the score of the team in round $r \in R$
T^{total}	The maximum number of trades that can be used across the season
T_r	The maximum number of trades that can be used in round $r \in R$

12.3.1 *Optimal Team*

In this section, we introduce a MIP model that determines all the decisions a coach should have made to obtain the best score possible (OT-MIP). The decision variables are defined in Table 12.3. It is important to note that the OT-MIP does not need to consider the vice captain assignment, as an optimal strategy is to always assign the captain to the highest scoring player in the team each round. Likewise, the OT-MIP does not need to consider emergencies in the substitute positions, as an optimal strategy is to always have the highest scoring players in scoring positions. These two observations greatly simplify the problem.

The OT-MIP can now be expressed as follows:

Table 12.3 Decision variables used in the optimal team model

Variable	Type	Description
$x_{p,q,r}$	Binary	If player $p \in P$ is in position $q \in Q$ for round $r \in R$
$\overline{x}_{p,r}$	Binary	If the score of player $p \in P$ is included in round $r \in R$
$c_{p,r}$	Binary	If player $p \in P$ is captain for round $r \in R$
$t^{\text{in}}_{p,r}$	Binary	If player $p \in P$ is traded into the team for round $r \in R : r > 1$
$t^{\text{out}}_{p,r}$	Binary	If player $p \in P$ is traded out of the team for round $r \in R : r > 1$
b_r	Cont.	Remaining budget at round $r \in R$

$$\max \quad \sum_{p\in P}\sum_{r\in R}\psi_{p,r}(\overline{x}_{p,r}+c_{p,r}) \tag{12.1}$$

$$\text{s.t.} \quad \sum_{p\in P}\sum_{r\in R:r>1} t^{in}_{p,r} \le T^{\text{total}} \tag{12.2}$$

$$\sum_{p\in P} t^{in}_{p,r} \le T_r \qquad \forall r \in R : r > 1 \tag{12.3}$$

$$\sum_{q\in Q_p}(x_{p,q,r}-x_{p,q,r-1}) = t^{in}_{p,r} - t^{out}_{p,r} \qquad \forall p \in P; r \in R : r > 1 \tag{12.4}$$

$$\sum_{p\in P} c_{p,r} = 1 \qquad \forall r \in R \tag{12.5}$$

$$c_{p,r} \le \sum_{q\in Q_p^{\text{score}}} x_{p,q,r} \qquad \forall r \in R; p \in P \tag{12.6}$$

$$\sum_{p\in P_q} x_{p,q,r} = C_q \qquad \forall r \in R; q \in Q \tag{12.7}$$

$$\sum_{q\in Q_p} x_{p,q,r} \le 1 \qquad \forall r \in R; p \in P \tag{12.8}$$

$$\overline{x}_{p,r} \le \sum_{q\in Q_p^{\text{score}}} x_{p,q,r} \qquad \forall p \in P; r \in R \tag{12.9}$$

$$\sum_{p\in P}\overline{x}_{p,r} \le X_r \qquad \forall r \in R \tag{12.10}$$

$$b_1 + \sum_{p\in P}\sum_{q\in Q_p} v_{p,1}\cdot x_{p,q,1} = B \tag{12.11}$$

$$b_r = b_{r-1} + \sum_{p\in P} v_{p,r}\cdot t^{\text{out}}_{p,r} - \sum_{p\in P} v_{p,r}\cdot t^{\text{in}}_{p,t} \qquad \forall r \in R : r > 1. \tag{12.12}$$

The objective function (12.1) maximizes the score obtained by a team across the season. Constraints (12.2) and (12.3) ensure that the season and weekly trade limits are not exceeded, respectively. Constraints (12.4) are trade consistency constraints, which enforce that one of the following four scenarios holds

- if a player is in the team in round r–1 but not in the team in round r then they have been traded out,
- if a player is not in the team in round r–1 but in the team in round r then they must have been traded in,
- if a player is in the team in both round r–1 and r then either they have been both traded out and in in round r, or they have neither been traded out nor in,
- if a player is not in the team in both rounds r–1 and r, then either they have been both traded in and then out in round r, or they have neither been traded in nor out.

Constraints (12.5) and (12.6) ensure that each week there is only one captain and the captain is in a scoring position, respectively. Constraints (12.7) and (12.8) ensure that the number of spots in each position are filled and that each player can only play in one position per round, respectively. Constraints (12.9) ensure that a player must be in a scoring position in order to have their score counted. Constraints (12.10) ensure that only the appropriate number of players' points count towards the total of the team each week. Constraint (12.11) ensures that the value of the initial side plus the remaining budget in the first round is equal to the starting budget. Constraints (12.12) are budget constraints which ensure that the current budget equals the budget from the previous round plus the value of the players traded out for the current round minus the value of the players traded in for the current round.

From the results of the OT-MIP, it is possible to understand the difference between the scores obtained by normal coaches in the competition and what was theoretically possible. Another way to understand this gap is to consider what the lowest starting budget would be from which it is theoretically possible to have won the competition.

12.3.2 Budget Team

In this section we introduce a MIP model that determines the lowest starting budget from which it is theoretically possible to win the competition (B-MIP). A single continuous variable β is introduced to represent the initial starting budget, i.e. instead of the parameter B. The score obtained by the winning team of the competition is denoted S^{win}.

The B-MIP can now be expressed as follows:

$$\min \quad \beta \tag{12.13}$$

$$\text{s.t.} \quad \sum_{p \in P} \sum_{r \in R} \psi_{p,r}(\overline{x}_{p,r} + c_{p,r}) \geq S^{\text{win}} + 1 \tag{12.14}$$

$$b_1 + \sum_{p \in P} v_{p,1} \cdot x_{p,q,r} = \beta \tag{12.15}$$

$$(2) - (10), (12).$$

The new objective (12.13) is to minimize the starting budget variable. Many constraints are the same as OT-MIP. Constraint (12.14) ensures the score obtained across the season is better than the winning score. Constraint (12.15) ensures that the budget left over after the initial team is selected plus the value of the initial squad is equal to the starting budget.

12.3.3 Ghost Team

In this section, we describe the MIP model that determines the highest score that a ghost team can obtain (G-MIP). Recall that a ghost team is one where the team does not change at all after it is set for the first round. This problem is considerably different to the previous two problems. In a ghost team, no trades are used and thus the decision variables representing the trades and the budget are not required. On the other hand, the vice captain and emergency assignments must be considered as it is not possible to change these decisions in a ghost team.

The emergencies are non-trivial to take into account. Recall that if multiple emergencies are selected on the same line, then only the lowest scoring emergencies will be considered. Hence for each substitute position $q \in Q^{\text{sub}}$ the possible *slots*, $S_q := \{0, \ldots, C_q - 1\}$, are enumerated. The intuition behind the slots is as follows. If an on-field player scores a zero then the first slot is opened up and filled by the lowest scoring emergency (if one exists) in the substitute position on the same line. If two on-field players score a zero then the first two slots are opened, the first slot is filled by the lowest scoring emergency and the second slot is filled by the second lowest scoring emergency assuming at least two emergencies are assigned on this line. This continues until the number of on-field zeros exceeds the number of emergencies on the same line.

The decision variables for G-MIP are defined in Table 12.4. The G-MIP can now be expressed as follows:

Table 12.4 Additional decision variables required by G-MIP

Variable	Type	Description
$x_{p,q}$	Binary	If player $p \in P$ is in position $q \in Q$
$\overline{x}_{p,r}$	Binary	If the score of player $p \in P$ is included in round $r \in R$
c_p^{cap}	Binary	If player $p \in P$ is captain
c_p^{vice}	Binary	If player $p \in P$ is vice captain
$\overline{c}_{p,r}^{\text{vice}}$	Binary	If the score of vice captain $p \in P$ is included in round $r \in R$
$y_{p,q,r,s}$	Binary	If player $p \in P$ is in slot $s \in S_q$ of position $q \in Q_p^{\text{sub}}$ in round $r \in R$
$e_{p,q}$	Binary	If player $p \in P$ is set as an emergency in position $q \in Q_p^{\text{sub}}$

$$\max \quad \sum_{p\in P}\sum_{r\in R}\psi_{p,r}(\overline{x}_{p,r} + c_p^{\text{cap}} + \overline{c}_{p,r}^{\text{vice}}) \tag{12.16}$$

$$\text{s.t.} \quad \sum_{p\in P} c_p^z = 1 \qquad \forall z \in \{\text{cap, vice}\} \tag{12.17}$$

$$c_p^z \le \sum_{q\in Q_p^{\text{score}}} x_{p,q} \qquad \forall p \in P; z \in \{\text{cap, vice}\} \tag{12.18}$$

$$\sum_{p\in P_q} x_{p,q} = C_q \qquad \forall q \in Q \tag{12.19}$$

$$\sum_{q\in Q_p} x_{p,q} \le 1, \qquad \forall p \in P \tag{12.20}$$

$$e_{p,q} \le x_{p,q} \qquad \forall p \in P; q \in Q_p^{\text{sub}} \tag{12.21}$$

$$\sum_{p\in P}\sum_{q\in Q_p^{\text{sub}}} e_{p,q} \le E \tag{12.22}$$

$$\sum_{p\in P_q} y_{p,q,r,s} \le 1 \qquad \forall q \in Q^{\text{sub}}; r \in R; s \in S_q \tag{12.23}$$

$$\sum_{s\in S_q} y_{p,q,r,s} \le e_{p,q} \qquad \forall p \in P; q \in Q_q^{\text{sub}}; r \in R \tag{12.24}$$

$$\sum_{p\in P_q}\sum_{s\in S_f(q)} y_{p,f(q),r,s} \le \sum_{\substack{p\in P_q \\ \psi_{p,r}=0}} x_{p,q} \qquad \forall q \in Q^{\text{score}}; r \in R \tag{12.25}$$

$$\sum_{p \in P_q} (y_{p,q,r,s} - y_{p,q,r,s-1}) \leq 0 \qquad \forall q \in Q^{\text{sub}}; r \in R; s \in S_q^+ \tag{12.26}$$

$$\sum_{\substack{p' \in P: \\ \psi_{p',r} < \psi_{p,r}}} e_{p',q} + \sum_{s \in S_q} (|S| - s) y_{p,q,r,s} \leq |S| \qquad \forall q \in Q^{\text{sub}}; r \in R; p \in P_q \tag{12.27}$$

$$\sum_{p \in P} \sum_{q \in Q_p} v_{p,1} \cdot x_{p,q} \leq B \tag{12.28}$$

$$\overline{x}_{p,r} \leq \sum_{q \in Q_p^{\text{score}}} x_{p,q} + \sum_{q \in Q_p^{\text{sub}}} \sum_{s \in S_q} y_{p,q,r,s} \qquad \forall p \in P; r \in R \tag{12.29}$$

$$\sum_{p \in P} \overline{x}_{p,r} \leq X_r \qquad \forall r \in R \tag{12.30}$$

$$\overline{c}_{p,r}^{\text{vice}} \leq c_p^{\text{vice}} \qquad \forall p \in P; r \in R \tag{12.31}$$

$$\sum_{p \in P} \overline{c}_{p,r}^{\text{vice}} \leq \sum_{\substack{p' \in P: \\ \psi_{p',r} = 0}} c_{p'}^{\text{cap}} \qquad \forall r \in R. \tag{12.32}$$

The objective (12.16) is to maximize the number of points scored across the season. Constraints (12.17) ensure that there is only one captain and one vice captain. Constraints (12.18) ensure that the captain and vice captain have to be in a scoring position. Constraints (12.19) ensure that the number of players in each position is equal to the capacity of those positions. Constraints (12.20) ensure that each player can only be in at most one position at a time. Constraints (12.21) ensure that a player can only be an emergency if they are playing in that position. Constraint (12.22) constrains the maximum number of emergencies, E, which can be assigned each round.

Constraints (12.23) ensure that each player can only be in a single slot. Constraints (12.24) ensure that a player can only be a candidate for a slot if they are an emergency. Constraints (12.25) ensure that each week the number of players who are playing in the corresponding scoring position $q \in Q^{\text{score}}$ who scored a zero represent the maximum number of scoring bench slots in the corresponding substitute position $f(q)$, where $f : Q^{\text{score}} \rightarrow Q^{\text{sub}}$ is a bijective mapping from the scoring positions Q^{score} to their corresponding substitute positions Q^{sub}. For example, $f(DEF) = DEF^{\text{sub}}$. Constraints (12.26) ensure that the lower indexed slots are used before the higher indexed slots, where S_q^+ represents the set of all slots for position $q \in Q^{\text{sub}}$ except index 0, i.e. $S_q \setminus \{0\}$. Constraints (12.27) enforce the ordering on the emergencies in the corresponding slots. An emergency can only be allocated to a slot if the index of the slot is greater or equal to the number of other emergencies in the same position who scored less than that player in the same round. For example, a player can go in slot zero only if no other emergency in the same position scored less than that player. A player can go in slot index one only if there is at most one other emergency in the same position who scored less than that player and so on.

Constraint (12.28) ensures that the value of the team is within the initial budget. Constraints (12.29) ensure that a player's score can count towards the total if they are either in a scoring position or a scoring emergency. Constraints (12.30) ensure that only the scores of an appropriate number of players count towards the team score. Constraints (12.31) ensure that that a player can only be a scoring vice captain if they are the vice captain. Constraints (12.32) ensure that a player can only be a scoring vice captain in a round if the captain does not score any points that round.

12.4 Results

Data was obtained through the SuperCoach website with permission from the Herald Sun for the 2018 competition. The models were implemented in Gurobi 7.5.0 and executed on the MonARCH HPC Cluster. The processors are Intel Xeon E5-2667 v3 3.2GHz, 20M Cache, 9.60GT/s QPI, Turbo, HT, 8C/16T (135W). 32GB of memory and 6 CPUs were requested.

A summary of the solve statistics is given in Table 12.5. The *Bin.*, *Cont.* and *Constr.* columns represent the number of binary variables, continuous variables and constraints, respectively. The objective of the OT-MIP and G-MIP are the total score obtained by the teams at the end of the season where the objective for the B-MIP is the minimum budget from which it is possible to win.

A summary of the scores per round for the teams associated with the optimal solutions is given in Table 12.6. The *Optimal*, *Ghost* and *Budget* columns represent the teams associated with optimal solutions to the OT-MIP, G-MIP and B-MIP, respectively. For each model both the score (*score*) obtained in each round, as well as a cumulative total (*cum.*) is given. Note that there might be multiple teams associated with each optimal solution; however, we only report the one team here. The *High Score* column represents the highest score obtained by any of the real players in the competition. The *Round Leader* column represents the score of the player who was currently leading after each round. Hence note that the score of 53,825 obtained by the leader after the final round is the score that won the competition.

Table 12.5 A summary of the solve statistics for the models on the 2018 SC data

Model	Bin.	Cont.	Constr.	Wall time (HH:MM:SS)	Nodes explored	Objective
OT-MIP	239,382	23	244,233	1:58:27	308,283	62,208
B-MIP	239,382	24	244,234	37:23:31	863,712	$5,137,200
G-MIP	90,895	0	111,892	2:52:05	11,180	52,935

Table 12.6 Summary of the team scores through the 2018 SC season

Round	Optimal		Ghost		Budget		High score	Round leader
	score	*cum.*	*score*	*cum.*	*score*	*cum.*		
1	2,558	2,558	2,413	2,413	1,747	1,747	2,629	2,629
2	2,747	5,305	2,324	4,737	2,069	3,816	2,605	4,996
3	2,539	7,844	2,322	7,059	1,965	5,781	2,479	7,335
4	2,627	10,471	2,249	9,308	2,078	7,859	2,548	9,582
5	2,626	13,097	2,196	11,504	2,049	9,908	2,505	11,834
6	2,633	15,730	2,369	13,873	2,077	11,985	2,471	14,026
7	2,813	18,543	2,280	16,153	2,251	14,236	2,603	16,343
8	2,738	21,281	2,324	18,477	2,160	16,396	2,499	18,612
9	2,834	24,115	2,348	20,825	2,366	18,762	2,581	20,967
10	2,754	26,869	2,368	23,193	2,315	21,077	2,677	23,473
11	2,632	29,501	2,221	25,414	2,254	23,331	2,508	25,751
12	2,245	31,746	1,467	26,881	1,977	25,308	2,285	27,476
13	2,302	34,048	1,731	28,612	1,936	27,244	2,305	29,416
14	2,283	36,331	1,992	30,604	2,124	29,368	2,340	31,254
15	2,839	39,170	2,527	33,131	2,689	32,057	2,639	33,501
16	2,927	42,097	2,519	35,650	2,799	34,856	2,711	36,044
17	2,772	44,869	2,354	38,004	2,580	37,436	2,751	38,444
18	2,783	47,652	2,365	40,369	2,715	40,151	2,812	41,016
19	2,862	50,514	2,480	42,849	2,686	42,837	2,705	43,474
20	3,196	53,710	2,548	45,397	3,057	45,894	2,907	46,043
21	2,805	56,515	2,299	47,696	2,652	48,546	2,679	48,462
22	2,972	59,487	2,754	50,450	2,718	51,264	2,842	51,102
23	2,721	**62,208**	2,485	**52,935**	2,602	**53,866**	2,892	**53,852**

The optimal team score of 62,208 would have beaten the winning team by 8,356 points. This is a remarkable amount given that the top 5,000 teams finished with over 51,406 points. Even more remarkably, the optimal team outperforms a team made by all of the weekly highscores, which would have obtained a score of 59,973 points. This is quite an achievement as the optimal team still needs to accommodate the trade limitation constraints and there are almost 200k teams in the competition.

The ghost team finishes with a score of 52,925 points which would equate to 77th position in the competition. Although it would not have won, given that not a single trade is used, the captain and vice captains are not changed, the positions are not changed and the emergencies are not changed this is still a very impressive result.

The scores of the *Budget* team significantly improve throughout the season. The initial budget of the team is only $5,137,200, which is 51.372% of the usual amount. As the value of the team increases throughout the season so too does the scoring

Table 12.7 Summary of the team value and remaining budget through the 2018 SC Season

Round	Optimal		Ghost		Budget	
	value ($)	*budget ($)*	*value ($)*	*budget ($)*	*value ($)*	*budget ($)*
1	9,998,500	1,500	9,996,000	4,000	**5,137,200**	–
2	9,998,500	1,500	9,996,000	4,000	5,137,200	–
3	9,998,500	1,500	9,996,000	4,000	5,137,200	–
4	11,086,600	61,700	10,662,700	4,000	6,392,900	6,200
5	11,956,100	67,700	11,072,900	4,000	7,415,200	19,500
6	12,362,000	166,300	11,337,100	4,000	8,109,400	59,800
7	12,831,700	86,100	11,632,100	4,000	8,861,000	18,300
8	13,365,800	1,200	11,856,200	4,000	9,333,000	77,800
9	13,752,300	300	12,029,300	4,000	9,946,800	38,000
10	13,926,700	224,100	12,219,100	4,000	10,516,200	31,100
11	14,381,500	28,300	12,366,100	4,000	10,937,500	18,000
12	14,502,300	40,500	12,442,000	4,000	11,158,900	99,400
13	14,643,100	14,200	12,524,300	4,000	11,270,300	216,100
14	14,554,400	161,200	12,476,500	4,000	11,641,600	46,900
15	14,695,800	116,500	12,587,700	4,000	11,889,300	93,900
16	14,815,700	116,500	12,768,200	4,000	12,352,500	64,700
17	15,035,400	116,500	12,998,000	4,000	12,928,100	1,900
18	15,151,800	220,500	13,143,300	4,000	13,336,300	1,900
19	15,205,800	220,500	13,173,000	4,000	13,592,900	1,900
20	15,167,700	278,100	13,217,400	4,000	13,711,600	57,000
21	15,363,400	263,900	13,280,000	4,000	13,963,900	57,000
22	15,525,000	263,900	13,190,300	4,000	14,092,800	57,000
23	15,777,000	263,900	13,220,900	4,000	14,230,000	57,000

output. In fact, towards the end of the season the scores obtained by the Budget team outscore the weekly high score on three occasions (rounds 15, 16 and 20). This allows the Budget team to eventually catch the competition leaders and finish with a final score that would have won the competition by 14 points.

A summary of the team value and remaining budget for the teams associated with the optimal solutions is given in Table 12.7. The *value* column represents the total price of all the players current in the team, whereas the *buget* column represents the remaining budget that can be used for trades.

An optimal solution to the B-MIP must have at least one round where there is zero remaining budget otherwise a team with a better objective function could be found by simply reducing the initial starting budget by the minimum remaining budget of any of the rounds. This is reflected in the first three rounds where the *Budget* team has zero remaining budget. It is interesting to note that despite starting with just over half of the initial budget, the team value of the *Budget* team at the end of the season of $14,230,000 is actually significantly more than the final team value of the winning team of $13,671,100.

The remaining budget of the *Ghost* team does not change at all during the season. Again this is to be expected as the remaining budget only changes when players are traded in and out of the team. As the ghost team does not use a single trade then the remaining budget is never changed.

The *Optimal* team finishes with a large remaining budget of $263,900. The combination of the final team value and remaining budget of over $16m is significantly larger than any player in the competition. Although the objective function of the OT-MIP is to score the most points possible, clearly there is a correlation between teams that score well and teams that are valued highly.

12.5 Conclusion and Future Work

This paper demonstrates a novel application of MIP that of analyzing fantasy sport competitions. Due to their well-defined nature and increasing popularity, fantasy sport competitions provide the perfect sandpit to demonstrate the sort of questions that MIP can answer to the general public. This paper used three different MIP models to highlight the difference between what real players achieved and what was theoretically possible for the 2018 season of SC.

The first MIP model (OT-MIP) determines the highest score that was theoretically possible (62,208), which is vastly more than the competition winner (52,852). To provide an alternative way to communicate how significant this difference is, the second MIP model (B-MIP) shows that it was theoretically possible to win the competition starting with only 51.37% of the typical starting budget. Both the OT-MIP and B-MIP benefit from being able to make many changes each round based on information that could not be known at the time. To avoid this, the third model (G-MIP) shows that a team in which not a single change is made after round one could have finished the competition in 77th position. As the competition has just under 200k teams (199,243) this is perhaps the most remarkable result of the three problems.

The main fantasy competition of the EPL, the Fantasy Premier League (FPL), has significantly more users than SC, 5.5 million teams (5,596,712). Hence determining the optimal teams for the FPL would be a logical extension of this work to reach an even larger audience. The FPL has a number of additional challenges that make modelling the problem non-trivial, such as taxing of profits made on players and special tokens that can be used once a season to, for example, change all the players in the team with no cost in a single round.

Another way of extending this work is to use it to try and win the competition. Instead of obtaining data and analyzing the competition retrospectively, the OT-MIP model can be used to convert forecast data into actionable decisions. This approach is clearly fundamentally limited by the inability to perfectly forecast results. However, as very clearly highlighted by the results in this paper, there is currently a significantly large gap between the results obtained by actual teams in the competition and the theoretically best score possible. A forecaster would only need to outperform real teams.

The types of problems modelled and solved in this paper would provide practical assignment problems for graduate-level students in an introduction to MIP modelling class. Each fantasy competition has their own unique set of rules and thus different models are required. Typically data is freely available online and the rules of the competition are both well-defined and readily understood. Furthermore, a number of constraints such as the linking constraints (12.4) in the OT-MIP and the slot ordering constraints (12.27) in the G-MIP can be quite challenging to determine.

Finally, it is worth mentioning that some of the decisions that are required by the teams found by the MIP models in this problem would have been extremely counter-intuitive at the time the decision would need to have been made. For fans of these sorts of competitions these results can be both very interesting as well as extremely frustrating.

References

1. Edwards SJ (2017a) The 2017 Optimal AFL Fantasy Team finally proves link between Genius and Madness. https://www.dreamteamtalk.com/2017/09/11/the-2017-optimal-afl-fantasy-team-finally-proves-link-between-genius-and-madness/
2. Edwards SJ (2017b) The Crazy, Perfect SuperCoach Team. https://www.heraldsun.com.au/sport/afl/supercoach-news/not-even-albert-einstein-could-have-picked-this-incredible-supercoach-team-to-win-50000/news-story/6eb413fbca2ca661a64f0170eeb3f8e3
3. Edwards SJ (2018a) The Optimal SuperCoach Team 2018. https://www.heraldsun.com.au/sport/afl/supercoach-news/the-optimal-supercoach-2018-team-would-have-scored-11000-more-points-than-the-overall-winner/news-story/7114f1a9d11643a9abfb774ae9ae174f
4. Edwards SJ (2018b) The SuperCoach ghost' team that would have finished in top 100. https://www.heraldsun.com.au/sport/afl/supercoach-news/the-supercoach-ghost-team-that-would-have-finished-in-top-100-without-a-single-trade/news-story/5bda3054014637b52894a88e1ce5c249
5. Hunter DS, Vielma JP, Zaman T (2016) Picking winners using integer programming, pp 1–37. http://arxiv.org/abs/1604.01455,
6. Newell S (2017) Optimizing daily fantasy sports contests through stochastic integer programming. PhD thesis, Kansas State University

Part IV
Defence Decision Support Analysis

Chapter 13
Strategic Risk Management in Practice

Hossein Seif Zadeh, Terence Weir, Alexei I. Filinkov, and Steven Lord

Abstract Contemporary risk management methodologies are typically used for identification and prioritisation of strategic risks. The International Risk Management standard, ISO 31000:2009, is the world-wide basis for best practice in strategic level risk processes. However, due to the qualitative and subjective nature of strategic risk, its analysis requires a more nuanced approach than that used in more tactical or operational settings and this paper discusses the need to understand the range and nature of strategic threats, and how to represent risk assessments. As such, a particular focus of this work is on how to incorporate best practices in strategic risk analysis, and operations research into the design and application of strategic risk management in the Defence context. A number of steps are recommended incorporating international risk management best practices within the context and uncertainties unique to strategic risk management for Defence (as opposed to tactical or engineering risk management).

Keywords Strategic risk management · Risk management practice · Strategic resilience

13.1 Introduction

The International Risk Management standard, ISO 31000:2009 [1], is the world-wide basis for best practice in strategic level risk processes. This paper discusses the need to understand the range and nature of strategic threats and how to represent strategic risk assessments. Because of the qualitative and subjective nature of strategic risk, when compared to typical engineering type approaches to risk management, best practice for strategic analysis requires a more nuanced approach than that used in more tactical or operational settings. This work focuses on how to incorporate best practice in strategic risk analysis and operations research into the design of strategic risk management in the Defence context.

H. Seif Zadeh (✉) · T. Weir · A. I. Filinkov · S. Lord
Strategy and Joint Force, JOAD, Defence Science and Technology Group, Canberra, Australia
e-mail: Hossein.SeifZadeh@dst.defence.gov.au

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_13

13.2 Best Practice in Strategic Risk Assessment

Joint Directive 30/2015 by the Chief of the Defence Force and the Secretary of the Department of Defence directs that risk management be integrated into all decision-making processes in Defence, at all levels. The International Risk Management standard, ISO 31000:2009 [1], is the world-wide basis for best practice in strategic level national security risk processes [2–6]. Figure 13.1, taken from [1] depicts the principles, framework and process of risk management. The process of risk management under ISO 31000:2009 may, in essence, be summarised in a few key steps (see right hand panel in Fig. 13.1):

- "Establishing the context" and "Risk identification" to identify threat (source of potential risk in specific context including intent where relevant);
- "Risk analysis" to make assessment:
  - identify vulnerabilities (to specific threats);
  - assess the vulnerabilities; and
  - determine the risk (anticipated likelihood and consequences of specific threads actually occurring);

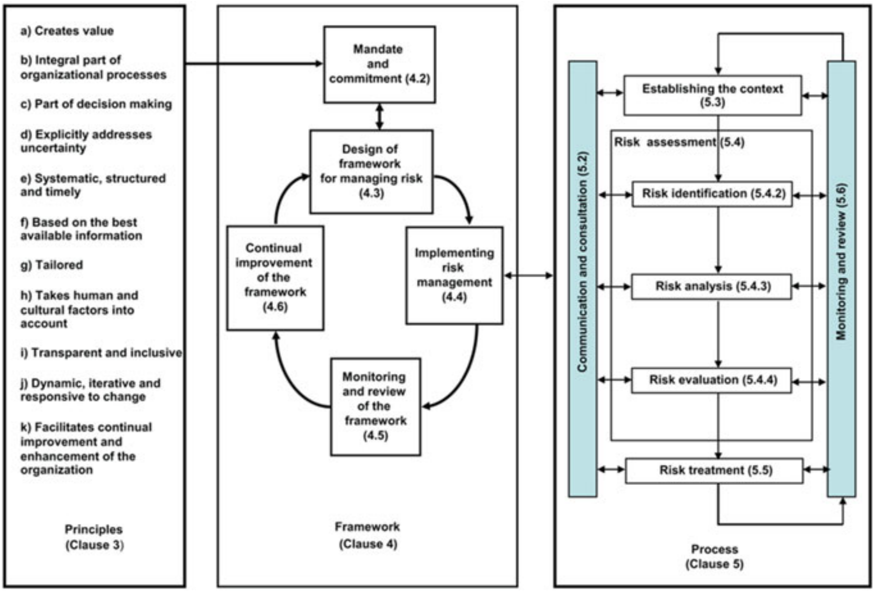

a) Creates value
b) Integral part of organizational processes
c) Part of decision making
d) Explicitly addresses uncertainty
e) Systematic, structured and timely
f) Based on the best available information
g) Tailored
h) Takes human and cultural factors into account
i) Transparent and inclusive
j) Dynamic, iterative and responsive to change
k) Facilitates continual improvement and enhancement of the organization
Principles (Clause 3)
Mandate and commitment (4.2)
Design of framework for managing risk (4.3)
Continual improvement of the framework (4.6)
Implementing risk management (4.4)
Monitoring and review of the framework (4.5)
Framework (Clause 4)
Communication and consultation (5.2)
Establishing the context (5.3)
Risk assessment (5.4)
Risk identification (5.4.2)
Risk analysis (5.4.3)
Risk evaluation (5.4.4)
Risk treatment (5.5)
Monitoring and review (5.6)
Process (Clause 5)

Fig. 13.1 ISO31000:2009 [superseded]: Relationships between the risk management principles, framework and process [1]. © Standards Australia Limited. Copied by the lead author with the permission of Standards Australia under Licence CL1506DoD

- "Risk evaluation" and "Risk treatment" to develop mitigation strategies that either seek to reduce those risks (either the likelihood or consequence, or both), or to tolerate the level of risk as it stands; and
- "Monitoring and review" (in parallel to all the above steps) to prioritise risk mitigation measures.

It is important to note, however, that the ISO standard is not prescriptive on the methodology of risk assessment, it merely provides guidance [7, 8]. It is argued that while a 'standard risk matrix' approach to assess risks, accompanied with short narrative description of threats and assignment of a likelihood and consequence is common in tactical (e.g. engineering) practice, it is not best-practice for strategic risk assessment, as it can lead to under- or over-estimation of strategic risk. Other errors observed in strategic risk assessment include lack of stakeholder involvement, possible false-positive risk scenarios, rigid multi-criteria impact evaluation and inconsistent likelihood estimation [9]. When applied to rare events, particularly those at the strategic or enterprise levels, the context within which any potential event occurs is of critical importance. Given the complexity of strategic risk in the Defence context, it is imperative to understand the range and nature of strategic threats. For instance, the threat posed by an earthquake extends beyond magnitude, and may include location, population density and dispositions (e.g. density might vary at different times of the day or night), seasonal issues, and underlying infrastructure and community preparedness or resilience (Fig. 13.2).

In cases where the nature and extent of a threat is driven by a conscious decision, the intent of those responsible for the threat should also be considered. Furthermore the vulnerabilities inherent in the system under investigation and their potential for exploitation need to be considered. This provides a basis for understanding how risk might manifest itself given potential threats and intents. Before undertaking any assessment of likelihood or impact, an analysis of threat-intent-vulnerability relationships must be explored. For instance, as well as specific targets, a belligerent

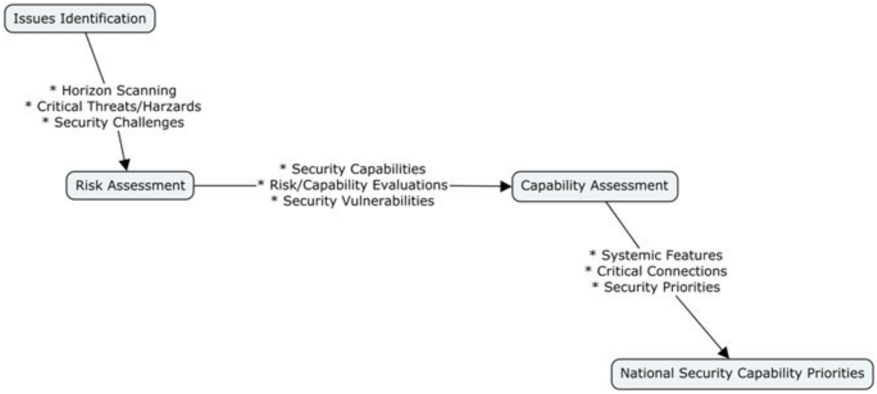


Fig. 13.2 National security risk methodology. Adapted from [22]

entity can choose the intensity and duration of a particular aggressive action; their choices likely aligning with the strategic ends they are seeking, mediated by the ‘cost’ of such actions, both to them and to us.

In other words, strategic risks do not have a single impact and a single likelihood, but a range of impacts and likelihoods, the spectrum of which need to be considered. To ignore these ranges of possibilities could result in risk misrepresentation, overconfidence, or bias estimates of likelihood and impact to either an overly positive or negative extreme [10, 11]. Additionally, quantification of likelihood and impact requires an appreciation of differences between strategic risk management and that found in operational or tactical management. For instance, probability of rare, but catastrophic events suggests a logarithmic (rather than a linear) scale should be used to better capture large differences. Furthermore, purely narrative, rather than more structured, representations of risks often make these ranges of likelihood and consequence ambiguous.

The combination of these factors, even when ambiguity is reduced, means that each ‘threat’ or ‘risk’ can have a range of different chances of occurring with different levels of impact. Imprecision in defining risks generally leads to reducing the assessment to merely the two extreme cases of the most likely and the most impactful. Best practice in strategic risk assessment suggests risk posed by a threat should be represented as a range of impact, rather than a singular point that is often represented in (engineering) risk matrices [12–20]. For quantification of likelihood and impact, a rigorous and consistent approach is critical. For instance, an actuarial approach to assigning probability of rare events suggests a logarithmic scale, rather than a linear scale is more appropriate [21] because it avoids range compression where discrete categorization lumps together very dissimilar risks. Similarly, assessing impact requires a clear articulation of the implications on the achievement of strategic objectives.

13.3 An Analytical Perspective to Strategic Risk Management Process

In line with the literature [22–34], for a mature strategic risk process the production of two distinct elements are proposed: a comprehensive foresight analysis; and an assessment of the implications for Defence. Figure 2 shows a simplified strategic risk process that illustrates features common to the cited literature. Best-practice methodology of identifying issues and assessing risks over a planning horizon consists of three broad aspects:

- establishing and building up over time the underlying evidence base, e.g. thematic analysis across previous reviews;
- employing expert elicitation processes to identify evolving and emerging issues; and
- an appropriate Strategic Risk Assessment approach for those issues.

It is important to ensure a strategic risk management process provides a bridge between:

- the immediate strategic risk outlooks;
- the medium term ones so that mitigations identified can be enacted; and
- the longer term ones so that the strategic direction underpinning long-lasting decisions can adapt to changing circumstances in a considered manner.

13.4 Building the Underlying Evidence Base

The foresight process consists of identification and monitoring of issues, trends, developments and changes in the domestic and international environments that could impact policies and programs. Regardless of how uncertainties are captured, there is general agreement in the relevant body of academic literature that the quality and representativeness of experts, selection and development of scenarios and the conduct of data collection are critical elements of the process [35].

It was recommended that the foresight process use a standard methodology for horizon scanning [36–39], where emerging issues, trends, developments and changes are broken down and mapped to key elements.

Another important step is to determine where and how to capture subject matter expertise. One approach is to employ a workable set of STEEPV variables (social, technological, economic, environmental, political and values, framed originally as PEST or STEP [25, 40–42]) may be used to create a 'panel of experts' with diverse, but relevant and complementary, expertise. Care should be taken to identify and recruit internal and external experts and stakeholders based on the focus areas and/or STEEPV domains.

13.5 Expert Elicitation

There are a plethora of techniques, such as Delphi, to bring together experts to identify, assess and prioritise strategic risks. Given the complexity of the environment, it is recognised that no single expert would have the capacity to make informed judgements across all areas [43]. Best practise to elicit such judgments from expert communities should be focused on specific questions relevant to the expertise of the individuals [44]. This is illustrated by examples such as the World Economic Forum Global Risks Report series which demonstrate the practice of issue selection, expert elicitation and strategic risk assessment [45].

A sound and proven elicitation approach is the combination of expert workshops and asynchronous communication [46]. The Investigate-Discuss-Estimate-Aggregate (IDEA) approach for structured expert judgement, developed by the University of Melbourne [47] is such an approach.

We also note that there is consensus in the literature that improved assessments occur when this is augmented with a scenario-planning process [44, 46, 48–50] to better understand relative impact(s). It has been demonstrated that multiple distinct scenarios, augmented by minor counterfactuals should be used to explore the strategic space [50]. It has been demonstrated in United States' Defence Strategic Planning, that without this, there is the danger that risk assessment in making strategic choices is undermined [11]. To that end the following observations are noted:

- Evidence demonstrates that expert panels are better at forecasting rare events than inexpert ones [46]. However, this comes with caveats [51, 52], as experts are prone to cognitive and motivational bias, require calibration, and 'expertise' or 'standing' are not necessarily an indicator of accuracy [53, 54]. We note that composition of panels between Defence and externals is known to affect outcomes [55].
- Evidence suggests that human experts are superior for qualitative predictions ('if' something will happen), while quantitative forecasting performs better on predictions of how much and when [56]. Methodology and time horizon have been identified in some studies as the most sensitive factors affecting prediction [57]. Other nations and large corporations are developing computer-based forecasting systems [26, 58, 59], but there is currently limited evidence whether a forecasting system improves the quality of foresight or narrows foresight through implicit standardisation. Strategic foresight is an area where it is difficult to determine value and effectiveness because of 1) the nature of the difference between foresight and forecasting; and 2) the timeframes and complexity of attributing strategic success or failure to one component of a strategic process such as horizon scanning [60–62].
- Care should be taken not to get carried away with technology-hype resulting in over-estimating the impact of emerging technologies. The expected speed at which different technologies mature must form part of the discussion; for example technological advances in heavy manufacturing may require years before they can materially impact capability, whereas technological advances in software may mature to a useable state in fraction of the time.

13.6 Strategic Risk Analysis

Having gained independent expert judgments about the tracking of issues, trends and other signals, and developed a sense of the 'likelihood' of these maturing in the relevant time, the next step is to synthesize this information within the strategic context. In effect, this requires an understanding of the implications for Defence. We recommended running 'implications' workshops that bring together 'suitable representation' of experts and Defence stakeholders, in order to assess the implications for Defence strategic planning and decisions. This would be commenced by bringing a number of experts together for a face-to-face meeting with key Defence personnel, as this would likely limit any corruption or biasing of the information-content of those earlier judgments, and likely strengthen the validity of the facilitated discussions [63]. This process has been developed and is successfully being used by Defence Science and Technology Group [64].

13.7 Recommendation

Defence Science and Technology Group has conducted work focussed on undertaking strategic risk assessment [65]. In these works, as in this paper, and following well recognised standards, the importance of understanding threat and vulnerability is emphasised as a basis for making such an assessment, as well as using a structured approach to capturing the assessments. Similarly, assessing impact requires a clear articulation of the implications on achieving strategic objectives.

With this in mind, we recommend organisations perform ‘issues identification’ and ‘risk assessment’ through a three-step process:

- Establish the strategic risk context in the near-, mid-, and long-term timeframes by identifying, analysing and challenging the assumptions of the organisation’s strategic documents (e.g. strategic plan, etc.). This informs the determination of those topics and themes that require further investigation, both to assure that the contexts identified in those documents remain current, and to help capture emerging threats/intents and vulnerabilities;
- Analyse these topics/themes by engaging an asynchronous expert elicitation approach. This requires groups of subject matter experts for each theme/topic to work through a Delphi-like approach. The output of this activity is the provision of evidence and assumptions to inform assessment of the strategic risks. It is noted that in addition to analysing these topics and themes, the subject matter experts may also identify ‘weak signals’ that may warrant further analysis and attention; and
- Synthesise these findings through a strategic risk assessment process which represents in a structured way the progression of threats to impacts. This may involve two iterations: (1) risk-based short listing of issues (i.e. risks emerging in specific contexts), and (2) detailed assessment to produce outcomes that can be used to explore the implications to policy and organisational design.

To undertake such a sophisticated approach requires time and resources. It is recommended that a level of rigour be maintained when limiting the scope in line with the available time and resources. Given strategic risk management is not a once-off product, but a repeating activity, future iterations could then update the now-extant elements while developing newer threads in order to improve rigour of the process.

Acknowledgements Authors thank Dr. Nigel McGinty and Mr. Richard Bartholomeusz from the Defence Science and Technology for useful discussions and comments.

References

1. Standards Australia and Standards New Zealand (2009) AS/NZS ISO31000:2009 Risk management—principles and guidelines. IEC, Geneva, Switzerland
2. Department of Homeland Security (2011) The strategic national risk assessment in support of PPD 8: a comprehensive risk-based approach toward a secure and resilient nation. Department of Homeland Security, U.S. Government. Washington, DC
3. Friend A (2012) The UK national risk assessment. Global Uncertainties Annual Meeting, Swindon
4. Mennen MG, van Tuyll MC (2014) Dealing with future risks in the Netherlands: the national security strategy and the national risk assessment. J Risk Res
5. Netherlands Government (2007) National security: strategy and work programme 2007-2008. Ministry of the Interior and Kingdom Relations, The Hague
6. UK Cabinet Office (2002) Risk: Improving government's capability to handle risk and uncertainty, Strategy Unit, UK Cabinet Office, Editor. London
7. International Standards Organisation (2009) ISO/IEC 31010:2009: risk management—risk assessment techniques. IEC, Geneva
8. Gibson C et al (2006) Handbook: security risk management. HB 167:2006. Standards Australia, Sydney
9. Vlek C (2013) How solid is the Dutch (and the British) national risk assessment? Overview Decision-Theoretic Eval Risk Anal 33(6):948–971
10. Fitzsimmons M (2006) The problem of uncertainty in strategic planning. Survival 48(4):131–146
11. Flyvbjerg B (2008) Curbing optimism bias and strategic misrepresentation in planning: reference class forecasting in practice. Eur Plan Stud 16(1):3–21
12. Bilusich D, Lord S, Nunes-Vaz R (2015) The implications of empirical data for risk. J Risk Res 18(4):521–538
13. Ball DJ, Watt J (2013) Further thoughts on the utility of risk matrices. Risk Anal 33(11):2068–2078
14. Cox LA Jr (2008) What's wrong with risk matrices? Risk Anal 28(2):497–512
15. Chapman C, Ward S (2011) Uncertainty, risk and opportunity. How to manage project opportunity and risk: why uncertainty management can be a much better approach than risk management. Wiley, Hoboken, NJ, pp 43–71
16. Cresswell S (2013) Qualitative risk: time for a rethink? http://www.intorisk.com/knowledge-centre/white-papers/qualitative-risk-assessmens. Accessed 16 Jan 2014
17. Hubbard DW (2009) Worse than useless: the most popular risk assessment method and why it doesn't work. In: The failure of risk management: why it's broken and how to fix it. Wiley, Hoboken, NJ
18. Pearce C (2013) Advice on the risk estimation matrix used by DAFF Biosecurity as part of the Import Risk Analysis process (Client Report CR0127 Australian Senate Rural and Regional Affairs and Transport Committee). Risk Management Ltd, Wellington, NZ
19. Pickering A, Cowley S (2010) Risk matrices: implied accuracy and false assumptions. J Health Saf Res Pract 2(1):9–16
20. Wall KD (2011) The trouble with risk matrices. In: DRMI working paper. Naval Postgraduate Schoool (DRMI): Monterey, CA
21. Levine ES (2012) Improving risk matrices: the advantages of logarithmically scaled axes. J Risk Res 15(2):209–222
22. Dupont A, Reckmeyer WJ (2012) Australia's national security priorities: addressing strategic risk in a globalised world. Aust J Int Affairs 66(1):34–51
23. Voros J (2003) A generic foresight process framework. Foresight 5(3):10–21
24. Garnett K et al (2016) Integrating horizon scanning and strategic risk prioritisation using a weight of evidence framework to inform policy decisions. Sci Total Environ 560–561:82–91
25. Durst C et al (2015) A holistic approach to strategic foresight: a foresight support system for the German Federal Armed Forces. Technol Forecast Soc Change 97(Supplement C): 91–104

26. Habegger B (2010) Strategic foresight in public policy: reviewing the experiences of the UK, Singapore, and the Netherlands. Futures 42(1):49–58
27. Nunes-Vaz R, Lord S, Bilusich D (2014) From strategic security risks to national capability priorities. Secur Chall 10(3):23–49
28. De Spiegeleire S (2011) Trends in capability planning for defence and security. RUSI J 156(5):20–28
29. Meyer-Nieberg S, Pickl S, Zsifkovits M (2015) Operations research for risk management in strategic foresight. Planet@Risk, Global Risk Forum GRF Davos 3(2):1–4
30. Department of Homeland Security, Risk Management Fundamentals (2011)
31. National Research Council (U.S.) (2010) Committee to Review the Department of Homeland Security's Approach to Risk Analysis, Review of the Department of Homeland Security's approach to risk analysis. National Academies Press, Washington, DC
32. Rajaratnam S (2011) Centre of excellence for National Security, Resilience and National Security in an Uncertain World. School of International Studies: Singapore
33. Talbot J, Jakeman M (2009) Security risk management body of knowledge. Wiley Series in Systems Engineering and Management. Wiley, Hoboken, NJ
34. Department of Homeland Security (2011) The Strategic National Risk Assessment (SNRA). http://www.dhs.gov/strategic-national-risk-assessment-snra. Accessed 15 Jan 2014
35. Pincombe B et al (2013) Ascertaining a hierarchy of dimensions from time-poor experts: linking tactical vignettes to strategic scenarios. Technol Forecast Soc Chang 80(4):584–598
36. Sutherland WJ, Woodroof HJ (2009) The need for environmental horizon scanning. Trends Ecol Evol 24(10):523–527
37. Ingle T, Staniforth A (2017) Horizon scanning for law enforcement agencies: identifying factors driving the future of organized crime. In: Larsen HL et al (eds) Using open data to detect organized crime threats: factors driving future crime. Springer International Publishing, Cham, pp 119–136
38. Delaney K, Osborne L (2013) Public sector horizon scanning-stocktake of the Australasian joint agencies scanning network. J Futures Stud 17(4):55–70
39. Urquhart GJ, Saunders P (2017) Wider horizons, wiser choices: horizon scanning for public health protection and improvement. J Pub Health 39(2):248–253
40. Davenport TH, Prusak L (1997) Information ecology: mastering the information and knowledge environment. Oxford University Press, p 272
41. Fleisher CS, Bensoussan BE (2003) Strategic and competitive analysis: methods and techniques for analyzing business competition. Prentice Hall
42. Miller KD (1992) A framework for integrated risk management in international business. J Int Bus Stud 23(2):311–331
43. French S (2012) Expert judgment, meta-analysis, and participatory risk analysis. Decis Anal 9(2):119–127
44. Derbyshire J (2017) The siren call of probability: dangers associated with using probability for consideration of the future. Futures. 88(Supplement C):43–54
45. WEF, The Global Risks Report 2017 (2017) World economic forum: Davos, Switzerland
46. Parente R, Anderson-Parente J (2011) A case study of long-term Delphi accuracy. Technol Forecast Soc Chang 78(9):1705–1711
47. Hanea AM et al (2017) Investigate Discuss Estimate Aggregate for structured expert judgement. Int J Forecast 33(1):267–279
48. Burt G, van der Heijden K (2003) First steps: towards purposeful activities in scenario thinking and future studies. Futures 35(10):1011–1026
49. Goodwin P, Wright G (2010) The limits of forecasting methods in anticipating rare events. Technol Forecast Soc Chang 77(3):355–368
50. Powell JH (2014) System/scenario duality—a supporting equivalence. J Oper Res Soc 65(9):1344–1360
51. Burgman MA et al (2011) Expert status and performance. PLOS ONE 6(7):e22998
52. McBride MF, Fidler F, Burgman MA (2012) Evaluating the accuracy and calibration of expert predictions under uncertainty: predicting the outcomes of ecological research. Divers Distrib 18(8):782–794

53. Tichy G (2004) The over-optimism among experts in assessment and foresight. Technol Forecast Soc Chang 71(4):341–363
54. Montibeller G, von Winterfeldt D (2015) Cognitive and motivational biases in decision and risk analysis. Risk Anal 35(7):1230–1251
55. Förster B, von der Gracht H (2014) Assessing Delphi panel composition for strategic foresight—a comparison of panels based on company-internal and external participants. Technol Forecast Soc Change 84(Supplement C): 215–229
56. Fye SR et al (2013) An examination of factors affecting accuracy in technology forecasts. Technol Forecast Soc Chang 80(6):1222–1231
57. Victor A, Banuls JLS (2011) Scope and design issues in foresight support systems. Int J Foresight Innov Policy 7(4)
58. Rohrbeck R, Thom N, Arnold H (2015) IT tools for foresight: the integrated insight and response system of Deutsche Telekom innovation laboratories. Technol Forecast Soc Chang 97:115–126
59. Kim S et al (2013) NEST: a quantitative model for detecting emerging trends using a global monitoring expert network and Bayesian network. Futures 52(Supplement C):59–73
60. Rohrbeck R, Schwarz JO (2013) The value contribution of strategic foresight: insights from an empirical study of large European companies. Technol Forecast Soc Chang 80(8):1593–1606
61. Balarezo J, Nielsen BB (2017) Scenario planning as organizational intervention: an integrative framework and future research directions. Rev Int Bus Strat 27(1):2–52
62. Gong M et al (2017) Testing the scenario hypothesis: an experimental comparison of scenarios and forecasts for decision support in a complex decision environment. Environ Modell Softw 91(Supplement C):135–155
63. Bolger F, Wright G (1994) Assessing the quality of expert judgment: Issues and analysis. Decis Support Syst 11(1):1–24
64. Emerging Disruptive Technology Assessment Symposium (EDTAS), https://www.dst.defence.gov.au/events/emerging-disruptive-technology-assessment-symposium-edtas. Accessed 8 Aug 2018
65. Institute for Regional Security, Kokoda Papers. Kingston ACT, 2005–2016

Chapter 14
A Security Focused Global Petroleum Trading Model

Gregory Calbert

Abstract Without liquid petroleum, either jet fuel, diesel, or other products such as lubricants and fuel oil, the Defence force ceases to function. Furthermore, the national support base, the bed-rock of the Defence force stops as well. Petroleum products will continue to form an energy source of choice for Defence because of superior energy density for decades to come. While Australia is seen as a regional power, its energy resilience is in a state of change and is generally seen as declining. Currently the Government sees market forces as providing petroleum supply security, given the nation stores 50–55 days of stocks which is below the mandatory 90 days required by the International Energy Agency. With this context, the Defence Science and Technology (DST) Group has developed a security-based global petroleum simulation, called SPECULA, in order to model the effects of regional and global changes in oil production, refining, shipping or distribution of petroleum products during conflict or significant environmental events. SPECULA is a simulation model, where petroleum is transported globally, based on regional variations in price. Price here is modelled as a "pseudo-price" which has a global component and a regional component. The global component is based on the difference between global supply and demand. The regional component is based on the regional inventory level. If the inventory level is low, the regional price rises. Trading, or the movement of petroleum from one region to another occurs because of inter-regional price differences, commonly known as arbitrage. The SPECULA model is spatial, as tankers move cargos along inter-regional seaborne routes. This paper briefly describes the security context of the petroleum supply chain in the Asian region. Then previous economic models of Australia's petroleum supply security are reviewed and critiqued in terms of their ability to model conflict scenarios. The SPECULA model is then described along with model parameters and outputs. Finally, the future challenges of this model are addressed in the discussion/conclusion.

Keywords Simulation · Petroleum supply chains · National security

G. Calbert (✉)
Strategic Analysis, Strategy and Joint Force, Defence Science and Technology Group, Canberra, Australia
e-mail: Greg.Calbert@dst.defence.gov.au

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_14

14.1 Introduction

Apart from bases where electricity and gas are the energy sources, nearly all Defence energy requirements are sourced from liquid fuels. Liquid diesel has an energy density of approximately 48 mega-joules per kilogram [1]. While battery research aims at increasing storage energy density, perhaps two or threefold, the latest lithium ion batteries have an energy density of no more than 2 mega-joules per kilogram. According to recent reviews, battery energy densities are unlikely to significantly increase in the short-medium term [2]. Other energy sources, such as hydrogen from ammonia-based fuel cells are promising, but have yet to show significant uptake. Therefore, at least at the time of writing this paper, liquid petroleum-based fuels will probably be the mainstay, or be used in hybrid form for Defence energy, especially for systems with high mobility, for many decades to come.

Australia has relied on the market as the source of energy security for liquid petroleum products. In the 2011 National Energy Security Assessment (NESA), the last undertaken by the Australian government, liquid fuels were assessed as having high, trending to moderate security over the medium term [3]. The NESA in 2011 stated that supply diversity and efficient markets underscored supply security.

Since 2011, the trend of refinery closures has continued with only four local refineries in operation nationally. New South Wales, the largest consumer of petroleum products in all categories, has moved to a product-import only business model, closing the Kwinana and Clyde refineries. Australia is the only International Energy Agency partner, not to comply with its 90-day net import rule, as only 50–55 days of net imports are stored [4].

Within a regional context, it is well recognised that the Asian region has become the key strategic vulnerability point for oil security [5]. There are a number of reasons, which compound each other, that lead to this conclusion. These reasons relate to: the high rates of regional growth; strong dependency on the Middle East for crude supply; lack of crude production; long supply chain distances; and a lack of a regional energy security alliances. So significant is the issue of energy security in the eyes of Chinese strategists, compounded by choke points in the Hormuz and Malacca straits, that it is termed the “Malacca dilemma” [6].

Energy supply chains may be interdicted or storage destroyed in conflict [7]. This occurs at all intensity levels of warfare, including the recent conflict with Daesh on the Iraqi-Syrian border [8]. Thus, from a Defence perspective it is important to understand how conflict could alter the petroleum supply chain and its implications for Defence sustainment.

The Security-Based Petroleum Trading Model, called SPECULA[1] in being developed at DST Group with this background in mind. Given petroleum trading and petroleum supply chains are global in extent, in SPECULA, thirteen regions, for example North America and South America trade fuel. Each region has a fuel price. The regional price is a function of overall global petroleum availability, when compared to global demand, and the region’s inventory level. If the regional inventory

[1] The acronym SPECULA is loosely based on the words “security”, “petroleum” and “speculation.”

level falls below some norm, the price increases in comparison to other regions. A state-machine simulates tankers moving petroleum cargos between regions. Tankers load cargo at regions with excess inventory, that are geographically proximate, and discharge at regions where the proximity and price are ideal. SPECULA is a spatial model, and tankers move between inter-regional seaborne routes.

Following this introduction, we describe and summarise some modelling approaches to the global oil/petroleum supply chain. These approaches motivated the development of the SPECULA model which is discussed in Sect. 3. Section 4 describes the outputs of the model, and the discussion-conclusion in Sect. 5 summarises future challenges and developments.

14.2 Global Petroleum Modelling

There are many publications on global petroleum models. Different models have distinct aims. The overarching theme of most is to form some predictive model of oil prices as a function of the parameters that affect the price of oil [9]. Such parameters are extensive, and include supply; demand, production; cartel behaviour, such as with the Organisation of Petroleum Exporting Countries (OPEC); the futures market; geopolitical events; and government policies. The success, or lack of success, of forecasting prices is aptly summarised by Michael Lynch in 2002, when he wrote “The history of oil forecasting has been a sorry one” [10]. Many papers focus on the role of OPEC in setting production targets in order to optimise the price for maximum profitability [11].

Modelling conducted by the Australian Government makes extensive use of the generalised computable equilibrium modelling (GCE) [12]. While a specialist discipline in macro-economics, GCE modelling seeks to model consumers’ optimal consumption levels of different commodities subject to price constraints. Producers in turn seek to maximise profit, which in turn is subject to the consumers demand, and other producers’ demands for the commodity produced and the cost of other factors such as labour and technology. At equilibrium one solves for commodity production levels, commodity prices and labour costs. In the context of the global oil market, changes in production or supply cause one to re-calculate ripple through effects, such as changes in price and production levels in other industry sectors. This approach has been used by the Australian government to assess energy security over oil supply shocks over a number of scenarios [13]. The Australian government assessed a closure of Singapore refineries for one month and the closure of the Strait of Hormuz for one month.

The use of GCE modelling essentially focuses on price rises in the market. Oil demand or supply shocks induce price rises, and demand-supply elasticity respond accordingly, surging production and decreasing demand, in short, the market balances itself out. However, there are a number of issues associated with the scenarios and models used to assess petroleum security.

First, the scenarios, developed through the Department of Energy and the Environment, fall short of conflict, where sea lines of communication may be blocked, refining destroyed and inventory storage damaged. Overall, the logistics of the supply chain, including the restrictions on sea-borne tanker numbers and the inventory levels are ignored in the GCE model. In conflict or during environmental events (such as a hurricane destroying port infrastructure), seaborne tankers can alter routes, with commensurate changes in the delays of oil or product flows. As important nations may choose to hoard stocks, regardless of the oil price.

Regarding the economic assumptions of the GCE models, the supply and demand elasticity of oil and petroleum are uncertain parameters. For example, Caldara et al. review estimates of the supply elasticity to be between 0 and 0.25 and the demand elasticity to be between -0.9 and -0.03 [14]. Such variation in these parameters can significantly alter the modelling results.

The reasons outlined above provide sufficient impetus for the development of alternative models of the global petroleum supply chain, which have sufficient flexibility to include spatial constraints and other factors which influence the supply for oil or product.

14.2.1 The Security-Based Model

With this background in mind the paper now discusses developing the SPECULA model. This model aims to take the intermediate ground between detailed econometric models of price and a logistics simulation model. It is global in scope, and subdivides the world into 13 regions, across which crude oil and petroleum products are traded. These regions are North America, South America, China, Russia, the Commonwealth of Independent States,[2] Japan, India, the Middle East and North Africa, Africa, Singapore, Australia and New Zealand, and the Other Asia Pacific nations.

These regions induce a sea-based distance matrix. Seaborne tankers, either for crude or petroleum product, transport fuel (either oil or product) between regions, according to a "proxy-price" model. The basic idea behind the proxy-price model is that it reflects both global attributes of price, excess global inventory or a shortfall, production and also regional price components. The regional price component is driven by local inventory levels and regional production. If the regional inventory level is below an average level, the price rises, and if the inventory level is above the average the price falls. Price differentials between regions generate arbitrage and affect the shipping patterns in the model.

Formally, let the 13 regions be denoted by $i \in \{1, 2, \ldots, R\}$ and the simulation time be denoted by $t \in \{1, \ldots, T\}$, then the inventory levels for crude oil and petroleum product at any time t are denoted by $I^c_{i,t}$, $I^r_{i,t}$. With these definitions one

[2] The Commonwealth of Independent States are the former Soviet territories (Armenia, Azerbaijan, Kazakhstan, Kyrgyzstan, Moldova, Turkmenistan, Tajikistan and Uzbekistan).

can define the proxy-price of either crude or product at region i at time t as

$$p_{i,t} = \frac{\sum_{i=1}^{R} I_{i,0}^{c,r}}{\sum_{i=1}^{R} I_{i,t}^{c,r}} + \max\left(K\left(1 - \left(\frac{I_{i,t}^{c,r}}{I_{i,0}^{c,r}}\right)\right), 0\right). \tag{14.1}$$

Readers will notice that equation one has the following properties. First as the total global inventory decreases compared to the start of the simulation, the proxy-price overall "global price" rises. Regionally, if $I_{i,t}^{c,r} \to 0$, the "arbitrage" component of the proxy-price also rises. Thus, in some sense, the proxy-price represents a regional "attractive force" that when high in a region compared to other regions, encourages the flow of full tankers to improve that region's inventory position. The approach of using an inter-regional proxy price is not dissimilar to the approach taken by Beyeler, Corbet and Hobbs, where flows in a pipeline network are not modelled by inter-regional prices, rather, by differences in "potentials" between regions [15]. In this model, it is assumed that the demand elasticity is zero, reflecting a risk-averse approach to assessing shortfalls in the supply chain. Furthermore, at this stage of research, the arbitrage component of the proxy price is assumed to be linear. Non-linear versions will be considered in subsequent forms of the model.

Having defined the price model, the transport of crude or petroleum product is now addressed. It is assumed that tankers are either full with cargo or empty. A tanker that is full seeks a region to supply fuel both geographically close and with a high price of fuel. In turn a tanker that is empty seeks to load petroleum cargo at a port that is geographically proximate, with a lower price of fuel. For simplicity, at this stage of the modelling process, it is assumed that seaborne tankers don't travel to intermediate nodes. Thus, full tankers will discharge at a port, and then decide which subsequent port is optimal for loading. There are no intermediate decisions on-route, a full tanker will decide where to discharge and an empty tanker where to load. Let $d_{ij}, i, j \in \{1, \dots, R\}$ denote the sea distance between two regions, ρ denote the daily cost of tanker transport and V be the cargo volume. A full tanker at region i seeks a set of regions $j \in J$ such that

$$j = \operatorname{argmax}_k\left(-\rho d_{ik} + p_{k,t}\mathrm{V}\right). \tag{14.2}$$

Similarly, an empty tanker seeks regions that minimise $\left(\rho d_{ik} + p_{k,t}\mathrm{V}\right)$. For crude transport, an additional rule restricts the possible destinations of full tankers. Full tankers do not travel to regions with excess crude production (for example Russia and the MENA regions). For this initial version of the model, decisions by the tankers are made myopically, that is the next port is chosen, not a sequence of ports. Subsequent versions of the model will consider such methods as depth search.

With these two rules for the evolution of global/regional proxy-prices and destinations for crude, product, full and empty tankers, one has the foundations for an executable simulation model, also termed a state machine. Each tanker has its own internal state $\{full, empty\}$, $\{loadingatport, dischargingatport, intransit\}$ and the state transition diagram is displayed in Fig. 14.1.

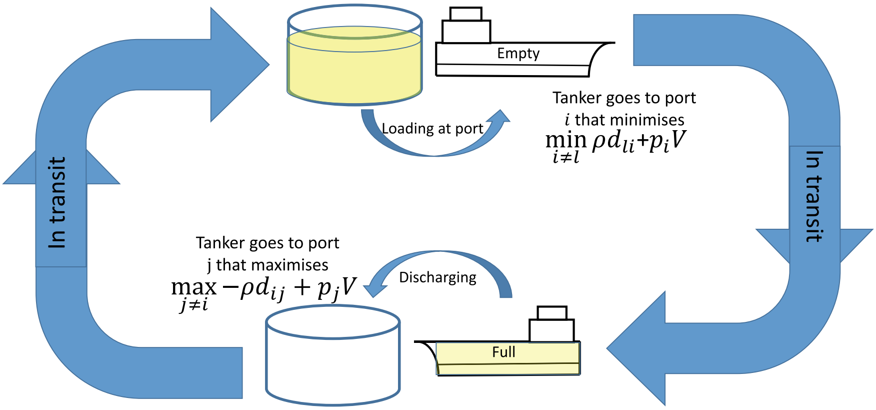


Fig. 14.1 Dynamics of full and empty tankers in the SPECULA simulation

The movement of crude or product via seaborne tankers, the production of crude, refining and finally consumption, generate the global inventory dynamics of the system. If $\mathbf{I}^c_t$,$\mathbf{I}^r_t$ denotes the vector of crude/refined (product) inventory globally, $\mathbf{c}$ is the regional crude production rate, $\mathbf{r}$ is the refining rate, **Cons** is the consumption rate, then the inventory dynamics for crude and refined product are simply

$$\mathbf{I}^c_{t+1} = \mathbf{I}^c_t + \mathbf{c} - \mathbf{r} + \mathrm{V}\mathbf{n}^c_{in,t} - \mathrm{V}\mathbf{n}^c_{out,t}, \tag{14.3}$$

$$\mathbf{I}^r_{t+1} = \mathbf{I}^r_t + \mathbf{r} + \mathrm{V}\mathbf{n}^r_{in,t} - \mathrm{V}\boldsymbol{n}^c_{out,t} - \mathbf{Cons}. \tag{14.4}$$

Here, $\mathbf{n}^{r,c}_{in,t}$, $\mathbf{n}^{r,c}_{out,t}$ denotes the number of full crude or product tankers offloading cargo at time t and the number of empty crude or product tankers loading cargo at time t.

With this core model, one is able to make the oil production, refining rates, consumption, or even inter-regional distances dynamic to explore the effects on stocks of various scenarios. The core parameters of the model as listed in Table 14.1.

14.3 Preliminary Results

In this section the outputs of the model are described. The computer code for SPECULA is currently written in the R language. Simulation time steps are taken as days. The inter-regional seaborne distance parameters are taken as days sailing, assuming tanker speed of 12 knots. Both the crude and product taken volume is assumed to be 100 mega-litres, approximately the size of a long-range type one (LR1), or long-range type two tanker (LR2), which are typical tanker sizes for inter-regional transport [16]. The parameters for crude production, refining and product

Table 14.1 Core parameters of the SPECULA model

Symbol	Meaning
i	Region in the simulation
t	Time measured in days
R	Number of regions
T	Maximum simulation time
d_{ij}	Seaborne distance between regions i and j
ρ	Daily price of seaborne tanker transport
V	Volume held by each petroleum tanker, crude or refined, in barrels
p_i	Price at region i
$p_{i,t}$	Price at region i at time t
$I^{c,r}_{i,t}$	Crude (c), refined (r) inventory level in region i at time t
K	Maximum regional price rise
c_i	Crude production in region i
r_i	Refined production in region i
$n^{c,r}_{in,i,t}$	Number of Crude (c), refined (r) tankers offloading cargo in region i at time t
$n^{c,r}_{out,i,}$	Number of Crude (c), refined (r) tankers loading cargo in region i at time t
$Cons_i$	Refined petroleum consumption in region

consumption are taken from the 2016 BP Annual Statistical Review [17]. Inventory capacities for various regions are estimated through the United States Energy Information Agencies (EIA) various country reports. The daily rate for tanker transport was taken at $20,000, and the price of oil was taken as $1.20 per litre. The simulation was done over 200 days.

One simulation, over the period of one year takes approximately 3 min to compute on an ASUS Quad core laptop. Therefore, a risk based model will require significant computational processing, when exploring sensitivities over the parameter space.

Figure 14.2 plots the inventory levels over the 13 regions assuming the simulation parameters shown.

Figure 14.2 shows a general rise in inventory levels globally in 2016. During this period, the oil price (West Texas intermediate) was at its lowest level since 2009, after the global financial crisis, because of a crude oil inventory glut, induced by shale production in the United States [18]. Therefore, at least in this first instance of running the SPECULA model with its input parameters, there is some congruence with the activity of the global market as a whole.

In the SPECULA model, both the crude and product trading dynamics can be visualised. Figure 14.3 shows a visualisation of a weighted trade matrix between regions for the flow of full crude tankers, over the simulation period.

Similarly, refined product trade in the SPECULA can be visualised, as is shown in Fig. 14.4.

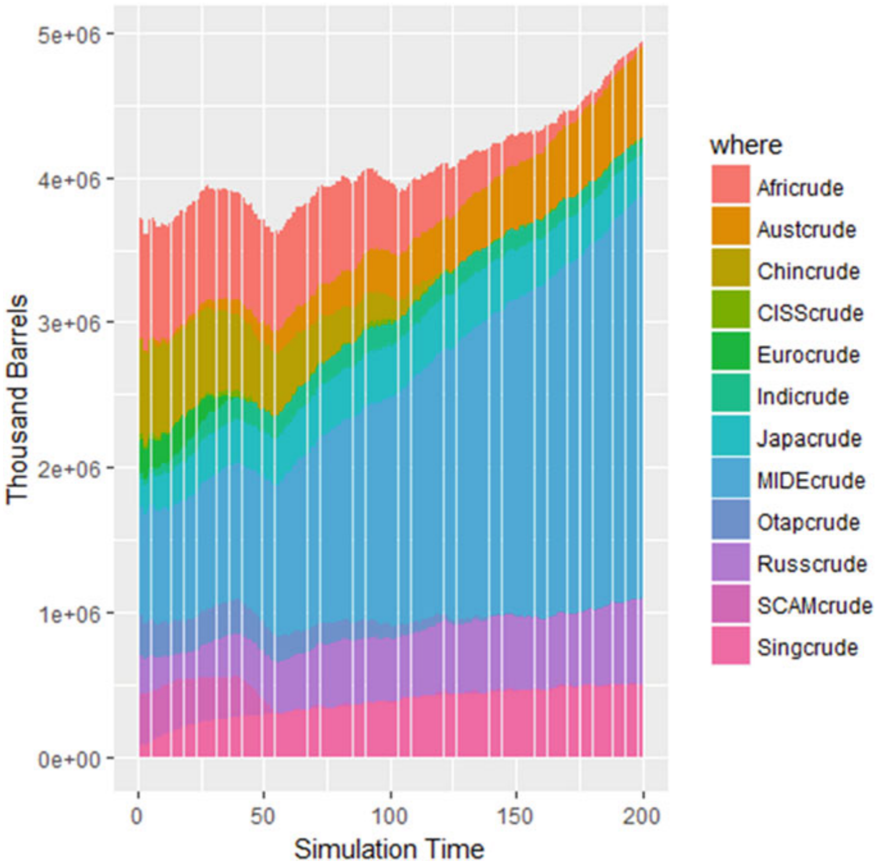


Fig. 14.2 Inventory levels across 13 regions in the SPECULA simulation model

Again, there is some model congruence with actual petroleum product trading in that inter-regional trading is far more uniform that crude oil trading, where the flows from Russia and the Middle East North Africa and South America dominate the trading routes, because they are the largest nett exporters of crude.

14.4 Discussion and Conclusion

The SPECULA model is in its nascent stages of development, and there are a number of challenges. First, the proxy-price model needs to be improved to better reflect the actual statistical patterns of global trading observed in data sourced from such bodies as the International Energy Agency. The model should include some look ahead in which tankers plan a sequence of ports to visit. In a highly dynamic situation, decisions regarding which port to load/unload could be made on a daily basis.

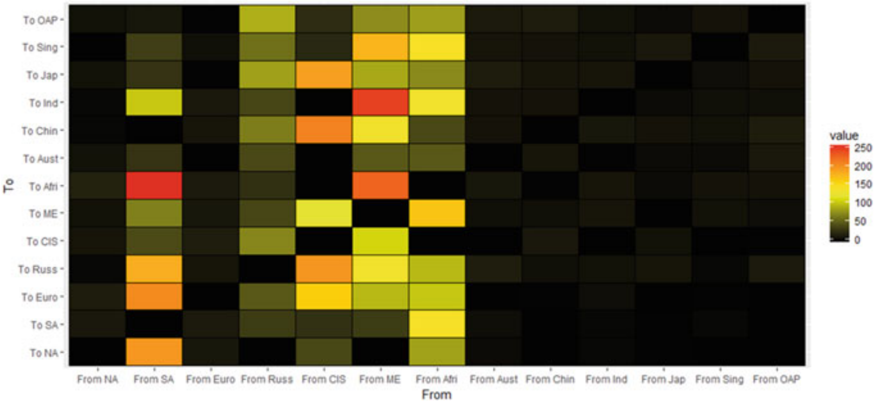


Fig. 14.3 Visualisation of the trading dynamics for crude cargoes between regions in the SPECULA model. NA- North America, SA- South America, Euro-Europe, Russ-Russia, CIS- Commonwealth of Independent States, ME-Middle East and North Africa, Afri-Africa, Aust-Australia and New Zealand, Chin-China, Ind-India, Jap-Japan, Sing-Singapore, OAP-Other Asia Pacific

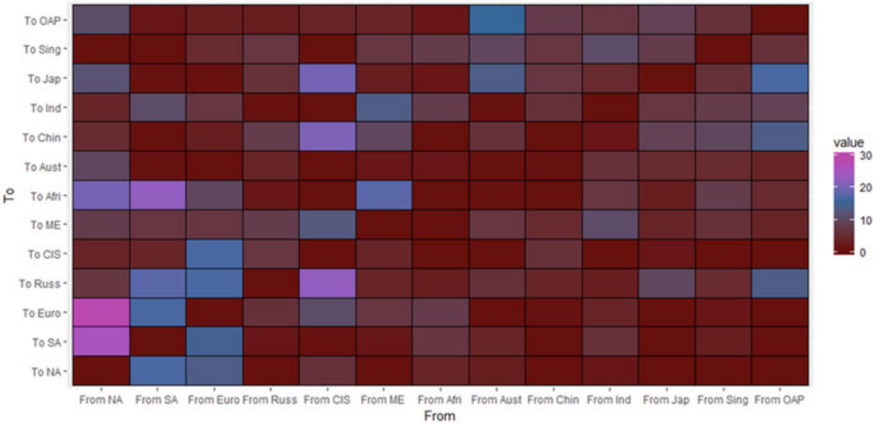


Fig. 14.4 Refined product trading in the SPECULA model, as is visualised by the number of refined tankers transiting across regions during the simulation model

In Asia, petroleum cargoes are typically not re-sold on-route, however in a crisis situation, decisions may be more dynamic. Even under normal circumstances, in Europe, petroleum cargoes are bought and sold a number of times on-route, because many ports are proximate [19]. This is the current subject of DST research, and a number of approaches may be taken drawing from non-linear optimisation methods to machine learning approaches, where such methods as reinforcement learning and value function approximation could be applied.

Another quite simple challenge is sourcing the data on fuel inventories and stocks. Generally, the nations that form part of the alliance that is the International Energy

Agency (including Australia) are quite transparent regarding their stocks and publish inventory levels. However, large petroleum consumers, such as China do not publish their data, and their inventory levels can only be estimated within some set of bounds [20].

GCE models estimate economic damage, through lost productivity, associated with oil shocks. In progressing the research of the SPECULA model, the aim is to communicate how changes in oil/product supply affect the national support base, and in turn military capability. More work is required on how this is done.

Given there is so much uncertainty with the parameters applied in the modelling, ideally Monte-Carlo sampling should be conducted to estimate risk profiles. SPECULA lends itself to parallel computing and this will be a subject of future research.

While the oil/petroleum supply chain is of strategic significance to Defence and the nation as a whole, it is not the only commodity. Australia also depends heavily on imports of fertiliser and fertiliser products for its agricultural productivity. Therefore, in the long-term SPECULA modelling may move to include other commodities, for an analysis of their respective strategic significance during conflict.

In summary, this paper has argued that there is a key modelling gap in the analysis of the global oil/petroleum supply chain. While GCE models are useful in this area, as with any other analytic endeavour, alternative modelling approaches will yield other insights. SPECULA attempts to bridge the gap between economic modelling, in its formulation of regional proxy prices, and direct supply chain modelling, given that seaborne tankers move oil/product. In its nascent stages, the next steps in its development include improving the proxy-pricing model and the testing of the model on various scenarios.

Acknowledgements Thanks to Dr. Richard Taylor, Dr. Richard Bartholomeusz, Professor Andreas Ernst, and a number of anonymous reviewers for their comments and feedback.

References

1. Wikipedia, Energy Density. https://en.wikipedia.org/wiki/Energy_density. Accessed 3rd Sept 2018
2. Choi JW, Aurbach D (2016) Promise and reality of post-lithium-ion batteries with high energy densities. Nat Rev Mater 1
3. Australian Government, Department of Energy Resources and Tourism, National Energy Security Assessment 2011, Fact Sheet. https://www.energy.gov.au/sites/g/files/net3411/f/national-energy-security-assessment-factsheet-2011_2.pdf. Accessed 3 Sept 2018
4. International Energy Agency, Closing stock levels in days of nett imports. https://www.iea.org/netimports/. Accessed 3 Sept 2018
5. Mitchell JV (2014) Asia's oil supply, risks and pragmatic remedies. Chatham House Research paper, 7th May 2014. https://www.chathamhouse.org/publication/asia-s-oil-supply-risks-and-pragmatic-remedies. Accessed 3 Sept 2018
6. Zhang ZX (2011) China's energy security, the Malacca Dilemma and responses. Energy Policy 39(12):7612–7615

7. Yergin D The Prize, The Epic Quest for oil, power and money. Free Press, 23rd Dec 2008
8. Johnson PB (2017) Oil, extortion still paying off for ISIS. RAND Corporation, October 27, 2017. https://www.rand.org/blog/2017/10/oil-extortion-still-paying-off-for-isis.html. Accessed 1st March 2018
9. Gyagri M, Amafio E, Marfo S (2018) Determinants of Global pricing of crude oil, a theoretical review. https://www.researchgate.net/publication/318571984_Determinants_of_Global_Pricing_of_Crude_Oil-A_Theoretical_Review. Accessed 4th Sept 2018
10. Michael C (2002) Lynch, Forecasting oil supply: theory and practice. Quart Rev Econ Financ 42:373–389
11. Al-Qahtani A (2019) A model for the global oil market: optimal production levels for Saudi Arabia. https://www.iaee.org/en/students/best_papers/Al-Qahtani.pdf. Accessed the 1st March 2019
12. Wing IS (2018) Computable General Equilibrium models and their use in economy wide policy analysis. Everything you ever wanted to know but were afraid to ask. http://www.rri.wvu.edu/CGECourse/Sue%20Wing.pdf. Accessed 5th Sept 2018
13. ACIL Tasman, Liquid fuels vulnerability assessment. https://www.energy.gov.au/sites/g/files/net3411/f/liquid-fuels-vulnerability-assessment.pdf. Accessed 5th Sept 2018
14. Caldara D, Cavallo M, Iacoviello M (2018) Oil price elasticities and oil price fluctuations, International Finance Discussion Papers, Board of Governors of the Federal Reserve System, https://www.file:///C:/Users/Greg%20Calbert/Desktop/Global%20fuel%20simulation%20one/global%20elasticities%20of%20oil.pdf. Accessed 5th Sept 2018
15. Beyeler WE, Corbet TF Jr, Hobbs JA (2018) A demand-driven, capacity-constrained adaptive algorithm for computing steady state and transient flows in a petroleum transportation network. Sandia report, SAND2012-9487. https://www.sandia.gov/CasosEngineering/_assets/documents/Beyeler%20et%20al%202012%20SAND2012-9487%20RAA%20%205315160%20errata.pdf. Accessed 6 Sept 2018
16. European Commission, Legal and Economic Analysis of Tramp Maritime Services, Annex 1: Glossary of key shipping terms, February 2007. http://ec.europa.eu/competition/sectors/transport/reports/tramp_annex1.pdf. Accessed 1 March 2019
17. BP, BP Statistical Review of World Energy (2016). https://www.bp.com/content/dam/bp/pdf/energy-economics/statistical-review-2016/bp-statistical-review-of-world-energy-2016-full-report.pdf. Accessed 6th Sept 2018
18. Trading Economics, Crude Oil, 1946-2018. https://tradingeconomics.com/commodity/crude-oil. Accessed 6th Sept 2018
19. Collins GB, Murray WS (2008) No oil for the lamps of China? Naval War College Rev 61(2)
20. Faucon B, McFarlane S (2018) China's secretive oil stocks are OPEC's latest concern in managing global supply. Wall Street J. https://www.marketwatch.com/story/chinas-secretive-oil-reserves-are-opecs-latest-concern-in-managing-global-supply-2018-06-05. Accessed 6th September 2018

Chapter 15
A Systems Approach to Analysing Organisational-Level Adaptability: Review of the Australian Army Lessons Network as a Case Study

Amina Omarova, Matthew Richmond, and Vernon Ireland

Abstract This paper describes a methodology for reviewing organisational-level adaptability from a systems perspective. Taking an action learning approach, we reviewed the Australian Army Lessons Network (ALN) as a case study in order to: (i) develop practical options for improvement of the ALN, (ii) reflect on the review methodology and identify options for improving its effectiveness in subsequent reviews, and (iii) demonstrate the utility of adaptive review and lay the foundations for its further application.

Keyword Organisational adaptability

15.1 Introduction

This paper describes an approach for analysing organisational-level adaptability from a systems perspective; i.e. by defining and mapping the relationships between the components of the system, to highlight shortfalls in the system structure and processes. We define highly adaptable organisations to be those that have the capacity to successfully modify themselves in response to a wide range of environmental changes (internal or external, current or unforeseen) by employing a structured process of variation generation, test and evaluation to underpin implementation. The objective of this work was to initiate the development of a methodology for reviewing organisational-level adaptability. We took an agile approach to its development, building on action learning principles; i.e. an approach based on iterative and

A. Omarova · V. Ireland
ECIC, the University of Adelaide, Adelaide, SA 5005, Australia
e-mail: amina@omarova.com.au

V. Ireland
e-mail: vernon.ireland@adelaide.edu.au

M. Richmond (✉)
Defence Science & Technology Group, Edinburgh, SA 5015, Australia
e-mail: matthew.richmond@dst.defence.gov.au

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_15

incremental development, where requirements and solutions evolve through collaboration between analysts and stakeholders. In this paper we describe a review of the Australian Army Lessons Network (ALN), which has been used as a case study to trial and improve the methodology iteratively.

15.2 Context

In 2008, Australian Army launched an ambitious program of putting its entire organisation on an adaptive footing, resulting in over 40 separate initiatives over the following 12 months [1]. Furthermore, the Directives at the time stressed that "*realising these goals for Adaptive Army requires specific targets for the improved outcomes to be set and regularly reviewed by Functional Commanders*", and that interdependencies between these areas were to be addressed explicitly to "*…allow us to adapt the ongoing prioritisation of effort and resources to the best effect, and avoid diminishing returns and negative cross-impacts*". A key to achieving the goals is the ability to improve the effectiveness of the adaptive processes themselves, by also subjecting them to adaptive review. As a step towards this, a limited scope review was initiated of one aspect of the Adaptive Army: effectiveness of the processes, structures and systems that comprise the *Army Lessons Network*.

A recent ANAO report reviewed "The Australian Defence Force's Mechanisms for Learning from Operational Activities" [2]. It found that Army has a sound process for learning immediate and short term lessons from operations and has effective structures to ensure these lessons are captured and communicated; however, it suggested that the learning loop model appears less suitable for medium and longer term lessons. It went on to point out that "*for these higher levels of analysis a stronger framework of measurement and evaluation would provide more robust information for decision-makers and planners*" [2]. This review only made broad recommendations which advocated consistent methods, a structured approach, and cross-service lessons sharing, supported by well-scoped and integrated knowledge management repositories.

15.3 Background

Our current research was initiated from the idea that by combining concepts from adaptability frameworks with those from decision-making cycles a methodology could be developed to underpin appraisals of organisational adaptability. At this point there appear to be no published methodologies to practically support this need. This section provides background on the published literature most relevant to a methodology.

15.3.1 Organisational Change and Adaptation

Increases in the level of complexity and the rate of change of operating environments, especially due to technological advances, make organisational adaptation a critical research area. Organisational adaptation is an internal property of organisations that facilitates successful organisational change. The literature related to organisational change is extensive and stems from several disciplines including political science, anthropology, biology, physics, psychology, as well as the business and management sciences [3–5]. Reviewing existing theories of organisational change, we have focused on *complexity theory* which has "presented a way of better describing and understanding dynamics and processes of change found in a range of physical and biological phenomena" [6].

Among a number of frameworks devoted to organisational adaptation [7], we have identified the Conceptual Framework for Adaptation (CFA) as the most relevant to our research objectives [8, 9]. The CFA is a framework to review organisational adaptability and builds on biological evolution and complexity theories and claims to provide "a generic model of adaptation, natural and hybrid types of adaptive mechanism, four classes of adaptation, five levels at which adaptation can be applied, and a discussion of the health of an adaptive mechanism, the levels of scale at which it can operate, and the factors that influence its effectiveness" [8]. However, it has not been operationalised so that it can be readily applied to practically support organisational adaptability appraisal. Note that the CFA model is not regarded as an empirically tested and validated model but rather a conceptual approach to support understanding and therefore resides more in the interpretive research paradigm than the positivist one. The conceptual and descriptive nature of the CFA to review organisational adaptability was seen as a flexible and comprehensive approach to understand and analyse characteristics of adaptability of a complex system such as the ALN.

15.3.2 Learning Cycles and Organisational Learning

We have considered two decision-making/learning cycles; Observe-Orient-Decide-Act (OODA) and Act-Sense-Decide-Adapt (ASDA) [10, 11]. Notwithstanding the differences between the OODA and ASDA loops, it is recognised that the two concepts have significant similarities and both can be used to assist in understanding and appraising continuous learning and adaption of organisations operating in rapidly changing environments [12]. There are also a number of theories that describe learning as a cycle. For example, Kolb's experiential learning, Deming's cycle of plan-do-check-act, Kofman's cycle of observe-assess-design-implement and Argyris's and Schon's a discovery-intervention-production-generalization cycle of learning [13]. The learning cycle approach is very close to the OODA and ASDA concepts; however, none of these cycles aim to build a system view of organisational learning processes and nor do they describe the necessary processes and principles that underpin organisational adaptation.

Complementing learning cycle concepts, Huber's framework of *organisational learning* provides a good foundation for understanding and describing processes of organisational adaptation. Huber argues that "*An entity learns if, through its processing of information, the range of its potential behaviors is changed*" [14]. He divided his framework into several parts: sensing and interpreting the environment, organisational decision-making, organisational learning and knowledge acquisition, knowledge managing, innovation and organisational culture [15]. Huber argues that in order to survive, an organisation should continuously change and adapt to new environmental conditions. Moreover, continuous adaptation requires constant innovation; this is not possible without obtaining new knowledge through learning. Therefore, we see Huber's approach to organisational learning as assisting to inform the relevant processes necessary for organisational learning and, more specifically, adaptation.

15.3.3 Systems Thinking

Conceptually, systems thinking "has shifted from structure (reflected in the use of modularization to deal with complexity), to organization or form (accentuated in the cybernetic approaches) to the network dynamics of adaptation and transformation (within the paradigm of complex systems science)" [16] and the modern idea of systems thinking is based on a belief that "systems cannot be understood by analysis—the properties of the parts can only be understood within the larger context of whole" [16]. Two approaches that were used to support this review are: (i) business architecture which "identifies its purpose, vital functions, active elements, and critical processes and defines the nature of the interaction among them" [17]; and (ii) business process modelling which provides a formal means to describes business processes in a flow-chart format, which assists visualisation and understanding those processes, including any issues or gaps with them.

15.4 Methodology

15.4.1 Approach

The review approach contains two main steps and is based on a combination of process modelling, action learning and conceptual analysis. In the *first* step: the system's business processes and information flows that facilitate organisational change to improve operational processes are described formally in process diagrams. The diagrams are used to support a gap analysis to identify both system's weaknesses and misalignment with the learning processes of the theoretical model. Noting that the quality of the processes is not assessed at this point, but only if they exist or not. In the *second* step, workshops are organised with key stakeholders to: (i) obtain their insights on the identified gaps in the processes, (ii) critically review processes described in the systems diagrams and assess their effectiveness, and (iii) engage managers/users/operators of the system in the analysis process to facilitate their ownership of any recommendations. Analysis of data from the workshops underpins the definition of key issues, which are corroborated with the analysis of historical examples of the organisation's responses to environmental changes and data analysis of process performance, leading to recommendations to improve the system's adaptability.

15.4.2 Theoretical Model of Organisational Adaptation

We have developed a theoretical model of organisational adaptability by combining aspects of learning cycles (OODA and ASDA), the Conceptual Framework for Adaptation (CFA) [9] and Huber's model of organisational learning [14, 15]. A list of desirable elements for a highly adaptable system is presented below which can be used to build a system view of learning processes in an organisation. At this stage the characteristics are used to identify the existence of the required processes only, but not to evaluate their quality.

Sense

- Sense information: data collection team; process/methods to support data collection; periodic sensing/monitoring process; data repository; process to observe triggers; and process to obtain feedback on prior adaptive changes.
- Conduct preliminary analysis: data analysis team; data repository for analysed data; periodic process/methods to analyse data from triggers to derive themes; periodic process/methods to analyse data from feedback; and process to report analysed information (including identification of possible variations) to appropriate decision authorities.

Analyse and Decide

- Analyse and Decide: decision-making authority; authority to direct tasking; the ability to influence indirect tasking; documentation of tasks; structured process of decision-making for variations approval (incl. documentation of reasons and guidance for further analysis); and data repository for decisions.
- Formulate Feedback: process to inform data collection teams about decisions and areas impacted; and define impact expected from decisions.

Adapt

- Implementation: process to capture parameters of implementation; process to capture conditions of implementation; data repository for implementation information; and process to support implementation information sharing.

- Formulate Feedback: process to inform data collection teams about expected impact of the implementation.

15.4.3 Gap Analysis Using Theoretical Models

Initially systems diagrams are developed, making use of the structure of a learning cycle mentioned in the previous section. Leveraging the ASDA learning cycle, we describe the business processes and information flows. There are two perspectives:

(i) *The designed view*: i.e. description of the functional workings of the system based on formal documentation; the diagrams describe the business processes, products and information flows between the relevant parts of the network; by explicitly documenting system diagrams we avoid any misinterpretations or misunderstandings when analysing and communicating the system processes.
(ii) *The as-is view*: i.e. how the system functions in reality. Stakeholders and operators are interviewed to obtain feedback on how the system functions in practice.

The *designed* and *as-is* views are initially compared to each other and implementation/institutionalisation weaknesses in the existing processes identified. The *designed* and *as-is* views are then compared with the *theoretical model* to provide insights into system deficiencies in terms of the key attributes of organisational adaptability.

15.4.4 Stakeholder Engagement

This step of the analysis focuses on obtaining input from stakeholders on adaptation processes, where issues highlighted during the gap analysis are used to initiate discussion. We employed the TOWS method as described by Coyle [18]. It provides a well-defined, structured approach to: (i) source insights from key stakeholders on relevant processes; (ii) develop options for overcoming/exploiting external threats and opportunities; and (iii) develop strategic plan, i.e. options for organisational change that can be traced back to the stakeholder's insights. TOWS is an extended version of SWOT (Strengths–Weaknesses–Opportunities–Threats) analysis, where the first part involves the running of workshops with key stakeholders to facilitate their input into "thinking about the threats and opportunities of the external world before considering the internal weaknesses and strength" and the second part involves the development of a "TOWS table … to compare, contrast and combine Ts, Os, Ws and Ss, in various ways, so that action plans which, say, use one of the strengths to exploit an opportunity [will] emerge" and guide strategic insight development [18, p. 89]. The first part only of TOWS (i.e. SWOT) analysis is applied in this step of the methodology; where the development of an action plan and strategic insights supports the final step of the methodology (Sect. 15.4.6).

15.4.5 Data Analysis Using the CFA

- **Historical analysis**. Historical analysis on organisational responses to environmental changes to provide evidence for the issues highlighted in the previous step and to clarify/characterise them more fully. The data set is categorised by classes and levels of adaptation, as well as by the scales and areas of the organisation, highlighting deficiencies (defined within the CFA) [9]:

  (i) *Scales of adaptation*. Organisational adaptation is a combination of different adaptive processes aimed at achieving a common level of success. Thus, organisational adaptation is a network of adaptive processes at various organisational scales. The organisational learning literature highlights three main levels of organisational learning: individual, group and organisational. Review of historical examples of learning and adaptation incorporates scales of adaptive processes as an integral part of the appraisal.

  (ii) *Levels of adaptation*. The CFA describes five levels of adaptation: action in the world, learning, learning-to learn, define success and co-adaptation.

  (iii) *Classes of adaptation* represents possible organisational responses to an external change. Taking a 'black box' perspective, this part of the CFA uses classes of adaptation to identify how an organisation responds to different types of unexpected changes. These classes are: agility, flexibility, responsiveness and robustness.

- **Quantitative data analysis**. The second type of analysis is quantitative data analysis of individual process performance to gather evidence for how well processes function. This analysis should be based on statistical analysis of operational performance and possible surveys of system users; for example, (i) to assess the process effectiveness, efficiency, and timeliness from a user perspective, and (ii) to assess whether relevant employees are aware of the process and use of it.

15.4.6 Synthesis of Analysis Leading to Recommendations

This step culminates with a key set of recommendations to improve the adaptive processes. The SWOT analysis should lead to a list of the key issues that require amelioration and also support a comparison of the internal weaknesses and strengths of the adaptive process with the external threats and opportunities of the environment leading to actions to improve the adaptive processes. At this point the outputs from the data analysis are used to provide evidence for issues and to support the development of actions. Actions are combined into strategic insights which are used to provide an indication of the main directions for change and to underpin the development recommendations.

Although the focus of this methodology is on system analysis, clearly other aspects of organisational learning are important and may constrain the operation of adaptive processes. This is especially true of cultural aspects. It is at this point of the methodology that any research on these aspects should be incorporated to ensure that any final recommendations are feasible and accommodate the culture and behaviours of the organisation. Issues and recommendations combined inputs from the SWOT workshops, data analysis and cultural studies.

The synthesis will lead to clarification of the issues which have been identified during the gap analysis and TOWS workshops. Results of the synthesis are: a list of explicitly expressed issues of the current system work supported by examples from historical and quantitative data analysis as well as from social/cultural perspective, and a list of recommendations aimed at improving the adaptive system which are feasible and practical. The final step of the synthesis is to clarify issues and recommendations with key stakeholders to obtain their feedback and acceptance.

15.5 Case Study

15.5.1 Process Models

This review has focused on the Army Learning Network (ALN) from a systems perspective. Formally, the ALN is divided into four temporal learning domains, referred to as learning loops: immediate learning loop (ILL), short learning loop (SLL), medium learning loop (MLL) and long learning loop (LLL). With the exception of the immediate learning loop, we investigated each of these areas with a focus on how these temporal organisational learning dimensions combine to address Army's overall learning. We have concentrated on identifying any gaps/overlaps, with the aim of developing proposals for practical improvements.

In order to analyse the effectiveness of the ALN, initially a systems model of each learning mechanism was developed based on available documentation; including the business processes, products and information flows between the relevant parts of the network. Stakeholders and operators were then interviewed to obtain feedback on how the system functions in practice. These data were mapped into diagrams making use of Act-Sense-Decide-Adapt learning loop and the CFA [9, 10]. Diagrams represent a system-of-systems view, generating two perspectives:

(i) The *designed view*: i.e. description of the functional workings of the ALN based on the formal documentation. The general view of processes supporting learning from operations contains a description of each of the three learning loops (SLL, MLL, LLL) and connections between them. Each loop is divided into the four steps of the ASDA loop. The development of the designed view was based on reference documents.

(ii) The *as-is view*: i.e. how the ALN functions in reality, including both formal and informal processes. The as-is view was created based on information from discussions during visits to four different parts of the Army.

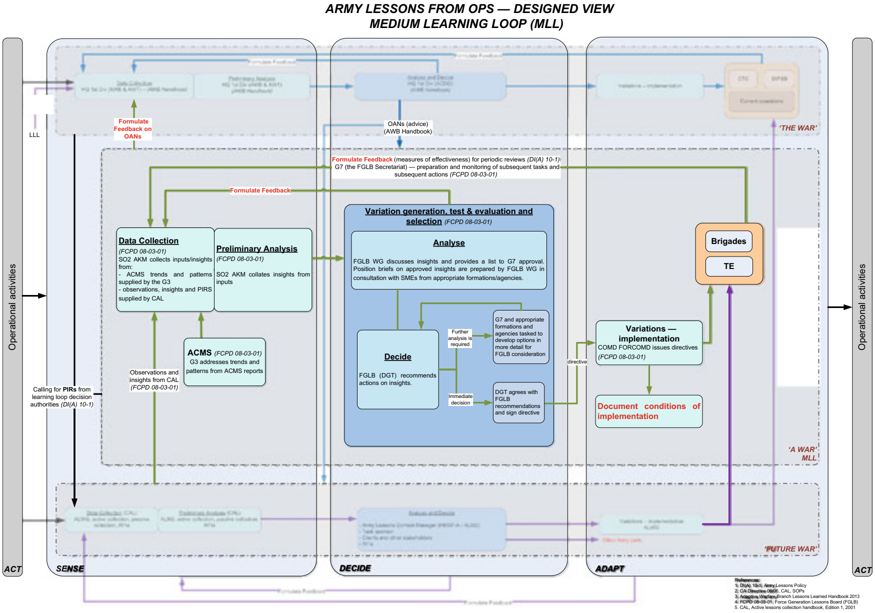

ARMY LESSONS FROM OPS — DESIGNED VIEW
MEDIUM LEARNING LOOP (MLL)
Operational activities
LLL
Formulate Feedback on OANs
OANs (advice) (AWB Handbook)
Formulate Feedback (measures of effectiveness) for periodic reviews (DI(A) 10-1)
G7 (the FGLB Secretariat) — preparation and monitoring of subsequent tasks and subsequent actions (FCPD 08-03-01)
Formulate Feedback
Data Collection
(FCPD 08-03-01)
SO2 AKM collects inputs/insights from:
- ACMS trends and patterns supplied by the G3
- observations, insights and PIRS supplied by CAL
Preliminary Analysis
(FCPD 08-03-01)
SO2 AKM collates insights from inputs
ACMS (FCPD 08-03-01)
G3 addresses trends and patterns from ACMS reports
Observations and insights from CAL (FCPD 08-03-01)
Calling for PIRs from learning loop decision authorities (DI(A) 10-1)
Variation generation, test & evaluation and selection (FCPD 08-03-01)
Analyse
FGLB WG discusses insights and provides a list to G7 approval. Position briefs on approved insights are prepared by FGLB WG in consultation with SMEs from appropriate formations/agencies.
Decide
FGLB (DGT) recommends actions on insights.
Further analysis is required
G7 and appropriate formations and agencies tasked to develop options in more detail for FGLB consideration
Immediate decision
DGT agrees with FGLB recommendations and sign directive
directive
Variations — implementation
COMD FORCOMD issues directives (FCPD 08-03-01)
Document conditions of implementation
Brigades
TE
'THE WAR'
'A WAR' MLL
'FUTURE WAR'
Operational activities
ACT
SENSE
DECIDE
ADAPT
ACT
References:
1. DI(A) 10-1, Army Lessons Policy
2. CA Directive 06/06, CAL, SOPs
3. Adaptive Warfare Branch Lessons Learned Handbook 2013
4. FCPD 08-03-01, Force Generation Lessons Board (FGLB)
5. CAL, Active lessons collection handbook, Edition 1, 2001

Fig. 15.1 Systems model of the medium learning loop of the Army Lessons Network—designed view

Figures 15.1 and 15.2 show the diagrams for the MLL designed and as-is views, respectively. They are included for illustrative purposes only, in order to give readers an idea of the style of the diagrams that were produced. The diagrams illustrate the links between the MLL and the other two learning loops (SLL and LLL) across the ASDA decision making/learning cycle.

15.5.2 Gap Analysis

The *designed view* and *as-is view* were compared to each other to identify any gaps or implementation weaknesses, and also any gaps when compared with the learning processes of the theoretical model were highlighted. These gaps formed the starting point for stakeholder workshops to (i) ascertain whether the stakeholders agreed, (ii) explore the issues in more detail and (iii) to stimulate discussion of related issues. The gaps are summarised below:

- Inconsistent processes for: (i) monitoring of decisions during implementation including defining expectations and setting goals to support feedback to decision-makers and decisions on any refinement required; (ii) developing options for decision-makers; (iii) prioritisation to focus data collection; and (iv) analysis methods.

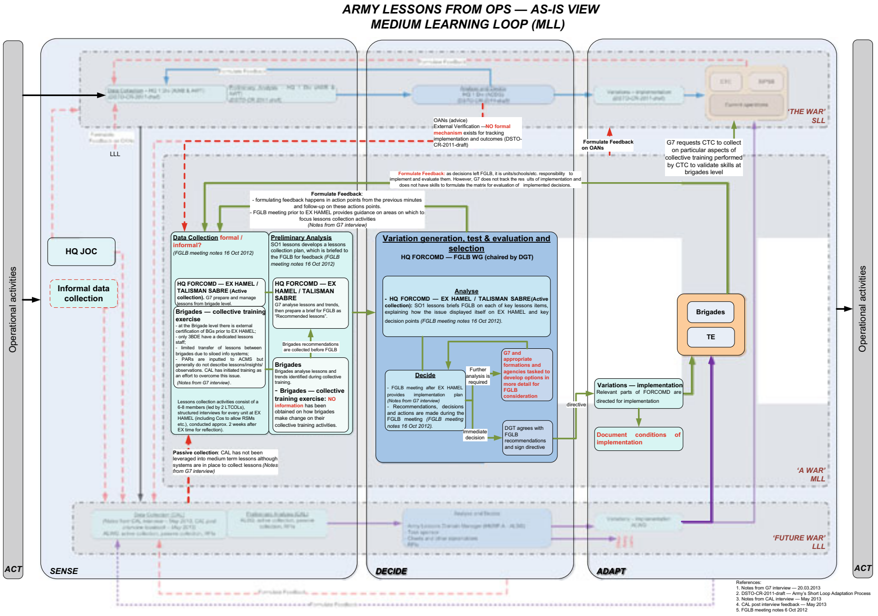

ARMY LESSONS FROM OPS — AS-IS VIEW
MEDIUM LEARNING LOOP (MLL)
Operational activities
ACT
SENSE
DECIDE
ADAPT
LLL
HQ JOC
Informal data collection
OANs (advice)
External Verification —NO formal mechanism exists for tracking implementation and outcomes (DSTO-CR-2011-draft)
Formulate Feedback on OANs
G7 requests CTC to collect on particular aspects of collective training performed by CTC to validate skills at brigades level
'THE WAR' SLL
Formulate Feedback: as decisions left FGLB, it is units/schools/etc. responsibility to implement and evaluate them. However, G7 does not track the res ults of implementation and does not have skills to formulate the matrix for evaluation of implemented decisions.
Formulate Feedback:
- formulating feedback happens in action points from the previous minutes and follow-up on these actions points.
- FGLB meeting prior to EX HAMEL provides guidance on areas on which to focus lessons collection activities
(Notes from G7 interview)
Data Collection formal / informal?
(FGLB meeting notes 16 Oct 2012)
HQ FORCOMD — EX HAMEL / TALISMAN SABRE (Active collection). G7 prepare and manage lessons from brigade level.
Brigades — collective training exercise
- at the Brigade level there is external certification of BGs prior to EX HAMEL;
- only 3BDE have a dedicated lessons staff;
- limited transfer of lessons between brigades due to siloed info systems;
- PARs are inputted to ACMS but generally do not describe lessons/insights/ observations. CAL has initiated training as an effort to overcome this issue.
(Notes from G7 interview).
Lessons collection activities consist of a 6-8 members (led by 2 LTCOLs), structured interviews for every unit at EX HAMEL (including Cos to allow RSMs etc.), conducted approx. 2 weeks after EX time for reflection).
Preliminary Analysis
SO1 lessons develops a lessons collection plan, which is briefed to the FGLB for feedback (FGLB meeting notes 16 Oct 2012)
HQ FORCOMD — EX HAMEL / TALISMAN SABRE
G7 analyse lessons and trends, then prepare a brief for FGLB as "Recommended lessons".
Brigades recommendations are collected before FGLB
Brigades
Brigades analyse lessons and trends identified during collective training.
- Brigades — collective training exercise: NO information has been obtained on how brigades make change on their collective training activities.
Passive collection: CAL has not been leveraged into medium term lessons although systems are in place to collect lessons (Notes from G7 interview)
Variation generation, test & evaluation and selection
HQ FORCOMD — FGLB WG (chaired by DGT)
Analyse
- HQ FORCOMD — EX HAMEL / TALISMAN SABRE(Active collection): SO1 lessons briefs FGLB on each of key lessons items, explaining how the issue displayed itself on EX HAMEL and key decision points (FGLB meeting notes 16 Oct 2012).
Decide
- FGLB meeting after EX HAMEL provides implementation plan (Notes from G7 interview)
- Recommendations, decisions and actions are made during the FGLB meeting (FGLB meeting notes 16 Oct 2012).
Further analysis is required
G7 and appropriate formations and agencies tasked to develop options in more detail for FGLB consideration
Immediate decision
DGT agrees with FGLB recommendations and sign directive
directive
Variations — implementation
Relevant parts of FORCOMD are directed for implementation
Document conditions of implementation
Brigades
TE
'A WAR' MLL
'FUTURE WAR' LLL
References:
1. Notes from G7 interview — 20.03.2013
2. DSTO-CR-2011-draft — Army's Short Loop Adaptation Process
3. Notes from CAL interview — May 2013
4. CAL post interview feedback — May 2013
5. FGLB meeting notes 6 Oct 2012

Fig. 15.2 Systems model of the medium learning loop of the Army Lessons Network—as-is view

- Limited coordination between lesson loops including processes that support information flows, across them.
- Services provided by lessons agencies are not always well integrated into the broader Army; e.g. quick look report production, ALIAS database as a knowledge repository to support Army strategic planning etc.
- Lessons from individual training and sub-unit collective training did not appear to be formally incorporated into business processes.

15.5.3 Stakeholder Engagement

The next step of the analysis focused on obtaining Army staff inputs on learning processes, where gaps highlighted in the first step were used to initiate discussion. Input was also sought in regards to the interaction of the ALN with capability development processes.

15.5.4 Data Analysis Using the CFA

- **Analysis of historical examples—ALIAS lessons database**. Data regarding the number of lessons and insights which have been identified over the lifespan of the

Army lessons database were provided. These data were classified, by two analysts independently, against 5 criteria; (i) data source that initiated the insight, (ii) the lessons loop where the insight would most likely lead to an action being implemented, (iii) FIC (Fundamental Inputs to Capability), (iv) classes of adaptation and (v) levels of adaptation. This analysis was used to corroborate issues and to support development of actions. Care must be taken not to over interpret these results as, first, they are not assessing the impact of the implementation of the lessons only a count (i.e. impact was not assessed) and, second, it is unclear what the distribution of issues should be for a highly adaptive organisation given the environment. Nonetheless, this data analysis does give an indication of the areas of focus for the lessons network. General observations from the results indicate that: (i) there were very few strategic lessons (for example, the Command and Management and Organisation FICs each make up less than 10% of the insights), (ii) few lessons are sourced from outside of operations or major exercises (i.e. few lessons from individual training or from sources external to Army), (iii) few insights relate to resilience (i.e. the ability to maintain core functions), and (iv) levels of adaptation primarily relate to levels 1 and 2 (i.e. actions in the world and the mechanisms to improve them).

- **Quantitative data analysis**. No quantitative data analysis of the core ALN business processes identified was conducted as unfortunately it was not feasible within the scope of the review. However, if the time and resources were available, it would have been useful to quantify how many personnel across Army are aware of the services that are available via the ALN and how many personnel make use of them. This type of analysis would provide insight into the effectiveness of individual processes and highlight areas for improvement.

15.5.5 Synthesis of Analysis Leading to Recommendations

- **Cultural perspective**. The Army Learning Organisation (ALO) team from the Land Division of DST Group had been studying Army's organisational learning properties for a number of years. To garner insights from the ALO studies on the issues related to the ALN, a workshop was run with the ALO team members. During the workshop we provided the ALO team a background brief on the methods and process employed in our review and then discussed each of the main issues and recommended actions in turn. The ALO team highlighted any constraints and other relevant information from ALO studies from a cultural perspective which added an additional dimension to review findings as well as often reinforcing them. We view ALO insights as a way of triangulating the ALN review results.
- **General synthesis**. The development of strategic actions and drivers was completed using the data from the stakeholder workshops and data analysis to provide evidence for any issues highlighted.

15.5.6 Findings

- **Issues**. There were 13 main issues identified during the review. The major issues highlighted during this review of the ALN can be characterised into four main categories: (i) ALN structures and processes, (ii) cultural issues, (iii) technical issues and (iv) Joint & Strategic issues. The main issues were listed for each category including evidence gathered that led to them being highlighted.
- **Recommendations**. The review resulted in 12 practical recommendations for improvement, which were presented at an internal Army advisory board in October 2013. Six recommendations were approved for implementation.

15.6 Conclusion

The implementation of the adaptive review process systematically studied the units and individuals within Army who collect, analyse or disseminate lessons and how the ALN contributes to Army becoming more adaptive at an organisational level. The issues identified detailed some limitations within the ALN. However, although this review has focussed on possible improvements to the ALN, there are many strengths of the ALN that became apparent during the review; including, the resilience and motivation of the personnel and units that make up the ALN, the well-developed processes internal to each of the learning domains, the strong international reputation of the ALN and a commitment from Army to support a mature organisational lessons capability.

15.7 Improvements to Methodology

- **Improvements to methodology**. There are a number of areas to improve the methodology:

  1. Description of the adaptation cycle and its properties should contain qualitative characteristics to make it possible to appraise processes.
  2. Application of a standardised approach to describing organisational processes, e.g. application of the Business Process Maturity Model (BPMN), which provides a formal means to describe business processes in a flow-chart format, which assists business managers to visualise and understand those processes, including any issues or gaps with them [19].
  3. A general view of a Capability Maturity Model (CMM) should be developed. CMM broadly refers to a process improvement approach that is based on a process model [20].

References

1. Breen B (2014) Preparing the Australian army for joint employment: a short history of the adaptive army initiative 2007–2010, Commonwealth of Australia 2014
2. The Australian Defence Force's mechanisms for learning from operational activities 2011, Australian National Audit Office
3. Kezar AJ (2000) Understanding and faciliating change in higher education in the 21st Century, Jossey-Bass
4. Rimmer G, Smith L (2006) Managing organisational change, 2nd edn. Markono Print Media, Singapore
5. Demers C (2008) Organizational change: a synthesis. In: Bruckner A (ed) Sage Publications, Inc, The United States of America
6. Ramalingam HJB, Reba JYWT (2008) Exploring the science of complexity: ideas and implications for development and humanitarian efforts, Overseas Development Institute
7. Omarova A, Ireland V, Gorod A (2012) An alternative approach to identifying and appraising adaptive loops in complex organizations. In: Complex adaptive systems conference, Washington D.C., USA
8. Grisogono A-M (2005) Co-adaptation. In: SPIE international symposium 6039-1: complex systems, Brisbane
9. Grisogono A-M (2010) Conceptual framework for adaptation. TTCP Technical Report
10. LWD 1 The Fundamentals of Land Warfare Commonwealth of Australia 2008
11. Boyd JR (1976) Destruction and creation. U.S. Army Command and General Staff College
12. Kelly J, Brenman M (2009) OODA versus ASDA: metaphors at war. Aust Army J Prof Arms 6(3):39–52
13. Botnis N, Crossan M, Holland J (2002) Managing an orgnaizational learning system by aligning stocks and Flows. J Manage Stud 39(4):437–469
14. Huber G (1991) Organizational learning: the contributing processes and the literature. Organ Sci 2(1):88–115
15. Huber G (2004) The Necessary nature of future firms: attributes of survivors in a changing world. SAGE Publications, Inc
16. Merali Y, Peter A (2011) Complexity and systems thinking. In: The Sage handbook of complexity and management, pp 32–53
17. Gharajedaghi J (2011) Systems thinking: managing chaos and complexity. Elsevier
18. Strategy Practical (2004) Structured tools and techniques. Prentice Hall, Geoff Coyle
19. Business Process Maturity Model (BPMM) Version 1.0, Object Management Group, June 2008
20. Humphrey WS Characterizing the software process: a maturity framework, Software Engineering Institute, CMU/SEI-87-TR-11, DTIC Number ADA182895, June 1987

Chapter 16
Stochastic Multi-criteria Decision Analysis of Combat Simulation Data for Selecting the Best Land Combat Vehicle Option

Thang Cao and Dion Grieger

Abstract Land Combat Vehicles (LCVs) are a critical fighting capability of the Australian Army. The operational effectiveness of a LCV is usually modelled via combat simulation in which the multi-criteria metrics are measured from the simulation output. Consequently, it is important to develop a multi-criteria decision-making procedure to support upcoming acquisition decisions for future vehicle options. Criteria measurements in combat simulation and decision-makers' preference often involve uncertainties; however, option ranking and selection procedures from simulations are normally limited to a single response metric or deterministic preference for the multiple metrics in the current literature. In this paper, we address these uncertainties by using a probability distribution function and Monte Carlo simulation in the stochastic multi-criteria acceptability analysis (SMAA) model for aiding this decision-making problem. Additionally, all uncertain preference information from DMs are represented as feasible weight space (FWS) and are used in combination with other weighting techniques such as analytical hierarchy process (AHP). The aim of this paper is to describe the application of SMAA, FWS and AHP to the results generated in a close-loop combat simulation, such that the options with uncertain data can be evaluated and analyzed, and the best option can be selected for a specific task or scenario. To the best of our knowledge, this combined approach has been applied for the first time to deal with the defence decision analysis problems with uncertainty and interdependency.

Keywords Defence · Stochastic · Multi-criteria decision analysis · Simulation

T. Cao (✉) · D. Grieger
Joint and Operations Analysis Division, Defence Science and Technology Group, Edinburgh, SA 5111, Australia
e-mail: thang.cao@dst.defence.gov.au

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_16

16.1 Introduction

Operational characteristics, future design and configuration options of Land Combat Vehicles (LCVs) have been investigated and explored by Defence Science and Technology Group (DST). The design of suitable LCVs for Australian Army's intended purposes necessarily should be influenced by close combat capability factors which can be established by subject matter experts (SMEs) and combat simulations. To this end, the operational effectiveness of key vehicle characteristics, such as lethality and survivability of combined arms teams, is studied in the context of close combat for varying levels of environmental complexity and enemy threat. Various metrics are used to provide an assessment of each option's lethality, survivability, signature and knowledge acquisition capabilities. The metrics have been selected to provide insights into specific characteristics of the vehicle equipped combat teams. In this paper, a multi-criteria analysis of operational effectiveness is established which combines all of the individual metrics together with associated SME weightings for these metrics.

In multi-criteria decision-making (MCDM), the most preferred alternative(s) from a set of m alternatives is often chosen by decision-makers (DMs) based on the evaluated n weighted criteria. The weights on these criteria are elicited from the subjective preferences of the DMs. Often the criteria are themselves composed of p_i weighted sub-criteria. The scores of the alternatives on the criteria or sub-criteria may be determined by interviews, surveys or activities with SMEs; direct measurements; or through the outputs of mathematical models or simulations. Whatever the sources of the scores (and there may be several in a MCDM problem) most real-world applications must cope with both uncertain and incomplete data. The weights are also frequently uncertain and occasionally incomplete. Particularly in our study, the criteria and preference weights are evaluated by combat simulation and a group of DMs, respectively, where both criteria and preference weights are uncertain. Ranking and selecting the "best option" from simulation has been extensively studied [14] for a single response metric, and there are a few studies [5, 6] which apply different MCDM methods to address multi-criteria problems. However, all methods were restricted to using a deterministic weighting. Consequently, we propose to represent the uncertain criteria and weight information by probability distributions. This provides a general and flexible way to represent various forms of uncertain information.

Lahdelma et al. [13] succinctly describe that there are several MCDM methods that support the handling of uncertain criteria information by way of various techniques including that *threshold models* are used in ELECTRE [16] and PROMETHEE [4], and that probability distributions are used in multi-attribute utility theory (MAUT) [10] and in Stochastic multi-criteria acceptability analysis (SMAA) methods (see e.g. [11, 12]). The use of probability distributions facilitates the handling of dependencies between the uncertainties in criteria and preference weights. Most relevant in this paper is that in MCDM defence problems and AHP weighting approach has been used successfully [5, 15]. This success is attributable to the approach enabling the weights of attributes to be decided upon in a manner which is consistent and robust. It also allows DMs to check the consistency and consensus of the rankings of the relative importance among the involved criteria in group decision-making. We propose a MCDM procedure based on the concept of feasible weight space (FWS), rather than based in deterministic weight vectors, since the weights should incorporate the DMs' preference information to the greatest possible extent. The FWS concept is not new, but it is compatible and helpful in our MCDM method. It is indeed much more flexible than having deterministic weight vectors for the purpose of MCDM. Consequently, we use AHP weighting method and the uncertainty interval to obtain the FWS in this paper. Additionally, we combine these approaches with SMAA for the multi-criteria comparisons of different vehicle options based on representative combat simulation data.

16.2 Evaluation Criteria of LCVs in Combat Simulation

The evaluation of LCVs in this study is based on a mission which required the rapid seizure of an urban fringe area. The Blue force (the side for which the different vehicle options were being assessed) was a Combat Team (CT) consisting of three infantry platoons, a Tank troop and two direct fire support vehicles. The Red (enemy) force was approximately one third the size of Blue and its mission was to delay Blue by 24 h. Other key aspects of this scenario were

– The CT assembly area was located approximately 10 km from the objective within a heavily vegetated area
– There was only one viable road-based approach to the objective
– Red had some indirect fire support options
– Red planned an ambush on the road-based approach

Three options considered here were

– Option A = Wheeled, 30 mm Cannon, Armour X
– Option B = Tracked, 35 mm Cannon, Armour Y
– Option C = Tracked, 30 mm Cannon, Armour Y.

Armour Y, in theory, offers slightly more protection than Armour X but both offered similar protection against the calibre of munitions encountered in this study. A number of metrics were used to provide an assessment of each option's lethality, protection, signature and knowledge acquisition capabilities. Protection was assessed by comparing casualty rates of infantry and various levels of vehicle damage. Similarly, the lethality-based metrics considered the same effects on enemy platforms and infantry. The signature metrics measured the total number of unique Blue force entities that were detected during the simulation and also the range at that each Platoon was first acquired. The knowledge metrics captured the same information in relation to acquisitions of Red entities by the Blue force. In addition, a binary mission success metric was used to determine whether Blue achieved its mission. In this case, mission success was based on the successful seizure of its objective within 24 h and before reaching certain attrition thresholds to critical assets. SMEs were consulted in order to extract the appropriate plans and tactics for both the Blue and Red forces to undertake their respective missions. These plans were then encoded in the COMBATXXI simulation, a stochastic, entity-based closed-loop combat simulation developed by the United States Army [3]. The simulation was replicated 10,000 times for each of the three vehicle options in order to allow for statistically robust analysis of the results. Each replication terminated either when Blue successfully completed its mission or reached one of the defined failure thresholds described earlier. In order to illustrate the methodology and to simplify the evaluation process, we limit our scope to survivability and lethality 'catastrophic kill' metrics only in this paper and note that it is straightforward to extend the process to include many more metrics. The considered metrics are described as below:

1. Blue Infantry Fighting Vehicle (IFV) K Killed: The number of Blue IFV vehicles which suffered a catastrophic kill, and this is a descending metric for blue survivability.
2. Blue Tank K Killed: The number of Blue tanks which suffered a catastrophic kill, and this is a descending metric for blue survivability.
3. Blue Infantry Killed: The number of Blue tanks which suffered a kill, and this is a descending metric for blue survivability.
4. Red IFV K Killed: The number of Red IFV vehicles which suffered a catastrophic kill, and this is an ascending metric for blue lethality.

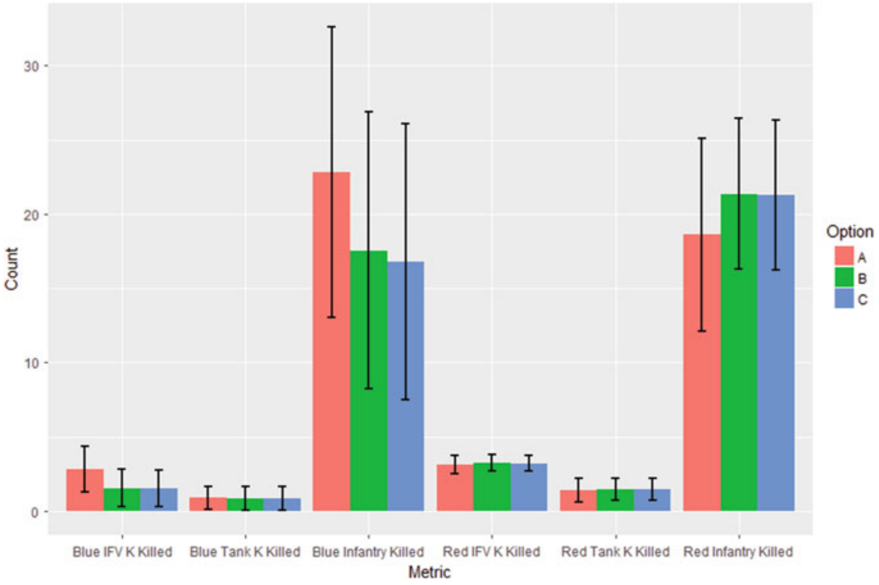


Fig. 16.1 Metrics representation by mean and standard deviation

5. Red Tank K Killed: The number of Red tank which suffered a catastrophic kill, and this is an ascending metric for blue lethality.
6. Red Infantry Killed: The number of Red infantry which suffered a kill, and this is an ascending metric for blue lethality.

The criteria representing metrics 1–6 are denoted by $c_1, c_2, \ldots, c_6$, respectively, where c_1, c_2, c_3 are descending and c_4, c_5, c_6 are ascending criteria for Blue force mission effectiveness. The mean and standard deviation of the metrics are shown in Fig. 16.1.

16.3 Stochastic Multi-criteria Acceptability Analysis

The evaluation and ranking of LCVs is a MCDM problem in which information in both criteria performance metrics and preference weighting is uncertain. Consequently, SMAA is applied here to handle this problem. SMAA was developed in [11] and extended to SMAA-2 in [12]. This family of models is based on the utility theory for quantitative and qualitative problems. The logic and notation for SMAA is developed in a clear manner in [20], and we adopt without modifying the mathematics and notation from that paper. Combined Heat and Power units were evaluated in [20], whereas in this paper we analyze and rank the LCV options. At the core of the method, Monte Carlo simulations are used to calculate the multi-dimensional integrals for the stochastic variables of rank acceptability indices, holistic acceptability indices, central weight vectors and confidence factors.

16.4 Weighting Method by Analytic Hierarchy Process

The preference choice by DMs is usually represented by the weighting process which structure and quantify the subjectivity of DMs. There are numerous methods developed to assess weights for multi-criteria evaluation, and as previously outlined in [5], the following are widely used:

- Direct Weight: subjective weights are entered directly.
- SMART and SMARTER methods [7]: defines relative importance using “swing weights”.
- Trade-off method [18] using pairwise trade-offs between criteria to define the weights.
- Pairwise Weight Ratios Method [2]: similar to the trade-off method, instead of defining a complete trade-off, simply enter the ratio between the two criteria weights.
- AHP Weight Method [17]: using criteria pairwise comparison hierarchically and obtaining priority weights via eigenvector method.

We apply the AHP weight assessment method to generate the relative weights associated with the criteria considered in this study, in which all criteria of the same

Table 16.1 AHP judgement scale for pairwise comparisons [17]

Verbal scale	Numerical values
Equally important	1
Moderately more important	3
Strongly more important	5
Very strongly more important	7
Extremely more important	9
Intermediate values	2, 4, 6, 8

hierarchical level are pairwise compared with respect to the corresponding criteria in the next higher level, and a matrix of pairwise comparisons is obtained. In order to represent the relative importance of one element over another, a judgement scale for pairwise comparisons is introduced in [17]. For this relative comparison, the judgement scale of Table 16.1 is used. It allows the quantifying of verbal comparisons to be expressed into the corresponding numbers. Particularly in our study for the LCV priority weight evaluation, a n-by-n matrix A of pairwise comparisons is constructed. The components a_{ij} $(i, j = 1, 2, \ldots, n)$ of the matrix A are numerical entries by the pairwise judgement scale which reflect the relative importance of the criteria i over j with respect to the LCV mission effectiveness. The relative priority weights among the n elements of the matrix A is computed via the principal eigenvector method and is normalized to obtain the priority vector. We prefer to use the AHP weighting method in this study because the consistency and consensus of the DM's judgements in group decision-making are measured.

16.4.1 Consistency of DM's Preferences

It was shown in [17] that the pairwise judgement for matrix A is consistent if $a_{ij}a_{jk} = a_{ik}$ for all i, j and k. However, human judgement is not always consistent. For example, one provides $a_{12} = 3$ (C_1 is moderately more important than C_2) and $a_{23} = 5$ (C_2 is strongly more important than C_3). It should follow that $a_{13} = 15$. If the numerical value of the judgement a_{13} is different from 15 then there would be a certain level of inconsistency in the pairwise comparison matrix. The question is how much inconsistency is acceptable. In order to address this issue, AHP calculates the Consistency Ratio $CR = CI/RI$, by comparing

- The Consistency Index $CI = (\lambda_{max} - n)/(n - 1)$, where λ_{max} is the largest eigenvalue of the matrix A, with
- The Random consistency Index RI of a random pairwise comparison matrix where the judgements have been entered randomly, and therefore it is expected to be highly inconsistent [17]

It has been shown that a CR of 0.1 or less is acceptable to continue the AHP analysis [17]. If the consistency ratio is greater than 0.10, it is required to revise the judgements to address the inconsistency. In our study, we have applied AHP for group decision-making situations. Consequently, it is necessary to analyze individual judgement as well as to measure the consensus of the group judgements. The group judgements are aggregated by geometric mean rather than arithmetic mean because arithmetic aggregation procedure violates the property of reciprocity [17] so that inconsistent matrices arise. Other advantage is that geometric mean can compensate for the individual inconsistent matrices to consistent group judgements [1]. Let m be the number of DMs and $a_{ij}^{(k)}$—the element of the individual matrix $A^{(k)}$ elicited from DM$_k$. We define

$$A^* \overset{\text{def}}{=} (a_{ij}^*), \text{ where } a_{ij}^* = \left(\prod_{k=1}^{m} a_{ij}^{(k)}\right)^{\frac{1}{m}}$$

Shannon entropy defined in [8, 9] is used with its two independent components (alpha and beta diversity) to derive an AHP consensus indicator (S^*). Denote by $w^{(k)} = \{w_i^{(k)} \mid i = 1, \ldots, n\}$ the weights resulting from the individual pairwise comparison matrix $A^{(k)}$ elicited from DM$_k$. Consensus indicator (S^*) is calculated as follows:

- Shannon alpha entropy:

$$H_\alpha = \frac{1}{m}\sum_{k=1}^{m}\sum_{i=1}^{n} -w_i^{(k)} \ln w_i^{(k)}$$

- Shannon gamma entropy:

$$H_\gamma = \sum_{i=1}^{n} -w_i^{(\text{avg})} \ln w_i^{(\text{avg})}, \text{ where } w_i^{(\text{avg})} = \frac{1}{m}\sum_{k=1}^{m} w_i^{(k)}$$

- Shannon beta entropy: $H_\beta = H_\gamma - H_\alpha$
- Consensus indicator: $S^* = \dfrac{n\exp(-H_\beta) - \exp(H_\alpha^{\min})}{n - \exp(H_\alpha^{\min})}$

  with $H_\alpha^{\min} = -\left(\frac{M}{n+M-1}\right)\ln\left(\frac{M}{n+M-1}\right) - \left(\frac{n-1}{n+M-1}\right)\ln\left(\frac{1}{n+M-1}\right)$,

  s ($M = 9$ for the fundamental AHP scale)

No consensus (high diversity of judgement) when $S^* < 50\%$ while high consensus (excellent agreement of judgement) when $S^* \geq 80\%$.

16.5 Uncertainty in Criteria Evaluation and Preference Information

Uncertain criteria values can be represented as a probability distribution and the most widely used distributions are uniform and normal [12]. For simplicity, we apply the discrete distribution to represent uncertain measurements in this paper.

Preference information from DMs can be uncertain and imprecise in most MCDM problems. Although this preference information can be represented by an arbitrary probability distribution in SMAA, DMs may prefer to express the preferences in term of constraints for the weight space. Consequently, the FWS is represented as a $n - 1$ dimensional simplex. Here, we consider the following types of weight constraints which were recommended in [12]:

1. Weight intervals $w_j \in [w_j^{\min}, w_j^{\max}]$
2. Weight ratios interval for trade-offs $w_j/w_k \in [w_{jk}^{\min}, w_{jk}^{\max}]$
3. Linear inequality constraints $Aw \leq c$
4. Nonlinear inequality constraints $f(w) \leq 0$.
5. Partial or complete weights order $(w_j > w_k)$

In this study, we examine the FWS with different constraints, for example, the interval constraint from the AHP priority weight vector and its uncertainty bound. The FWS idea presented here provides the flexibility and potentially the ability to improve the weight elicitation process in MCDM studies. We obtained one "consistent" AHP weighting priority vector for the LCV, and we then assume an interval of ±50% to obtain an FWS as shown in Table 16.2. As in [20], 10000 iterations of Monte Carlo simulation were used for the estimation of the stochastic variables that are needed for the application of the SMAA-2 method utilizing the FWS.

16.6 Result and Analysis

For simplicity to illustrate the proposed methodology, we use linear utility function and discrete probability distributions for criteria values. Note that there is no constraint on probability distribution of criteria values which can be Gaussian, Beta, Weibull, etc. The criteria values are scaled by the best and worst criteria values as follows: $u_{ij} = (x_{ij} - x_{ij}^{worst})/(x_{ij}^{best} - x_{ij}^{worst})$. Additionally, we consider four different kinds of criteria weights as below:

1. No weighting preference. This type of weight is chosen in case the DMs are not able to give a preference statement.
2. Ordinal weighting is used in case that the DMs are only willing to make the order preference statement.
3. Exact weighting is obtained by AHP for consistency and consensus from DMs.
4. Interval weighting is obtained by extending the consolidated AHP weighting to ±50% to obtain an FWS.

SMAA-2 with 10000 Monte Carlo iterations was used in the simulation, and the expected error limits are less than 0.01 [19]. The weighting input is shown in Table 16.2 for comparing the conclusions based on these weighting methods.

The stochastic acceptability and holistic rank indices calculated by using SMAA-2 method with different types of FWS is shown in Table 16.3. The a^h column presents

Table 16.2 Criteria weighting input for FWS

Metrics or criteria	Ordinal weight	AHP weight	AHP interval weight
Blue IFV Killed	2	0.247	(0.124, 0.371)
Blue Tank Killed	1	0.38	(0.19, 0.57)
Blue Infantry Killed	3	0.165	(0.083, 0.248)
Red IFV Killed	5	0.073	(0.037, 0.11)
Red Tank Killed	4	0.102	(0.051, 0.153)
Red Infantry Killed	6	0.035	(0.018, 0.053)

Table 16.3 Stochastic acceptability and holistic rank index

Weight type	Option	b^1	b^2	b^3	a^h
No preference	Option A	0.17	0.26	0.57	0.39
	Option B	0.43	0.36	0.21	0.63
	Option C	0.40	0.37	0.22	0.61
Ordinal weight	Option A	0.16	0.31	0.53	0.40
	Option B	0.43	0.34	0.23	0.65
	Option C	0.41	0.35	0.24	0.62
AHP weight	Option A	0.16	0.31	0.53	0.40
	Option B	0.42	0.35	0.23	0.62
	Option C	0.42	0.34	0.23	0.62
AHP interval weight	Option A	0.16	0.31	0.53	0.40
	Option B	0.43	0.34	0.23	0.62
	Option C	0.41	0.36	0.23	0.62

holistic acceptability indices, and the columns b^1, b^2, b^3 are the rank acceptability indices. Centroid meta-weights are used to calculate the holistic acceptability indices. Additionally, central weights and confidence factors are calculated and shown in Table 16.4 for better discrimination. The confidence factors indicate that option A should be rejected due to very low probability of achieving the first rank. It is very hard to discriminate option B and C based on rank acceptability indices, holistic ranks, however, indicate that option B is very marginally better than option C. Note that AHP and AHP interval weight methods produce almost identical results which confirm that there is no random effect on the ranking result by varying the weights in this case. It is simpler to implement AHP weight; however, uncertainty in weighting is a common problem for group decision-making, and we, therefore, recommend using SMAA in combination with AHP for the problem with uncertain weighting.

Table 16.4 Stochastic central weights and confidence factor

Weight type	Option	w_1	w_2	w_3	w_4	w_5	w_6	p^c
No preference	Option A	0.15	0.18	0.16	0.17	0.17	0.17	0.15
	Option B	0.17	0.17	0.17	0.17	0.16	0.17	0.43
	Option C	0.17	0.16	0.17	0.16	0.16	0.17	0.42
Ordinal weight	Option A	0.16	0.24	0.41	0.06	0.10	0.03	0.15
	Option B	0.16	0.24	0.41	0.06	0.10	0.03	0.42
	Option C	0.16	0.24	0.41	0.06	0.10	0.03	0.43
AHP weight	Option A	0.25	0.38	0.16	0.07	0.10	0.03	0.16
	Option B	0.25	0.38	0.16	0.07	0.10	0.03	0.42
	Option C	0.25	0.38	0.16	0.07	0.10	0.03	0.42
AHP interval weight	Option A	0.24	0.39	0.16	0.07	0.10	0.03	0.16
	Option B	0.25	0.38	0.16	0.07	0.10	0.04	0.42
	Option C	0.25	0.38	0.16	0.07	0.10	0.03	0.42

16.7 Conclusion

In this paper, three Land Combat Vehicle (LCV) options representing different configurations are considered for multi-criteria evaluation including the criteria of Blue force survivability and lethality. Mission effectiveness metrics of these options were calculated by the closed-loop combat simulation. Ranking and selection methods to determine the “best option” are often restricted to a single response metric or to deterministic weight for multiple metrics. Here, we address these shortcomings by using the SMAA model to account for uncertainties and imprecision in criteria evaluation and weightings, and the FWS to represent feasible preference information from DMs. An AHP weight and the interval are used to determine the FWS. Consequently, four different FWSs are used for the evaluation and ranking of LCV options. The first one is the general weight space which assume the DMs have no preference information, the second is the FWS with ordinal weight, the third is the aggregated deterministic weight from AHP, and the fourth is interval constraints on criteria base on AHP weight. The results show option A is well below option B and C on every SMAA measures for all types of weights and should be eliminated. It is impossible to separate options B and C based on the included metrics, scenarios and SMAA analysis. Option B and C are very similar so it may not be possible to separate them in a simulated environment. We also may need to introduce more design parameters, effectiveness metrics and scenarios to gain more insight of LCV operational effectiveness. Our conclusion is that the combination of the FWS, AHP group weighting and SMAA can make uncertain multi-criteria evaluation and ranking results a reliable and robust approach to the LCV MCDM problem.

References

1. Aull-Hyde R, Erdogan S, Duke J (2006) An experiment on the consistency of aggregated comparison matrices in AHP. Eur J Oper Res 171(1):290–295
2. Belton V, Stewart TJ (2002) An integrated approach multiple criteria decision Analysis. Springer
3. Balogh I, Harless G (2003) An overview of the COMBAT XXI simulation model: a model for the analysis of land and amphibious warfare. In: Proceedings of 71st military operations research society symposium
4. Brans JP, Mareschal B, Vincke Ph (1984) PROMETHEE: a new family of outranking methods in multicriteria analysis. In: Brans JP (ed) Operational Research, IFORS 84. North Holland, Amsterdam, pp 477–490
5. Cao T, Coutts A, Pietsch B (2010) Multi-attributes utility theory and statistical analysis for defence future vehicle options. ASOR Bull 29(2):69–81
6. Emond EJ (2006) Developments in the analysis of rankings in operational research. DRDC CORA TR 2006-37
7. Edwards W, Baron HF (1994) SMARTS and SMARTER: improved simple methods for multi-attribute utility measurement. Organ Behav Hum Decis Process 60:306–325
8. Goepel KD (2013) Implementing the analytic hierarchy process as a standard method for multi-criteria decision making in corporate enterprises—a new AHP excel template with multiple inputs. In: Proceedings of the international symposium on the analytic hierarchy process, Kuala Lumpur
9. Jost L (2006) Entropy and diversity. Oikos 113(2):363–375
10. Keeney RL, Raiffa H (1993) Decisions with multiple objectives. Cambridge University Press, Cambridge, UK, Preference and Value Tradeoffs
11. Lahdelma R, Hokkanen J, Salminen P (1998) SMAA-stochastic multiobjective acceptability analysis. Eur J Oper Res 106:137–143
12. Lahdelma R, Salminen P (2001) SMAA-2: Stochastic multicriteria acceptability analysis for group decision making. Oper Res 3:444–454
13. Lahdelma R, Makkonen S, Salminenc P (2009) Two ways to handle dependent uncertainties in multi-criteria decision problems. Omega 37(1):79–92
14. Nelson BL, Swann J, Goldsman D, Song W (2001) Simple procedures for selecting the best simulated system when the number of alternatives is large. Oper Res 49(6):950–963
15. Nguyen M-T, Cao T (2017) A hybrid decision making model for evaluating land combat vehicle system. In: Syme G, Hatton MacDonald D, Fulton B, Piantadosi J (eds) 22nd International Congress on Modelling and Simulation, MODSIM2017, pp 1399–1405
16. Roy B (1996) Multicriteria methodology for decision aiding. Kluwer Academic Publishers, Dordrecht, The Netherlands
17. Saaty TL (1982) Decision making for leaders. Wadsworth Inc, California
18. Schoemaker PJ, Waid CC (1982) An experimental comparison of different approaches to determining weights in additive value models. Manag Sci 28(2):182–196
19. Tervonen T, Lahdelma R (2007) Implementing stochastic multicriteria acceptability analysis. Eur J Oper Res 178:500–513
20. Wang H, Jiao W, Lahdelma R, Chuanzhi Z, Zou P Stochastic multicriteria acceptability analysis for evaluation of combined heat and power units. Energies 8(1):59–78 (2015)

Chapter 17
The Wheels Versus Tracks Problem for Armoured Fighting Vehicles in the Australian Context

Nikoleta Tomecko and Kasia Krysiak

Abstract In armoured fighting vehicle design, the Iron Triangle concept describes the design tensions that exist between the three primary characteristics of these vehicles: mobility, protection and lethality. Traditionally, wheels and tracks represent two different trade-off instances between different aspects of these three factors and are suited to different operational conditions. To provide some clarity to the wheels vs tracks argument for the ADF, a wheels vs tracks study was undertaken in the Australian context. This study collated results of previous studies and performed a meta-analysis, synthesizing the results to produce an understanding of the impacts of wheels and tracks on operational outcomes, analyzing the current evidence of the strengths and weaknesses of wheels and tracks, and interprets these in different contexts characterized by environmental and operational variables. The results of the meta-analysis show that overall a tracked vehicle will offer a greater operational capability advantage more often. Out of the 72 different contexts defined, 62 show an operational advantage for tracked vehicles. Only nine contexts had an overall utility skewed towards a wheeled vehicle, and in one context wheeled and tracked vehicles were judged as equal. The analysis identified 14 contexts with an intensity rating of extreme, and in all of those contexts tracked vehicles had an operational advantage over wheeled vehicles. In 15 of the 20 contexts judged to be most likely, tracked vehicles had an operational advantage over wheeled vehicles, while the remaining five showed an operational advantage for wheeled vehicles.

Keywords Vehicle mobility · Armoured fighting vehicles

N. Tomecko (✉)
Weapons and Combat Systems Division, DST Group, Edinburgh, Australia
e-mail: Nikoleta.Tomecko@dst.defence.gov.au

K. Krysiak
Joint and Operations Analysis Division, DST Group, Fishermans Bend, Australia
e-mail: Kasia.Krysiak@dst.defence.gov.au

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_17

17.1 Introduction

The wheels versus tracks question which arises as part of design tensions that exist between the three primary characteristics of armoured fighting vehicles, mobility, protection and lethality, continues to be a contentious issue. Traditionally, wheels and tracks represented two different design trade-offs between different aspects of these three factors, hence they are suited to different operational conditions. Any capability decision regarding the choice between wheels and tracks must be made in the context of what operational conditions the vehicle will be used in.

The LAND 400 Project faced such a decision, as it is replacing the armoured vehicle fleets currently in service in the Australian Army, ASLAV and M113, with modern Armoured Fighting Vehicles in the Cavalry Reconnaissance Vehicle and Infantry Fighting Vehicle roles [1]. The capability developers needed to make the wheels versus tracks decision for each vehicle role that is being acquired.

A difficulty in the wheels versus tracks argument is being able to quantify the overall impact of the various strengths and weaknesses of the two mobility classes on the operational outcomes. Any analysis is unlikely to be universally applicable, as different armies have different operational concepts for armoured fighting vehicles. To further complicate the assessment, the operational concept aims to make best use of the available capability, hence the operational concept of a wheeled vehicle is going to be different to that of a tracked one.

This study aimed to assist the decision-making for LAND 400 by providing some clarity for the wheels versus tracks argument in the Australian context, based on already existing evidence.

17.2 Methodology

17.2.1 Introduction

A two-step approach was adopted: Step 1 (previous studies) involved conducting complementary individual studies focused on summarizing the state of knowledge, and Step 2 (this paper) consisted of an over-arching study that analyzed and interpreted the collated data to highlight advantages or disadvantages of wheeled or tracked suspensions, as they apply to armoured fighting vehicles in the Australian context.

The individual studies focussed on different aspects of the wheels versus tracks argument:

- Literature review of wheels versus tracks studies identifying the performance differences of the vehicle types [2],
- Operational Lessons learnt extracting the mission impacts of these physical differences [3],

- Constructive simulation comparing wheels versus tracks [4].

Quantitative meta-analysis has long been successfully used in medical research to systematically assess the results of previous research to derive conclusions about that body of research [5, 6]. Selected parts of the reported results of multiple primary studies are used to generate a single data set and formal statistical methods are applied to this data. Meta-analysis is increasingly being used to analyze and reinterpret qualitative data from multiple studies, with synthesis approaches such as thematic analysis, content analysis and framework synthesis being utilized [6–8].

For this study, the data from individual studies was synthesized using a semi-quantitative method. Firstly, the operational contexts where armoured fighting vehicles might operate in the Australian context were defined. Secondly, the evidence presented in each of the studies relevant to each context was reviewed, and an assessment of the relative advantage offered by wheeled or tracked vehicles within that context was performed. Finally, the broader implications of the likelihood of operating within each of the contexts were analyzed through a most likely and most dangerous analysis.

This study focused on operational performance. Other considerations that are essential to capability decision-making such as the combat logistics implications of deploying vehicles, the costs of operating each fleet or other fundamental inputs to capability (FIC) considerations (e.g. training or support) have not been considered.

17.2.2 Data Set Description

To establish a comparison of the relative operational advantage that one type of vehicle could offer over the other vehicle type, three separate studies were conducted, each providing a different lens on the problem:

Study 1: Argument Diagramming the Tracked versus Wheeled Debate.

This study conducted a literature review of studies that investigated the differences between wheeled and tracked vehicles. It provided a comprehensive review of the physical attributes of the vehicles and how well they performed tactically in a variety of conditions. The findings were presented as a series of argument maps.

Study 2: Operational lessons learnt.

Study 2 looked at the recent operational lessons learnt, regarding wheeled and tracked vehicles and the associated trafficability, performance and protection across different terrain types and threats. Growth potential for emerging capabilities and its potential impact on the current operational performance of tracked and wheeled vehicles was also assessed in light of these lessons learnt. Close combat vehicle performance was assessed within the context of how the Australian Army intends to use such vehicles.

Study 3: Constructive simulation model.

This study performed a detailed analysis of how wheeled and tracked vehicles compare in a scenario specifically designed to bring out the differences between tracked and wheeled vehicles. The analysis was done using Combat XXI, a closed-loop simulation tool. The scenario was based around a Combat Team (CT) completing a rapid seizure of an urban fringe area.

17.2.3 Context Definition

Scenario-based analysis is a recognized method for dealing with uncertainty especially in future planning such as capability development. In future planning, a number of possible situations are represented as scenarios. Capability recommendations are usually made on the basis of being able to handle the most dangerous and most likely possibilities, based on rigorously testing capability options against a small number of selected scenarios.

The aim of this study was to go beyond a detailed analysis against a small set of scenarios, but rather to cover as many potential scenarios as possible. This shifted the focus from specific instances of using a particular vehicle configuration, to a generic assessment of the relative operational advantage each vehicle would offer given its capabilities. This required scenarios are to be defined not in terms of specific missions, but in terms of the different sets of conditions that affect how wheeled and tracked vehicles can be used.

In order to define these generic operational contexts, a morphological analysis was conducted with the aim of describing the different operational conditions relevant to the tracks versus wheels comparison in the Australian context. Morphological analysis was selected as it aims to describe all possible states of a multi-dimensional problem. In order to establish all states, a set of dimensions was determined. The dimensions and the values of the dimensions were established using the following sources:

- The Australian Capability Context Scenarios (ACCS)
- The three studies described above
- The LAND 400 OCD [1].

Each of these sources provided a different way of defining the scenario dimensions. A comparison was done of the various categories and several combinations of the dimensions were tested for their comprehensiveness and interpretation. The final set of contexts were needed to be comprehensive to cover as many different situations as possible; however, manageable enough to enable individual analysis and interpretation of results.

The final set of dimensions after consolidation is shown in Table 17.1. It defines 84 different combinations of these factors. Each generated context was assessed for feasibility, with infeasible contexts removed from further consideration. This resulted in 72 unique context descriptors for analysis.

Table 17.1 Dimensions used to define operational contexts in wheels versus tracks study

Environment type	Season	Operation type
• Desert	• Dry	• Joint land combat
• Savannah	• Wet	• Irregular warfare
• Jungle	• Snow	• Peacekeeping
• Mountainous		• Disaster relief
• Modern urban		
• Rural urban		

17.2.4 Performance Comparison

17.2.4.1 Operational Capability Metrics

A valid comparison of wheeled and tracked vehicles must take into account all aspects of vehicle performance and their individual and combined impact on the overall operational performance of the vehicles. Two main sources were used to generate a set of operational capability metrics, which collectively captured the performance aspects that differ between wheeled and tracked armoured fighting vehicles, resulting in different operational performance. These were the LAND 400 OCD and Study 1 described above (Argument Diagramming the Tracked versus Wheeled Debate).

Study 1 used the available evidence collected from literature to define the operational capability metric hierarchy shown in Error!Reference source not found. This structure captured only those aspects of operational performance that were relevant to the differences between wheeled and tracked armoured fighting vehicles. Protection and Lethality are wrapped up under the broader concepts of Survivability and Fightability, which describe the whole-of-system nature of vehicle operations. Despite the removal of characteristics that do not impact the wheels versus tracks argument, the metric hierarchy used in Study 1 is consistent with the hierarchy used in the LAND 400 OCD, and was adopted for the meta-analysis (Table 17.2).

17.2.4.2 Operational Capability Assessment

In order to assess the overall operational capability of wheeled versus tracked vehicles in each context, each metric was examined to assess the available evidence as shown by the three studies. In general, Study 1 provided the evidence of vehicle performance differences in each context, and Study 2 and Study 3 provided the operational impacts of these differences in performance characteristics. An overall judgement of whether each context is more favourable towards wheels or tracks was made using the following scale:

2 Tracked vehicles represent a significant capability advantage for the commander as their performance characteristics in the assessed metric make them particularly suited to the environment.

Table 17.2 Operational metrics of argument map

Mobility		
Trafficability	Obstacles	Agility
• Natural terrain	• Natural	• Acceleration/dash
– Clay soil	• Constructed	• Stopping
– Sand	• Gaps	• Sustained speed
– Snow	• Vertical steps	• Maximum speed
• Man-modified terrain	• Vegetation	• Slaton (turn/pivot)
• Man-made terrain	• Craters	
– Road	• Rubble/debris	
– Bridges	• Walls/building	
– Canals	• Vehicle roadblocks	
– Tunnels		
Fightability		
Threat defeat	*Human factors*	
• Accuracy—stationary	• Crew fatigue	
• Accuracy—on the move	• Dismount fatigue	
• Firepower		
Survivability		
Susceptibility	*Vulnerability*	*Recoverability*
• Visual signature	• Damage resistance	• Repair
• Infrared signature	– Driveline	• Replacement
• Acoustic signature	– Structure	
• Shaping	• Damage tolerance	
	• Damage control	

1 Tracked vehicles represent a slight capability advantage for the commander with better performance characteristics in the assessed metric in this environment.

0 Wheeled and tracked vehicles are equal, as the impact of performance characteristics that are important for this metric are evenly spread between wheeled and tracked vehicles.

−1 Wheeled vehicles represent a slight capability advantage for the commander with better performance characteristics in the assessed metric in this environment.

−2 Wheeled vehicles represent a strong capability advantage for the commander as their performance characteristics in the assessed metric make them particularly suited to the environment.

The choice of ± to represent wheels and tracks is arbitrary and does not affect the results. During evaluation, half-points on the scale were permitted where it was felt that a choice between the predefined levels could not be made. An example of

Table 17.3 Example rating of relative operational capability for a given context

Effectiveness aspect	Mobility	Obstacles	Agility	Susceptibility	Vulnerability	Threat defeat	Human factors
Context A	2	2	2	−1.5	−1	1	1
Context B	2	2	1	−0.5	−1	0	0

an assessment for two different contexts is shown in Table 17.3. In this example, in Context A, a tracked vehicle would have a strong capability advantage in Mobility, while a wheeled vehicle would have a slight advantage in vulnerability. In Context B, in this example, tracked and wheeled vehicles are equal with respect to threat defeat and human factors, with neither offering an operational advantage.

17.2.4.3 Overall Operational Capability Assessment

In order to provide an easier interpretation of the relative benefit of acquiring a wheeled versus a tracked armoured fighting vehicle, each context was assessed for the overall degree to which a tracked or a wheeled vehicle has an operational advantage, based on the individual scores.

A heatmap visualization was chosen, since it allowed an intuitive way of assessing the trade-off in performance offered by wheeled or tracked vehicles, and enabled the use of a single heatmap table containing all the contexts. Contexts in which wheeled vehicles were favoured were shaded green, while those in which tracked vehicles were preferred were shaded red. The entire heat map provides a quick overview of the proportion of contexts in which it is much more advantageous for the commander to have a wheeled vehicle (green cells), the proportion where it is advantageous to have a tracked vehicle (red cells) and the proportion where both vehicles are comparable (white cells).

Since the individual assessments used an ordinal scale, a series of overall utility measures was used in order to achieve a ranking of all contexts. A number of overall utility scores were considered for their suitability, examining their ability to convey the operational impact relationships identified in Study 2 and Study 3. Three metrics were selected for comparison of the contexts is as follows:

- **Total third powers**. Sum of squares is a utility score often used in situations where the assessment scale is not linear. Based on the operational impact relationships identified in Study 2, the operational impact of a vehicle with a strong performance advantage is disproportionately greater than the impact of a vehicle with a slight advantage. Third power rather than square of individual scores was used as the scale is centred on zero, with positive and negative signs used to signify preference for tracks and wheels, respectively.[1]

[1] This has no impact on the ranking, and is used purely to streamline the calculations.

- **Median score**. Median is the most appropriate measure where ordinal scores are used or when the distribution of individual measurements is skewed, and both of these situations apply here. In this case a distribution skewed to one side indicates a relative advantage, making the median a potential metric.
- **Total of individual scores**. A simple weighted average is the most commonly used utility function due to its simplicity and ease of interpretation. In this case, a non-weighted average was used. This is because the individual scores already include a weighted component, as each individual assessment includes consideration of the relative importance of the performance characteristic in each context.

A ranking of all contexts was produced using these three metrics together using the following outranking rules:

- A context outranks another where all three metrics individually resulted in the same ranking of the two contexts.
- A context outranks another where there are any two metrics that individually produce the same ranking.
- Given two metrics that produce an identical ranking (i.e. they are unable to distinguish between the contexts), a context outranks another based on a comparison of the third metric.
- Given two metrics at odds with each other, and a third that cannot differentiate, the metrics are used to inform outranking in the following order: total third powers metric, median score, total score.

As this produced an ordering of all the contexts, no further rules were developed.

17.3 Results

17.3.1 Overall Operational Effectiveness

A comparative analysis shows that the total utility of all the contexts is skewed, with more assessments favouring tracks. This means that, overall, a tracked vehicle will offer a greater operational capability advantage more often, i.e. the differences in favour of tracked vehicles are greater and there are more of them.

Figure 17.1 shows the overall scale of the difference for all contexts. Contexts are ranked with decreasing operational advantage for tracked vehicles, with blue and white shaded groups representing clusters of the same overall operational effectiveness. The area above the line represents the contexts where tracked armoured fighting vehicles have an operational advantage. The area below the line represents the contexts with an advantage for wheeled vehicles. It is clear that tracked armoured fighting vehicles offer a greater operational advantage in many more situations.

Examining the utility as indicated through the metrics used, in 62 out of the 72 scenarios the overall utility was skewed towards a tracked vehicle. There would

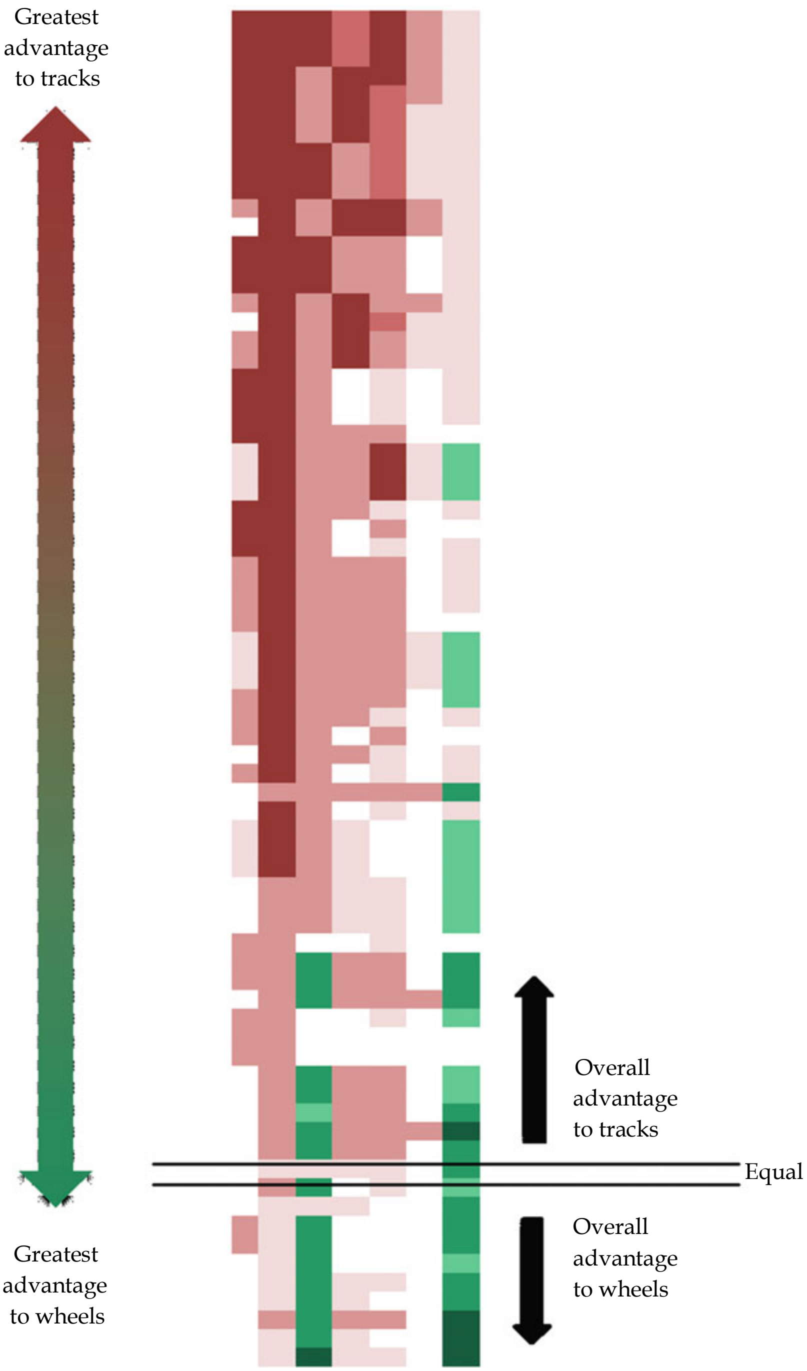


Fig. 17.1 Visualization of the scope of relative advantage of tracks (in red) versus wheels (in green) in all contexts

be a risk to achieving mission success if a wheeled vehicle was used in situations that are represented by these 62 contexts. Only nine contexts had an overall utility skewed towards a wheeled vehicle. There was only one context where all total utility measures were 0, i.e. where the individual performance characteristic differences between wheeled or tracked vehicles would equalize, despite having different individual advantages or disadvantages.

The analysis showed that in most contexts the advantage for tracks lies in their ability to cope with different types of soil especially in the wet, and their ability to negotiate obstacles without a dramatic reduction to speed. The ability to negotiate obstacles in particular offers a strong advantage in over half the contexts. Based on the literature, there are no circumstances where wheeled vehicles would outperform tracked vehicles on their ability to negotiate obstacles. In a contested environment with an active adversary, this would present a significant operational advantage.

Further examination of the metrics revealed that a tracked armoured fighting vehicle had an overall operational advantage on all except the human factors metric. The strongest advantage was in their ability to deal with obstacles. On this metric, a tracked armoured fighting vehicle always had an operational advantage in all contexts. The remaining two submetrics of mobility: trafficability, agility and the two survivability submetrics (vulnerability and susceptibility) all showed approximately the same level of advantage towards the tracked vehicle, followed by threat defeat. The human factors submetric was the only one where a wheeled vehicle had an overall advantage, though the margin of that advantage was smaller than the margin of the tracked vehicle on the threat defeat submetric.

With respect to individual scores, there were only five individual instances where a wheeled vehicle was judged to have a strong operational advantage over a tracked vehicle, with four of those scores on the human factors submetric, offering smoother rides on roads and more space inside the hull to accommodate better ergonomics, which translated to an operational advantage in a specific context.

17.3.2 Most Dangerous and Most Likely Analysis

The contexts were given a score by a military SME relating to the likelihood and intensity of the situation described by each context. A four-point scale was used for both of these ratings.

The contexts rated the most dangerous (i.e. those that had the highest rating of extreme on their Intensity) and their assessments of the relative advantage for the two platform types is shown in Table 17.4. The contexts rated the most likely (i.e. those that had the rating of Likely on their Likelihood) and their assessments of the relative advantage for the two platform types are shown in Table 17.5. There were 20 such contexts out of the total 72. In fifteen of those contexts, tracked vehicles had an operational advantage over wheeled vehicles, while the remaining five showed an operational advantage for wheeled vehicles. However, two of those (one showing an advantage for tracks and one for wheels) might be considered as showing no advantage for either platform type due to close scores.

Table 17.4. There were 14 such contexts out of the total 72, and in all of those 14 contexts, tracked vehicles had an operational advantage over wheeled vehicles.r

Except for the context defined by rural urban - joint land combat - snow, the contexts rated as extreme in their intensity all appear in the top 50% when ordered according to the level of operational advantage for a tracked vehicle. From a capability development point of view, this means that a fleet of wheeled vehicles would accept a greater level of operational risk than a tracked fleet while operating in all those environments.

The contexts rated the most likely (i.e. those that had the rating of Likely on their Likelihood) and their assessments of the relative advantage for the two platform types are shown in Table 17.5. There were 20 such contexts out of the total 72. In fifteen of those contexts, tracked vehicles had an operational advantage over wheeled vehicles, while the remaining five showed an operational advantage for wheeled vehicles. However, two of those (one showing an advantage for tracks and one for wheels) might be considered as showing no advantage for either platform type due to close scores.

17.3.3 Additional Considerations

The analysis focussed on the documented differences and similarities between the operational performance of wheeled and tracked armoured fighting vehicles, i.e. those that there is evidence for either in scientific literature or operations lessons learnt, as described in the three studies. There is a large body of evidence in the domain, which covers many aspects of vehicle characteristics, and the three individual studies made an explicit effort to ensure there are no gaps. Additional effort was made to cross-reference the metrics within the study with the consolidated operational needs defined for armoured fighting vehicles within the LAND 400 Project. This comparison found no gaps in the three studies, i.e. no operational performance metrics for which there was no evidence at all.

Nevertheless, it is possible that there are differences in the operational performance between the vehicles that have not been investigated, and have thus not been included in the study, or that additional studies will include conflicting or contradictory evidence. Similarly, technology advances might result in making some of the current evidence obsolete for a new generation of armoured fighting vehicles. If that occurs, the analysis with respect to the affected operational performance metrics will need to be re-evaluated.

Table 17.4 Contexts rated Extreme in their Intensity

			Manoeuvre			Survivability		Fightability	
Terrain type	Operation type	Weather	Mobility	Obstacles	Agility	Suscepta-bility	Vulnera-bility	Threat defeat	Human factors
mountainous	joint land combat	dry	2	2	2	1.5	2	1	0.5
mountainous	joint land combat	wet	2	2	2	1.5	2	1	0.5
mountainous	joint land combat	snow	2	2	2	1.5	2	1	0.5
jungle/forest	joint land combat	wet	2	2	1	2	2	1	0.5
jungle/forest	irregular warfare	wet	2	2	1	2	1.5	0.5	0.5
mountainous	irregular warfare	snow	2	2	2	1	1.5	0.5	0.5
jungle/forest	irregular warfare	snow	1	2	1	2	1	0.5	0.5
modern urban	joint land combat	dry	0.5	2	1	1	2	0.5	-0.5
modern urban	joint land combat	wet	0.5	2	1	1	2	0.5	-0.5
modern urban	joint land combat	snow	0.5	2	1	1	2	0.5	-0.5
modern urban	irregular warfare	dry	0.5	2	1	1	1	0.5	-0.5
modern urban	irregular warfare	wet	0.5	2	1	1	1	0.5	-0.5
modern urban	irregular warfare	snow	0.5	2	1	1	1	0.5	-0.5
rural urban	joint land combat	snow	0	1	-1	1	1	1	-1

Table 17.5 Contexts rated Likely in their Likelihood

Terrain type	Operation type	Weather	Manoeuvre			Survivability		Fightability	
			Mobility	Obstacles	Agility	Susceptability	Vulnerability	Threat defeat	Human factors
Littoral	Peacekeeping	Wet	2	2	0.5	1	1	0	0
Jungle/forest	Peacekeeping	Wet	2	2	0.5	1	0.5	0	0.5
Littoral	Disaster relief	Wet	2	2	0.5	0	1	0	0
Jungle/forest	Disaster relief	Wet	2	2	0.5	0	0.5	0	0.5
Jungle/forest	Peacekeeping	Dry	0	2	0.5	1	0.5	0	0.5
Jungle/forest	Disaster relief	Dry	0	2	0.5	0	0.5	0	0.5
Modern urban	Disaster relief	Dry	0.5	2	0	0.5	0	0	−0.5
Modern urban	Disaster relief	Wet	0.5	2	0	0.5	0	0	−0.5
Modern urban	Peacekeeping	Dry	0	1	1	0.5	0.5	0	−0.5
Modern urban	Peacekeeping	Wet	0	1	1	0.5	0.5	0	−0.5
Savannah	Disaster relief	Wet	1	1	0	0	0	0	0
savannah	Disaster relief	Snow	1	1	0	0	0	0	0
Rural urban	Irregular warfare	Wet	0	1	−0.5	1	1	0	−1
Rural urban	Peacekeeping	Wet	0	0.5	0.5	0.5	0.5	0	−1
Rural urban	Disaster relief	Wet	0	0.5	0.5	0.5	0	0	−1
Desert	Peacekeeping	Dry	1	0.5	−1	0	0	0	−1
Savannah	Disaster relief	Dry	0	0.5	−1	0	0	0	−0.5
Rural urban	Irregular warfare	Dry	0	1	−1	1	1	0	−2
Rural urban	Disaster relief	Dry	0	0.5	−1	0.5	0	0	−2
Rural urban	Peacekeeping	Dry	0	0.5	−2	0.5	0.5	0	−2

17.4 Conclusion

A meta-analysis of three studies comparing wheeled and tracked armoured fighting vehicles was conducted. The studies were designed to be complementary: an argument map of the current state of knowledge of differences between wheels and tracks, a closed-loop simulation looking at impacts on mission success and operational implications of these differences.

The results of the meta-analysis have shown that overall, a tracked vehicle will offer a greater operational capability advantage more often. An overwhelming majority of the defined contexts favoured a tracked vehicle, and a force option where this capability is unavailable would face a distinct operational disadvantage.

References

1. LAND 400 Integrated Project Team (2014) Program LAND 400 Land Combat Vehicle System Operational Concept Document, Version 4.0, Commonwealth of Australia, 2014a.
2. Kempt, N, Hemming, D, Tomkinson, P (2017) Argument diagramming the tracked versus wheeled debate, DST–Group-TR-3338
3. Matsumara J, Gordon J, Steeb R, Boston S, Lee C, Padilla P, Parmentola J (2016) Assessment of Tracked and Wheeled Vehicles for Australian Mounted Close Combat Operations: Lessons Learned in Recent Conflicts, Impact of Advanced Technologies, and System Level Implications, RAND National Security Research Division, PR-2496/1-AUS
4. Shine D, Chau W, Dexter R, Grieger D, Lohmeyer D, Owen K, Russack S, Sanderson D (2017) LAND 400 Phase 3 Capability Trade-Off Study—The Closed-Loop Analysis of Mobility Profiles (CLAMP), DST-Group-CR-2017–0034
5. Haidich AB (2010) Meta-analysis in medical research. Hippokratia. 14:29–37
6. Lorenc T, Felix L, Petticrew M, Melendez-Toress GJ, Thomas J, Thomas S, O'Mara-Eves A, Richardson M (2016) Meta-analysis, complexity, and heterogeneity: a qualitative interview study of researchers' methodological values and practices. Syst Rev 5:192
7. Timulak L (2009) Meta-analysis of qualitative studies: a tool for reviewing qualitative research findings in psychotherapy. Psychother Res 19(4–5):591–600
8. Barnett-Page E, Thomas J (2009) Methods for the synthesis of qualitative research: a critical review. BMC Med Res Methodol 9:59
9. Australian Government (2014) Department of Defence. Australian Capability Context Scenarios, Department of Defence, Canberra

Part V
Other Novel Applications and Data Analytics in Defence

Chapter 18
Evolutionary Algorithms for Force Structure Options

Connor Hicks, Elizabeth Kohn, and Thitima Pitinanondha

Abstract A modern Defence Force consists of a diverse range of capabilities to support missions at the tactical, operational and strategic levels. Designing a balanced and affordable force structure to meet Government strategic objectives and assure national security has always been a challenge. Force design is a centralized and enduring process within the Australian Defence Organisation that seeks to translate Government strategic objectives into a coherent force structure within specified time and budget envelopes. This process increasingly relies on analytical approaches and tools such as wargaming, simulation and optimization techniques. This paper investigates evolutionary algorithms (EAs) as a potential tool for generating and evaluating force structure options. EAs can evaluate an extremely large solution space of force mixes at a much faster rate than human cognition to determine a balanced and affordable force structure option according to an objective function. This paper also discusses the implementation of a software framework, dubbed "FORCESIGHT", which can be customized by developers to model any scenario where the use of EAs is appropriate. Based on the outcomes of a trial of FORCESIGHT, it is clear that the EAs approach could provide a result of respectable quality. It is demonstrated that EAs can lead to large increases in efficient evaluation of potential improvements to the Force-in-Being and Future Force.

Keywords Evolutionary algorithms · Wargaming · Force design · Force structure option · Modelling and simulation

C. Hicks · E. Kohn (✉) · T. Pitinanondha
Department of Defence, Defence Science and Technology Group, Canberra, Australia
e-mail: elizabeth.kohn@dst.defence.gov.au

C. Hicks
e-mail: connor.hicks@defence.gov.au

T. Pitinanondha
e-mail: thitima.pitinanondha@dst.defence.gov.au

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_18

18.1 Introduction

A modern Defence Force consists of a diverse range of capabilities to support missions at the tactical, operational and strategic levels. Designing a balanced and affordable force structure to meet Government strategic objectives and assure national security has always been challenging due to budgetary constraints and likely future threats and challenges. This process increasingly relies on analytic approaches and tools such as wargaming, simulation and optimization techniques.

Force design is a centralized and enduring process within the Australian Defence Organisation (ADO) which seeks to translate Government strategic objectives into a coherent force structure within specified time and budget envelopes. The Force Design Division within the Australian Defence Force Headquarters (ADFHQ) is responsible for the overall design of the military capability to provide the Government a capable, agile and potent force. This is achieved by continuous testing of the Force-in-Being and planned force, and guiding the design and development of a balanced and affordable Future Force.

Simulation is a powerful tool for exploration of future force structures. However, even when a simulation is available for a particular scenario, finding the balanced and affordable force is not trivial. Evolutionary algorithms (EAs) are optimization techniques for searching a defined solution space such as possible force designs to find the best mix of capabilities to achieve missions within the specified scenario. It also provides a performance metric for direct comparison to other force mixes in the form of a fitness score.

This paper investigates EAs as a potential tool for generating and evaluating force structure options. EAs can evaluate an extremely large solution space of force mixes at a much faster rate than human cognition to determine a balanced and affordable force option according to an objective function. This paper proceeds to describe the implementation of a software framework, dubbed "FORCESIGHT", which can be customized by developers to model any scenario where the use of EAs is appropriate. Two trials of the framework complete the analysis, one to demonstrate the general capabilities of EAs, and another to demonstrate in particular the sensitivity of EAs to minor changes in constraints.

18.2 Evolutionary Algorithms

An EA is a nature-inspired optimization method, which imitates the basic principles of life and applies genetic operators to an individual [1]. Effectively, the algorithm iteratively evolves a sample of individuals in order to determine the optimized solution to a given problem, as measured by an objective function. This solves the problem above as the fitness scores assigned by the objective function can be used as a metric for direct comparison of the effectiveness of potential solutions.

EAs typically start by generating a random population of sample individuals and rating them against the objective function. The algorithm then selects the prime individuals for reproduction and applies the genetic operators of mutation and crossover. This process repeats until either convergence of the population or a failsafe is activated to prevent the algorithm from consuming too many resources [1].

Some of the main strengths of EAs are the ability to handle difficult multi-modal problems, where many local optima may exist in the search space, and the relative ease of decomposing a problem into smaller and more easily evaluated sub-problems [1]. The ability to handle multi-modal problems provides more globally optimum results, as the algorithm has a reduced chance of converging on a local optima. The ability to perform problem decomposition means that the algorithm is able to rapidly determine which components of a solution are desirable, and recycle those attributes in subsequent generations.

18.3 Methodology

In a real-world scenario, a force is comprised of "units" which are grouped into "elements". A unit is defined as a single unique entity. For simplicity, only one type of unit can be contained in an element.

Each unit type has its own strengths and weaknesses compared to other unit types, such as ease of mobility and ability to detect hostile units. Each unit type is also associated with a relative cost of resources, and is outfitted with up to two types of weapons.

When two or more forces enter conflict, it is the units within the element group which engage other hostile element groups. The location in which these engagements take place is denoted as an "area". The effectiveness of an engagement is determined by the relationship between the attacking units' weapons and the defending units' type. For example, rifles and mortars have drastically different effects against armoured vehicles. To summarize

- A Unit is a single unique Entity;
- A Element is a group of like Entities and
- A Force is a group of Elements.

The FORCESIGHT framework operates on a three-step algorithm. The reproduction step generates a new population of Individuals (i.e. a Force) from either (a) a previous parent population or (b) by randomizing the attributes of new Individuals. The evaluation step iterates over the offspring population of Individuals and evaluates them against the Fitness Evaluator to determine each Individual's fitness score. The selection step applies a selection strategy to the combined parent and offspring populations to determine which individuals are candidates for survival; these Individuals become the next parent generation. The algorithm repeats these steps until it determines that it has converged on a global maximum, or it reaches the maximum number of generations.

During the design of FORCESIGHT, it was decided early that the module would consist only of the evolutionary aspects of the algorithm. In development terminology, the Fitness Evaluator and Individual classes would remain as blueprints for developers to create custom implementations of the problem to be modelled, as shown in Fig. 18.1. The three green ovals at the bottom of Fig. 18.1 represent either object descriptions in the case of Units and Area, or outcome descriptions in the case of Effects. The Units Comma Separated Values (CSV) file contains the values describing the capabilities of each unit in the scenario. The Area CSV file contains a description of the scenario's battlefield by dividing it into Areas, which roughly simulate which units are in the vicinity of each other. Finally, the Effects CSV file provides a table for determining the effectiveness of available weapons against different unit types.

There are numerous factors that should be taken into account with regard to the execution of an evolutionary algorithm. In this research, only the objective function and the selection strategy employed by the evolutionary algorithm were examined in depth. Exploration of other factors, such as crossover and mutation methodologies, were deemed out of scope of this research and remained static throughout the trials. The crossover methodology used was the single-point crossover method, and the mutation methodology was restricted to a variant of bit-string mutation, where the value of a chromosome was regenerated based on the mutation chance of the parameter.

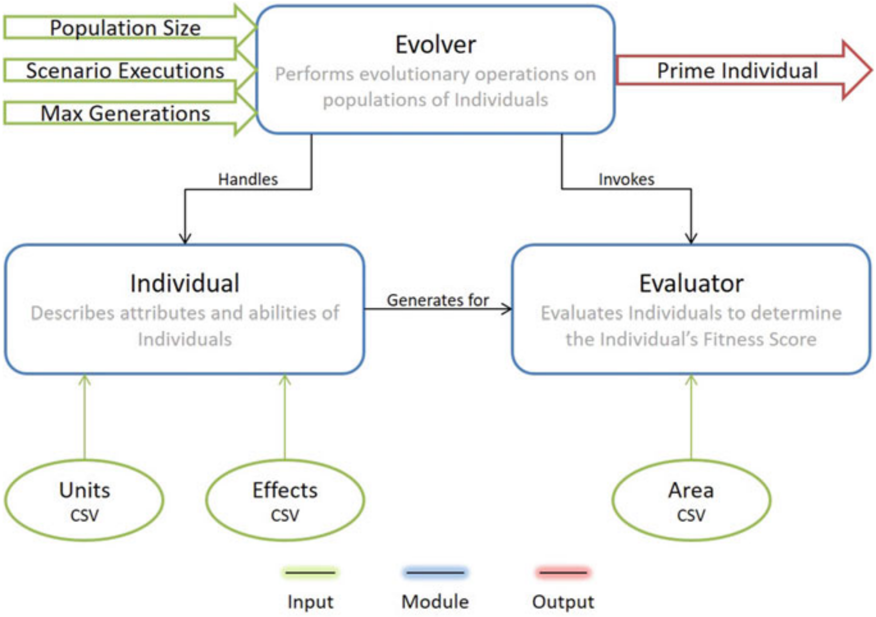

Population Size
Scenario Executions
Max Generations
Evolver
Performs evolutionary operations on populations of Individuals
Prime Individual
Handles
Invokes
Individual
Describes attributes and abilities of Individuals
Generates for
Evaluator
Evaluates Individuals to determine the Individual's Fitness Score
Units
CSV
Effects
CSV
Area
CSV
Input
Module
Output

Fig. 18.1 Overview of FORCESIGHT

The objective function $f(x_i)$ determines an Individual's fitness score. The objective function is always specific to the problem being optimized, and implementation will be described in the next section.

FORCESIGHT supports four types of selection strategies: Tournament, Truncation, and stochastic/non-stochastic Roulette [2]. These can be defined as follows:

- Truncation—the top percentage of Individuals are selected for progression;
- Tournament—Individuals are divided into groups, with the best Individual from each group selected for progression;
- Roulette (NS)—Individuals are selected randomly for progression, however, a higher grade increases the chance of selection; and
- Roulette (S)—Individuals are selected pseudo-randomly at set intervals of scoring, however, a higher grade increases the chance of selection.

The final component to a successful EA is the intelligence to know when to stop. FORCESIGHT implements the Convergence Stability Percentage Metrics (1) from the ANSYS software application [3]:

$$\frac{|\mu_i - \mu_{i-1}|}{\text{Max} - \text{Min}} < \frac{S}{100}, \quad \frac{|\sigma_i - \sigma_{i-1}|}{\text{Max} - \text{Min}} < \frac{S}{100}. \tag{18.1}$$

In Eq. (18.1) μ = Mean of Population Fitness, σ = Standard Deviation of Population Fitness, Max, Min = Max/Min Fitness in Original Population, i = Current Population; and S = Convergence Stability Percentage.

These metrics compare the mean and standard deviation of the current and previous populations against a user-defined percentage denoting an accepted level of convergence. If both metrics are below this percentage, then the algorithm considers the population converged, which is to say the algorithm believes that there are no further improvements that can be made to the population of Individuals. Accurately detecting convergence is an important step in the algorithm, as premature convergence may result in suboptimal solutions and late convergence unnecessarily wastes time and computational resources.

18.4 Implementation

18.4.1 Trial Preparation

In order to test FORCESIGHT, trial Fitness Evaluator and Individual modules were created based on the scenario examined by Revello et al. [4] expanded with informed assumptions, such as the effect of different weapons platforms against different unit types. This work can be found detailed below.

The trial scenario was created as a standard RED vs BLUE engagement. The BLUE force's primary objective is to maintain a port blockade for 48 days, while the RED force's primary objective is to destroy the BLUE Carrier (CV-class) units maintaining the blockade.

The scenario is considered a BLUE victory if any conditions below are met:

- The primary objective is completed.
- RED losses exceed a threshold of 25–35% of total unit value (determined randomly).

The scenario is considered a RED victory if any conditions below are met:

- The primary objective is completed.
- BLUE losses exceed 10 units.
- BLUE losses exceed a cost value of 6.0.

The RED force is composed of a static force of 22 submarines—8 air-propelled submarines (AIP-class) and 14 diesel submarines (SS-class). AIP-class submarines were equipped with 8 rounds of torpedoes and 8 anti-ship cruise missiles, while SS-class submarines were equipped only with 16 rounds of torpedoes.

Scenario Execution

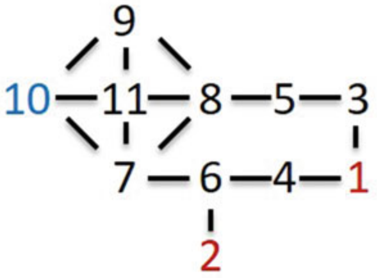
The Fitness Evaluator module in the trial is a turn- and speed-based model. Turns are modelled as days. Each day, the evaluator allows elements to take actions based on the comprising unit's "speed" value. A higher speed value results in more actions each day. Roughly, this results in alternating sides taking turns until all elements have taken their allotted number of actions, where the turn counter increments and the process reoccurs.

Each force consists of up to 19 elements, each of which consists of up to 9 units of the same type. When an element takes an action, it moves to an adjacent area and searches for hostile elements. The success of this search is dependent on the unit's "vision" value; a higher vision value provides a greater chance of successful detection. If a hostile element is detected, the element is able to open fire on the hostile element and reduce the quantity of units in the element. The outcome of engaging a hostile element is determined by the unit's equipped weapons. Each weapon has an effectiveness rating associated with it. A higher effectiveness rating results in more damage being dealt to an element. Every 100 points of damage to an element destroys a unit in that element, and an element is considered destroyed when it contains no more units. BLUE has access to four deployable types of units, as seen in Table 18.1.

Table 18.1 BLUE unit attributes

	Frigate (FFG)	Destroyer (DDG)	Carrier Escort (CVE)	Nuclear Submarine (SSN)
Cost	0.2	1.0	0.4	2.5
Speed	2	2	2	1
Vision	0.5	0.4	0.3	0.8
Wpn1	GM × 20	GM × 40	AST × 16	AST × 8
Wpn2	–	–	–	–

Fig. 18.2 Environment map



An element performs its actions based on one of four behaviours: stationary, hive, boundary and random. In the stationary behaviour, the element does not move to a new area. The hive behaviour makes the element patrol back and forth from a central “hive” area and adjacent areas. The boundary behaviour results in the element patrolling the areas surrounding the area containing the CV-class units. Finally, a random behaviour forces the element to move to an adjacent area at random. In all behaviours, the element searches for and engages threats after movement.

After every turn, the evaluator examines the victory conditions and events to determine if the scenario has been won or lost, and if anything extraordinary occurs between turns, such as reinforcements or a change in orders.

The scenario takes place in an environment consisting of 11 Areas, illustrated in Fig. 18.2. Areas 1, 2 and 10 serve as the “spawn” area, or the area where units enter the battlefield, for the respective coloured units.

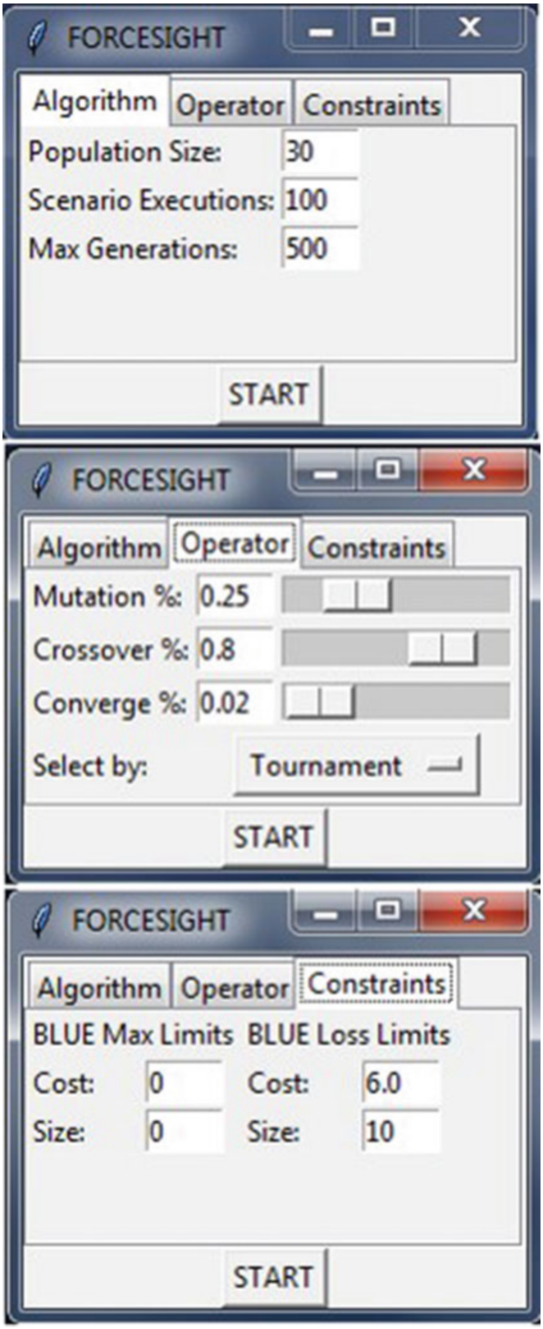
The objective function for the scenario (2) is similar to Revello’s approach. It emphasizes win rate, but also takes into consideration the total cost of ownership of the force mix. In this way, the most effective force can be described as the force that is capable of winning the scenario consistently with minimal units on hand.

$$f(WR, C) = WR - \frac{C}{200}. \tag{18.2}$$

In Eq. (18.2), WR = Win Rate of Individual in Scenario and C = Total Relative Cost of Individual. The Cost value is divided by 200 as a way to reduce the weight of the value in the objective function, as win rate is deemed of greater import than cost.

With the trial modules prepared, FORCESIGHT is now ready to determine the optimal solution to the scenario. The application is supported by a Graphical User Interface, which allows the executor to further customize particular aspects of the algorithm.

Fig. 18.3 Parameter settings for a FORCESIGHT simulation



Population Size: 30
Scenario Executions: 100
Max Generations: 500

Mutation %: 0.25
Crossover %: 0.8
Converge %: 0.02
Select by: Tournament

BLUE Max Limits
Cost: 0
Size: 0

BLUE Loss Limits
Cost: 6.0
Size: 10

Algorithm Parameters and Constraints

Simulation Parameters

The parameters on the upper image of Fig. 18.3 control the execution of the EA. *Population Size* denotes the number of Individuals per generation. *Scenario Executions* denotes how many times an Individual can be tested against the Fitness Evaluator. The *Max Generations* variable adds a failsafe to stop the algorithm from consuming too much time and resources if the algorithm fails to converge below the accepted Convergence Stability Percentage (S in Eq. (18.1)). All of these parameters accept inputs of any positive number.

Genetic Operator Parameters

The parameters on the middle image of Fig. 18.3 control the effect of genetic operators on an Individual. *Mutation* and *Crossover %* denote the chance of the respective operation taking place on an Individual. These have default values of 25% and 80%, respectively. *Converge %* denotes the threshold for the algorithm to consider a population converged. Values between 0 and 1 are acceptable for these parameters. The *Select by* option is a drop-down for selecting one of the four selection strategies.

Objective Constraints

The parameters on the lower image of Fig. 18.3 manage the constraints that apply penalties to the objective function (Eq. 18.2). The *BLUE Max Limits* options denote the amount of units in terms of maximum cost and size that can be included in the BLUE force. The *BLUE Loss Limits* options denote the amount of units that can be lost before the attempt is declared a loss. In both sets of limits, a value of 0 implies that there is no constraint in that particular area, while a positive number reflects the appropriate constraint.

18.4.2 Outcomes

Numerous tests of the evolutionary algorithm were conducted with different combinations of the parameters listed above. On average, each test performs at the rate of examining 10 generations of Individuals per minute with a population size of 30 individuals and 100 scenario executions per individual, on a machine using 16 GB RAM at a speed of 3.6 GHz. This calculates 3000 scenario executions per minute, a favourable timeframe when compared to the time required to evaluate force options by other means such as a wargame. However, the speed of execution is largely because of the simplicity of the simulation, and running EAs using a more realistic simulation would be considerably slower. Figure 18.4 below shows a sample simulation, indicative of approximately 95% of all simulations run.

Many of these tests were affected by convergence, either by premature convergence or late convergence. In Fig. 18.4, each of the selection strategies was tested

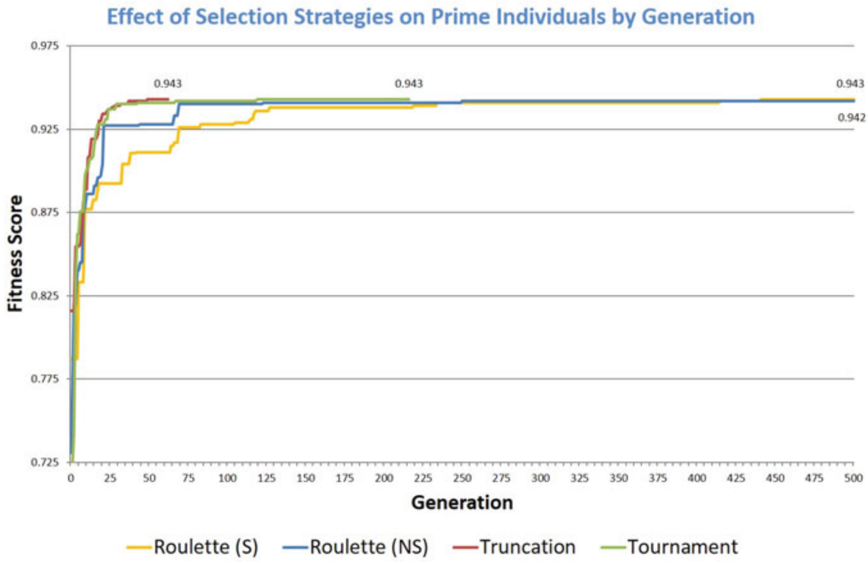


Fig. 18.4 Sample simulation of the effect of selection strategies on algorithm execution

for the ability to converge on a maximum. The vertical axis denotes the fitness score, as calculated by the objective function shown in Eq. (18.2), of the fittest individual in the population, while the horizontal axis denotes the generation of population, up to the maximum generation of 500. Upon examination, it was determined that using the Tournament selection strategy was the most consistent in terms of convergence on the fittest Individual. The ability to converge can be seen by the length of the tail of the graph: the longer the tail the more certain that the strategy has picked the fittest individual to converge on. The Tournament strategy's tail is of respectable length, meaning that the fittest individual in that simulation has been challenged with no improvements found. The Truncation strategy has an extremely short tail, which signifies that the fittest individual hasn't been challenged much at all. The Roulette strategies, on the other hand, have an extremely long tail, signifying that the fittest individual has been challenged to excess with no improvement.

The results of many simulations were very similar with regard to the fittest individual; in most cases, the fittest individual was a force consisting of a group of seven FFG-class units following a boundary behaviour. This can be seen in Fig. 18.5, which displays the evolution of unit composition and behaviour throughout the test using Tournament selection carried out in Fig. 18.4. The vertical axes of each graph show the total size of the force in units, broken down further into groupings of unit types and behaviours, respectively, and the horizontal axes denote the generation of population, up to the point of convergence. As can be seen, the SSN-class (purple in Composition) and DDG-class (green in Composition) units were quickly eliminated from the force mix due to their high cost, as were the stationary (red in Behaviour) and random (purple in Behaviour) behaviours due to limited effectiveness. This test determined

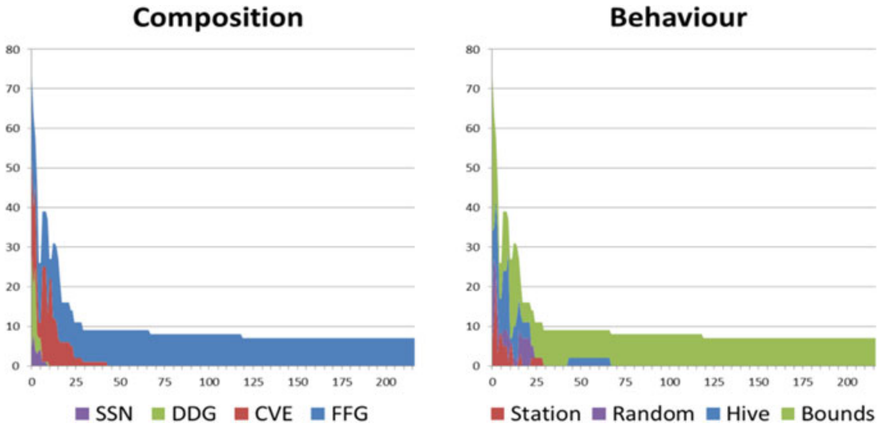


Fig. 18.5 Unit composition and behaviour of fittest individuals by generation

the fittest individual around generation 125, and failed to find improvements between this generation and the point of convergence around generation 225.

In many real-world scenarios, there are additional constraints placed upon scenarios such as this. To test the effect of such constraints, the trial in Fig. 18.5 was repeated, but with the additional requirement that the maximum number of units in the force was five. With this additional constraint, the output changes very dramatically. Instead of a force of seven FFG-class units with a behaviour of boundary, the algorithm instead presents a force consisting of two SSN-class units (purple in Composition) with a stationary behaviour in the main staging area for BLUE (red in Behaviour) as shown in Fig. 18.6. The reasons for this selection are most likely that

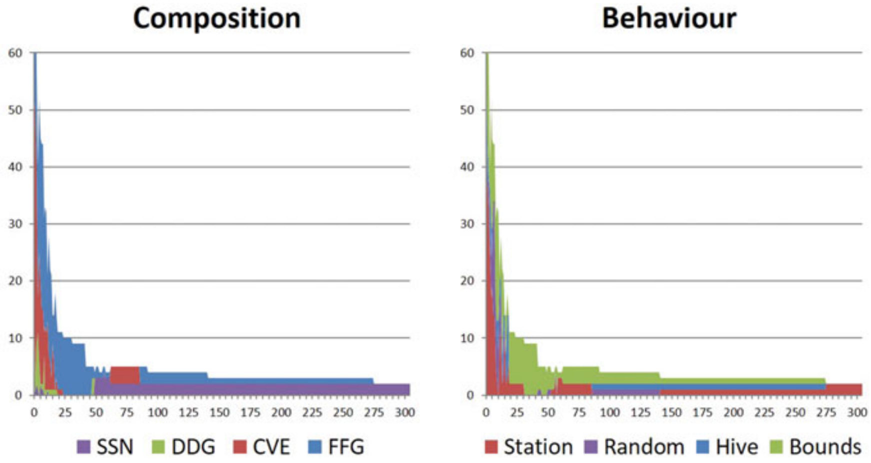


Fig. 18.6 Unit composition and behaviour of fittest individuals by generation where maximum force size = 5

the FFG-class units did not have the necessary firepower to win the scenario reliably when restricted to fewer units. This is a clear demonstration of an EA's inherent sensitivity, in that minor changes to input data and requirements can result in major changes to the outcomes.

18.5 Future Work

One of the defining characteristics of FORCESIGHT is the flexibility of software design. This enables improvements to be made to the base framework relatively easily. One of the first improvements that would be recommended is a way to further mitigate and/or manage the detection of convergence. Currently, it is recommended to execute a particular configuration at least ten times to minimize the impact of premature or late convergence.

There are many assumptions made about the trial scenario, such as not including weather conditions or the inability of either force to adapt to the opponent's strategy. The incorporation of these features would present a more robust system, as the simulation grows closer to the reality of such an engagement. There were also very few trials undertaken with regard to constraining the algorithm. Examining the effects of restricting a specific unit could also be the subject of future improvement.

At a higher level, EAs are only one of the many techniques that comprise the Evolutionary Computation suite. Further investigation into some of these techniques, such as particle swarm optimization, may prove to be effective compared to EAs.

At the framework level, FORCESIGHT can be improved in a number of ways. Firstly, FORCESIGHT is capable of having the output data enriched by displaying trends and patterns of behaviour, as well as specific groupings of units into elements. Furthermore, FORCESIGHT could be enhanced directly by the addition of specific functions. For example, the ability to natively display findings, rather than relying on manual generation, would reduce the time between simulating a scenario and presenting findings. Also, the creation of a standardized blueprint for modelling data would allow developers using FORCESIGHT to be able to collaborate with other developers on modelling scenarios. This could lead to improvements being made elsewhere in the scenario, bringing the model closer to reality.

18.6 Conclusions

The investigation into the aptitude of EAs as a tool for generating and evaluating force structure options reveals that the approach has significant potential to improve the overall process of identifying optimal force structure options. EAs were used to search a very large set of possible force structures and identify the best force mix to win in a given simulated warfighting scenario. Further work is required to develop the FORCESIGHT framework for routine application on more realistic simulations.

However, the success of FORCESIGHT has demonstrated that the approach is worthy of such further development.

Acknowledgements The authors thank Keely McKinlay and Cayt Rowe from the Defence Science and Technology Group, Department of Defence for valuable discussions and comments.

References

1. Rothlauf F (2006) Representations for genetic and evolutionary algorithms. Springer, Berlin
2. Dyer WD (2010) Chapter 3. Selection strategies and elitism. The Watchmaker Framework for Evolutionary Computation for Java. https://watchmaker.uncommons.org/manual/ch03.html. Accessed March 2018
3. SAS IP (2018) Convergence criteria in MOGA-based multi-objective optimization. SHARCNET. https://www.sharcnet.ca/Software/Ansys/16.2.3/en-us/help/wb_dx/dx_theory_MOOConverge.html Accessed April 4 2018
4. Revello T, McCartney R, Santos E (2004) Multiple strategy generation for wargaming. enabling technologies for simulation science VIII. SPIE, Orlando

Chapter 19
A Genetic Programming Framework for Novel Behaviour Discovery in Air Combat Scenarios

Martin Masek, Chiou Peng Lam, Luke Kelly, Lyndon Benke, and Michael Papasimeon

Abstract Behaviour trees offer a means to systematically decompose a behaviour into a set of steps within a tree structure. Genetic programming, which has at its core the evolution of tree-like structures, thus presents an ideal tool to identify novel behaviour patterns that emerge when the algorithm is guided by a set fitness function. In this paper, we present our framework for novel behaviour discovery using evolved behaviour trees, with some examples from the beyond-visual range air combat domain where distinct strategies emerge in response to modelling the effects of electronic warfare.

Keywords Behaviour tree · Air combat · Genetic programming

19.1 Introduction

Constructive simulations are used in defence operations research to support acquisitions, war gaming and the exploration and discovery of new tactical behaviour. In the air combat domain constructive simulations have been used to explore and develop new tactics in a more cost-effective and more flexible manner, especially in light of continuous technological advances such as stealth, advanced avionics and electronic warfare. Many of these simulation systems have incorporated artificial intelligence (AI) agents to model individual and team decision-making for the development and assessment of tactics. AI agents are used to represent the tactical decision-making of aircrew such as pilots. Conventional approaches for developing these agents require the manual encoding of significant domain knowledge. This is costly, labour intensive

M. Masek (✉) · C. P. Lam · L. Kelly
School of Science, Edith Cowan University, Perth, WA, Australia
e-mail: m.masek@ecu.edu.au

L. Benke · M. Papasimeon
Defence Science Technology Group, Fishermans Bend, Victoria, Australia

School of Computing and Information Systems, The University of Melbourne, Melbourne, Australia

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_19

and is not amenable to easily discovering emergent behaviour. The manual encoding of the domain knowledge results in agents that are designed to handle pre-defined opponents, situations and fighter aircraft. The requirement to model new opponents, situations and aircraft systems requires significant manual modification of the pilot agents.

In air combat, selecting a suitable strategy against an opponent depends on the details of the situation (i.e. situation awareness). Factors to consider include the number of aircraft involved in the engagement and the relative capabilities of those aircraft. In a “future warfare” scenario a diverse array of aircraft may take part, with a mixture of autonomous and human-piloted platforms, presenting situations that differ from a commander’s existing experience and complicating the choice of suitable strategy. Also, as the situation changes during the mission, a change in strategy may be required. For example, one side may employ electronic attack, reducing the effectiveness of sensors of the other side. As there is no universal tactic that performs well in all situations, a large library of suitable tactics would need to be developed for a comprehensive evaluation of a situation. New methods and techniques that can help discover novel tactical behaviour for pilots and aircrew are highly desirable.

Behaviour trees, as defined for the game AI domain [1], are one construct that can be used to decompose the behaviour of entities, such as aircraft, into a hierarchical structure that can then be examined and refined. A key advantage of the behaviour tree, as opposed to finite state machines, is that they are more scalable [2] and hence easier to apply to behaviour addressing more complex situations. Besides game development applications, the use of behaviour trees has also been explored in domains such as robotics [3] and for representing medical procedures [4].

The main contribution of this paper is a tactics exploration framework based on behaviour trees evolved with Genetic Programming (GP). Behaviour trees, capturing the behaviour of an aircraft are automatically constructed from a set of low-level aircraft actions and subsequently, can be incorporated into simulators. The behaviour trees are evolved to maximize effectiveness against a particular scenario with agent-based opponents. Unlike other work in the area, such as [5], our approach does not rely on initial solutions provided by subject matter experts. The use of behaviour trees to model aircraft behaviour also makes our approach more scalable to larger and more complex scenarios as opposed to earlier work based on finite state machines [6]. To evaluate our approach, a Beyond Visual Range (BVR) air combat scenario involving one Blue aircraft and two Red aircraft and the ACE-2 simulator [7] is used to explore tactical behaviour of the Blue aircraft under varying conditions.

The rest of the paper is organized as follows: first, background information on behaviour trees and genetic programming is presented along with related work in air combat scenarios. Next, we present our approach for evolving behaviour trees using genetic programming. This is followed by a case study scenario and experiments, discussion of the results and conclusions and future work.

19.2 Background

19.2.1 Behaviour Trees

A Behaviour tree (BT) is a goal-orientated model that represents agent behaviour within a system. A BT can be comprised of many sub-trees with smaller goals. The modular nature of BTs allows for the hierarchical combination of many simple BTs to create more complex behaviour.

BTs start execution at the root node and are comprised of control nodes and leaf nodes. Control nodes dictate the flow through the tree to determine which node should be executed next. Leaf nodes are nodes that can read or react to the simulation environment. As the tree is traversed and each node is executed it returns whether it was successful, it failed or it is currently running.

Control nodes can take the form of sequence, selection or decorator nodes. A sequence node executes each of its child nodes from left to right, until one of them fails. A selector node executes each of its child nodes from left to right until one is successful. A decorator node is used to alter the execution of a child node, for example, dictating that the child node is to be executed a set number of times before returning success.

Leaf nodes are either conditions or actions. Condition leaf nodes typically query the state of the environment returning successful if true or failure if false. Action leaf nodes allow the AI agent to interact with the simulation environment. If the action is completed, the node returns a success, if the action doesn't complete, the node returns a fail. Due to the modular and scalable nature of behaviour trees, actions can range from simple primitive behaviour, such as "turn", through to complex behaviour modelled as sub-trees, such as "engage in BVR combat".

An example of a simple behaviour tree, constructed from high-level actions and control nodes, is shown in Fig. 19.1. To save space, this tree is presented on its side and inverted on the horizontal so that it is traversed from left to right from the root node and top to bottom when traversing the branches. The behaviour corresponds to "Check to see if there is a threat, if there is a threat, launch a missile. If there is a threat and the missile is launched, crank, if you are unable to crank perform missile support" (in the crank action, the pilot turns away from the target while maintaining it locked on their radar).

19.2.2 Genetic Programming of Behaviour Trees

In genetic programming [8], a program is represented by a tree, where non-terminal tree nodes consist of functions and terminal nodes represent the input data for those functions. The algorithm works by starting with an initial population of programs, and evaluating them according to some measure of how the output meets a success criterion (the fitness function). Programs are then selected based on fitness to construct a

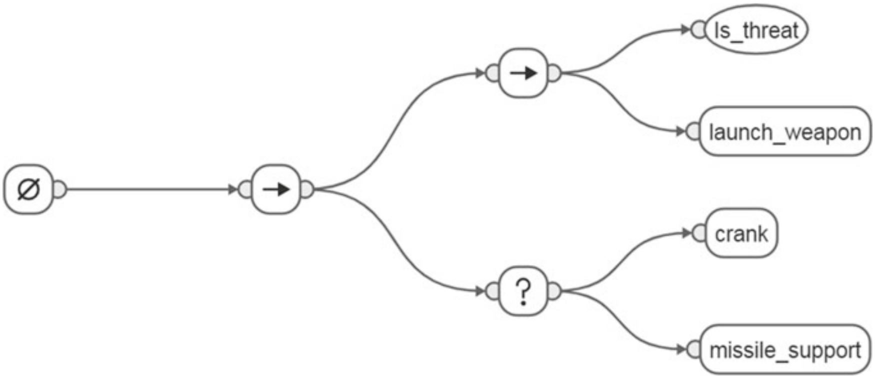


Fig. 19.1 Example behaviour tree (traversed left to right and top to bottom) using high-level actions (launch_weapon, crank and missile_support) with sequence (- >), selector (?) and conditional (is_threat) nodes

new population, with genetic operators used to vary aspects of the individuals. This process continues over a number of generations with the goal of improving fitness as the evolution progresses.

The process of genetic programming can be mapped to the generation of behaviour trees, which can be constructed from terminal and non-terminal components. Instead of using data as terminal nodes, action and condition nodes are used. An action node can equate either to a high-level task, such as ***search*** or ***crank***, or it could be a lower level action, such as performing a ***turn*** or ***flying straight***. A condition node relies on information from the scenario, such as whether an opponent is within a certain range. In place of the functions that act as the non-terminals of a genetic program, these are replaced by the standard behaviour tree control nodes such as ***sequence*** or ***selector***.

19.2.3 Other Work

Previous applications of behaviour tree evolution using a genetic programming approach in games and in a combat environment include agent control [9], development of automated players for a real-time strategy [10] and the exploration of submarine tactics [5].

DEFCON [11] is a multiplayer real-time strategy game, involved players coordinating placements of resources such as airbases, fleets, radars and missile silos in order to attack and destroy their opponents. Lim et al. [10] developed a set of 10 behaviour trees to model the role of an automated player for DEFCON [11]. They employed genetic programming to evolve these behaviour trees for optimizing the placement positions of offensive and defensive components for four sub-goals. The overall strategy used to compete against the in-built artificial intelligence (AI) in DEFCON [11] is constructed by combining the best performing behaviour trees associated with each of the sub-goals and it can defeat the in-built AI more than 50% of the time.

Perez et al. [9] applied Grammatical Evolution, a grammar-based form of GP, to evolve Behaviour trees for direct agent control in Mario AI Benchmark [12] with the aim of improving reactiveness of agents in a dynamic game environment. The behaviour tree controls the actions of the agent throughout a level in the Mario game using an approach involving querying a map of the world via condition nodes and use of action nodes that mimic button presses made by a human player. An advantage of their approach, as outlined by the authors, is that the evolved behaviour trees are human-readable and can be examined in greater depths.

A tactics exploration framework that employed grammar-based genetic programming for evolving tactics in the engagement-level simulation was proposed by Yao et al. [5]. In their work, tactics were represented using a grammar of behaviour tree formalism. The proposed framework incorporated an engagement-level simulator which supports submarine warfare with models of high-fidelity and a grammar-based GP as the tactic exploration engine. Initialization of population involved using 20 expert tactics produced by domain experts and the remaining 80 individuals were generated by completing crossovers and mutation operations on the 20 expert tactics. In terms of their fitness function, it is obtained from the final result of the engagement from the simulator. An individual (tactic) is scored 1 point if the submarine eliminates the enemy and 0 points for a loss. In the case of a draw where both sides survived, fitness is assigned 1/3, and where both sides are destroyed, a value of 2/3. To get a baseline value for fitness, the 20 expert tactics were tested in a simulation and experimental results were then compared to the baseline fitness value. In the two experiments conducted, the respective average fitness did not exceed the baseline value. However, the top 15% tactics at the end of the evolution were found statistically to have a fitness value greater than the baseline.

In comparison to other approaches, the closest work to ours is that of Yao et al. [5] where that approach relies on a set of existing tactics from subject matter experts. In our approach, tactics are constructed automatically based on a set of low-level aircraft actions and performance against an agent-based opponent. In our previous work [6], we have experimented with the evolution of transitions in a finite state machine using a genetic algorithm. That approach, demonstrated on a simple 1 versus 1 within visual range scenario, is limited by the finite number of state instances in the finite state machine and the static number of transitions in the genetic algorithm representation. Moving to behaviour trees allowed us to use their scalability and produce an aircraft controller model whose complexity changes through the evolutionary process in response to the opponent. As such, it becomes easier to consider more complex scenarios such as the ones described in this paper.

19.3 Methodology

We now describe our genetic programming approach to the evolution of behaviour trees which capture the behaviour of an aircraft as a composition of simple aircraft actions in response to the situation of the mission. The core components are the choice of nodes that the behaviour tree is evolved out of and the fitness function which guides the evolution. These are described next, followed by details of the evolutionary operations.

19.3.1 Behaviour Tree Node Set

Our chosen behaviour tree node set consists of the action, condition and control nodes presented in Table 19.1. The Selector and Sequence non-terminal nodes are standard control nodes for behaviour trees and the terminal nodes are a collection of aircraft control specific actions and condition.

The ***Turn*** and ***Level Flight*** action nodes rely on parameter values. The ***Level Flight*** parameter is the aircraft speed, and the ***Turn*** parameters are turn speed, and the amount and direction of yaw and pitch that will be executed in the turn. When concrete instances of these nodes are evolved into the behaviour tree the value of its parameters are set to a random value bound by the modelled aircraft capability.

The condition nodes query the state of the mission in the simulation environment. Similar to our previous work [6], we use the relative position of the current target from the aircraft. The ***Is Correct Position*** node checks whether the current target is within a set of minimum and maximum distance boundaries separately along the x-, y- and z-axes of the aircraft. The boundary values are evolvable parameters for each instance of the node. The ***In Fire Range*** node also uses an evolvable parameter, the percentage of maximum missile range at which the missile could be fired at the target. The remaining conditions are more simple checks. ***Is Threat*** and ***Is Threat Killed*** return whether an opposing aircraft has been detected or eliminated and ***Is Missile in Flight*** returns whether we have already launched a missile.

Table 19.1 The set of terminal and non-terminal nodes used in building behaviour trees in the experiments

Non-terminal nodes	Terminal nodes	
Standard BT control nodes	Action nodes	Condition nodes
Selector	Turn	Is threat?
Sequence	Level flight	Is threat killed?
	Missile support	Is missile in flight?
	Launch weapon	In fire range?
		Is correct position?

19.3.2 Fitness Function

To evaluate each individual solution, the solution is used to instantiate an aircraft agent which flies in the particular scenario we are trying to explore. Aspects of the agent's performance in the mission are evaluated to derive a measure of fitness. The fitness function in any approach based on an evolutionary algorithm should be carefully chosen as a measure of what a superior solution would be and to assign solutions that are inferior a lower value. In the combat domain, there are three clear success levels—win, lose or draw, but besides these it is not simple to rank two solutions with the same success level. For example, a method is required to assess the fitness or effectiveness of two solutions which both have been categorized as a loss. The fitness function needs to evaluate effectiveness of a solution by taking into account additional criteria which influenced the outcome. These additional criteria may include the time taken, the number of missiles used and some measure of the situational awareness of the agent.

The actual calculation and weighting of these factors varies in the literature. For example, Yao et al. [13] measure three factors: *score* which takes one of three possible values depending on win, draw or lose, *safe time ratio* is the fraction of the engagement time that the evaluated aircraft was being tracked by the adversary, and *missile hit ratio* being the fraction of missiles fired at the adversary that hit their target. These three factors were weighted and summed to produce a fitness value, with weights used in the experiments reported being: 0.7 (*score*), 0.1 (*safe time ratio*) and 0.2 (*missile hit ratio*).

Toubman et al. [14] also used *score*, but as a binary value (win or lose), a *time ratio* to give reward based on completing the mission early and a *missile used ratio*, the ratio of missiles fired during the mission. The weights used in that work are: 0.75 (*score*), 0.125 (*time ratio*) and 0.125 (*missile used ratio*).

In both the work of Yao et al. [13] and Toubman et al. [14], the fitness calculation is dominated, through a large weighting, by a single factor (*score*), which has a low granularity of only two to three possible values. In our study, rather than focusing on a single score for the side that we are evolving, we calculate a score for both sides, with fitness calculated as the differential of the two scores, as shown in Eq. 19.1:

$$\mathbf{fitness} = \mathbf{blueScore} - \mathbf{redScore}, \tag{19.1}$$

where ***blueScore*** is a measure of success of the blue side and ***redScore*** is a measure of success of the red side from the perspective of blue. This provides the advantage of increasing the granularity, by having more components as part of the fitness score. Additionally, it allows the assignment of weightings for both blue and red team success parameters, providing more control over the types of strategies that will be favoured. The score for a side is calculated for blue (Eq. 19.2) and red (Eq. 19.3) as

$$\mathbf{blueScore} = \mathbf{A} ^* \mathbf{blueTimeEfficiency} + \mathbf{B} ^* \mathbf{blueWeaponEfficiency}$$

$$+\,\mathbf{C}\,{}^{*}\,\mathbf{blueAwarenessEfficiency} \tag{19.2}$$

$$\begin{aligned}\mathbf{redScore} = \mathbf{D}\,{}^{*}\,\mathbf{redTimeEfficiency} + \mathbf{E}\,{}^{*}\,\mathbf{redWeaponEfficiency} + \\ \mathbf{F}\,{}^{*}\,\mathbf{redAwarenessEfficiency}\end{aligned} \tag{19.3}$$

In calculating these scores we emphasize three factors, winning early (***timeEfficiency***), maximizing efficiency of weapons (***weaponEfficiency***) and maximizing the proportion of the mission during which an aircraft was aware of its opponent's location (***awarenessEfficiency***). These are calculated in Eqs. 19.4–19.6 as follows:

$$\mathbf{timeEfficiency} = \mathbf{NumberOfKillsMade}\,{}^{*}\,(\mathbf{1}-(\mathbf{MissionDuration}/\mathbf{MaxMissionDuration})) \tag{19.4}$$

$$\mathbf{weaponEfficiency} = \mathbf{NumberOfKillsMade}/\mathbf{TotalMisilesFired} \tag{19.5}$$

$$\mathbf{awarenessEfficiency} = \mathbf{TimeSpentTrackingOpponent}/\mathbf{MissionDuration} \tag{19.6}$$

Here ***NumberOfKillsMade*** and ***TotalMisilesFired*** are the total for all aircraft on a particular side. ***TimeSpentTrackingOpponent*** is the sum of the durations for each aircraft on a side where the aircraft was tracking an opponent aircraft. ***MaxMissionDuration*** is the maximum duration before the mission simulation is terminated and ***MissionDuration*** is the actual total duration of the simulation (until the goal of either red or blue is met or ***MaxMissionDuration*** is reached).

The scaling factors: ***A***, ***B***, ***C***, ***D***, ***E*** and ***F*** depend on desired mission characteristics—***A***, ***B*** and ***D***, ***E*** prioritize lethality, ***C*** and ***F*** prioritizes situational awareness. Separating the fitness into a score for each side means that different values can be set for the blue and red constants. For example, from our perspective on the blue side, we might care more about weapon efficiency of blue than red. Likewise, the killing factors may depend of aircraft type—if Blue is a drone, ***D*** and ***E*** (related to score blue loses if shot down) could be lower than ***A*** and ***B***. The resulting fitness, assuming the sum of **A**, **B** and **C** and the sum of **D**, **E** and **F** are each equal to 1, results in a value between –1 and 1, which for the purposes of our experiments we normalized to be in the range from 0 to 1.

19.3.3 Population Management

In the first generation, an initial population of random behaviour trees is created, each tree to a depth in the range 1–4. These trees start from a root node, with a non-terminal arity (the number of child nodes a non-terminal node should have) chosen randomly between 2 and 4. For half of the initial population, the trees are generated until every leaf is at the set tree depth, the other half using a method that stops growing the tree when the first leaf node reaches the set depth, with the remaining leaf nodes set to be terminal nodes. This is a recommended procedure for generating an initial population in genetic programming as it provides an initial population of trees with a variety of sizes and shapes [8]. This differs from the initial population procedure of Yao et al. [5], who use an initial population consisting of 20 trees hand-crafted by experts (and 80 variants of those trees). The benefit of using a random initial population is that there is no limit to population size, there is no need to tie up experts' time in tree production and a source of possible bias that experts may have in particular existing tactics is avoided.

The population of subsequent generations are filled by selecting individuals from the preceding generation and probabilistically applying crossover and mutation operators. The individual with the highest fitness from the previous generation (the *elite*) is always selected to go to a new generation. To select remaining individuals we use tournament selection, with a tournament size of three. This means that to select one individual, we randomly choose three individuals to go into a competition and the individual with the resulting highest fitness score is chosen. Single point crossover was used, where taking two selected individuals, two sub-trees taken from random points in each tree were swapped over. The mutation scheme employed is sub-tree mutation. Here, a random point in the tree is selected and the sub-tree at that point is replaced by a randomly produced sub-tree. To prevent the tree from growing too large, a condition was imposed on the crossover and mutation operators to cap resultant trees to a maximum depth of six.

In our experiments, we run the algorithm for a set number of generations so as to compare evolution runs with different settings. In practice, other stopping criteria can be used to terminate the algorithm, such as when a solution of high enough fitness has been found or if the population has converged to a single optimum where all individuals represent the same solution. The aim in each of these experiments is to evolve a behaviour tree that encodes a tactical strategy for the blue agent, which addresses a version of the scenario.

19.4 Case Study

The scenarios used in this paper were adapted from the tactical situations presented in Shaw [15], a widely accepted treatise on fighter combat tactics and manoeuvring. To demonstrate how we use the proposed approach for tactic exploration, we use a scenario involving BVR combat where there is one aircraft on the blue side and the red side has two aircraft with noise jamming capabilities. Noise jamming is an electronic warfare technique that reduces the situational awareness of an opponent. In this study, we use an effects-based jamming model which reduces the effective range of the opponent's radar by a certain percentage. The starting configuration of the scenario is shown in Fig. 19.2.

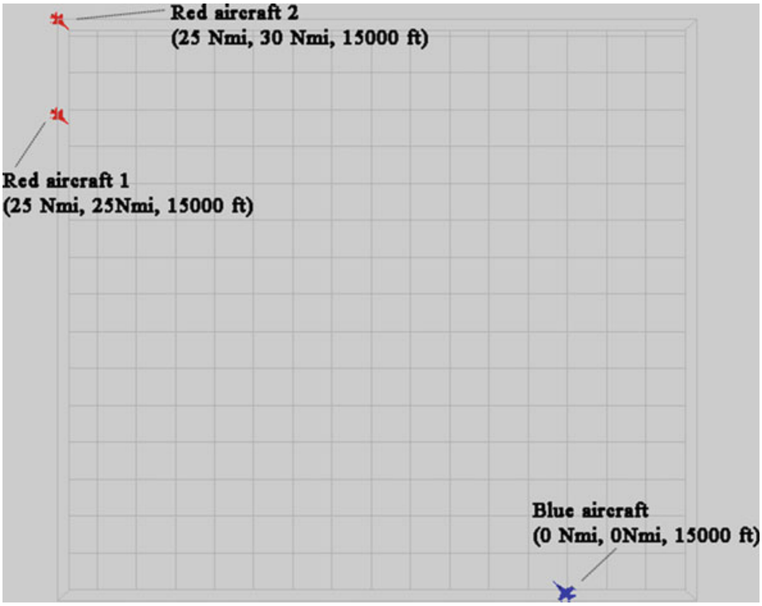


Fig. 19.2 Initial configuration of the case study scenario

All aircraft have the same capabilities apart from the range of the blue agent radar, which is affected by jamming. The radar range when not jammed is 20 nautical miles. The behaviour of the red team aircraft follows a static hand-tuned behaviour tree. In this strategy, once the red aircraft detects the blue, it approaches until within a set missile launch distance and then proceeds to launch a missile and crank. If the missile misses its target, the red aircraft performs a reset manoeuvre, coming back around and re-attempting the missile launch/crank manoeuvre.

In this paper, we explore tactics for future hypothetical unmanned combat air vehicles (UCAVs). Many UCAV options are currently being explored with lighter, smaller and low-cost variants being considered. As a result, it is not expected these will use operational concepts prevalent in current fighter technologies. Consequently, the tactical behaviours and parameters (such as speeds) explored and generated here are not necessarily indicative of current fighter tactics or of any specific future UCAV concept.

Simulation of the scenarios was performed in the ACE-2 simulation environment [7]. This scenario is chosen to demonstrate the approach as it is complex enough for a variety of strategies to emerge and constrained enough so that human-based verification can occur to evaluate whether there are strategies that the evolution has overlooked.

19.4.1 Experimental Setup

We explore four versions of the scenario, where the blue agent radar is jammed from its original range of 20 nautical miles using jamming powers 0, 25, 50 and 75%. Evolution of the blue agent is guided by the fitness function (Eq. 19.1), where the weights **A**, **B**, **C**, **D**, **E** and **F** associated with Eqs. 19.2 and 19.3 were set to 0.6, 0.2, 0.2, 0.8, 0.1 and 0.1, respectively. Putting a higher weighting on **D** than **A** in essence means the penalty for getting killed is higher than the reward for making a kill. Due to the non-deterministic nature of evolutionary operators, experiments were repeated five times to check for consistency of tactics produced. Experiments used a population size of 90, were run for 400 generations and used a crossover probability of 50% and mutation probability of 10%.

19.5 Results and Discussion

The strategies that emerge from the evolved behaviour of trees differ based on the level of jamming that occurs. For each of the jamming levels, an example evolved strategy is profiled in Table 19.2. Visually observing the aircraft behaviour, for 0% and 25% jamming, the blue aircraft flies a smooth run, with moderate speed and gradual turns. In the case of 0% jamming, both red aircraft are shot down, and in the 25% jamming case, one red aircraft is shot down. For the scenario with 50% jamming, the blue aircraft shoots down both red aircraft, with a distinctly different strategy characterized by fast interim speed (after avoiding the first missile) and sharp turns. For the case of 75% jamming, the blue aircraft avoids all conflict and does not attempt to engage either of the red aircraft.

The behaviour trees that are produced by the evolution process are quite large. However, insight into the strategies they represent can be gained by analyzing the path of triggered nodes at key points in the mission. For example, for the 75% jamming scenario, a portion of a sub-tree is shown in Fig. 19.3. Here the green path through the tree represents the only path taken as a threat was not detected during the scenario run due to the highly degraded radar performance of blue. The turn action was executed, with parameters evolved for the turn node resulting in the aircraft pitching up at 59 degrees at a high speed of 488 m/s (600 m/s was the maximum speed for this aircraft). This evasion tactic was the solution with the highest fitness, preserving the blue aircraft although it did not down any red aircraft.

Another example of sub-tree is shown in Fig. 19.4. This was evolved in response to the 50% jamming scenario, with this path through the tree being executed when the blue aircraft detects the first red aircraft. The blue aircraft launches a missile and executes a turn. The parameters of the turn are to speed up to 572 m/s, turn right at 60 degrees and pitch up 62°.

To explore novel behaviour further, we modified the genetic parameters for the 25% jamming case to a population of 400 and 50 generations. This reduces the

Table 19.2 Properties of selected evolved strategies, from the perspective of the blue aircraft

	0% jamming	25% jamming	50% jamming	75% jamming
Kills made	2	1	2	0
Missiles fired	2	2	2	0
First missile fire tactic	Drops speed from 310 to 150 m/s. Levels wings, pitch down at 6^{o}	Flies straight, hold speed at 151 m/s	Increase speed from 265 to 572 m/s, turn at 79^{o}, pitch down at 60^{o}	Pitch up at 59^{o}, hold speed at 488 m/s, don't fire
First red missile avoided by (m)	436	2353	368	No red missiles fired
Second red missile avoided by (m)	922	2717	1133	N/A
Start speed (m/s)	487	231	265	488
First detection speed (m/s)	310	151	151	N/A
First fire speed (m/s)	150	151	151	N/A
First missile evasion tactic	Turn at 15°, pitch down at 6^{o}	Keeps flying straight at 151 m/s	Increase speed from 265 to 572 m/s, turn at 79°, pitch down at 60° then, turn at 60°, pitch up at 62^{o}	N/A
Interim speed (m/s)	150	151	269	N/A
Second fire speed (m/s)	150	151	151	
Second missile evasion tactic	Fly slow so it burns out	Fly slow so it burns out	Fly slow so it burns out	N/A

number of generations but increases the number of individual solutions per generation. This resulted in an interesting behaviour discovery where blue executes a turn that keeps it out of the red aircraft radar cone while allowing it to cover the red aircraft with its own radar and launch a missile. This can be seen in Fig. 19.5. The outcome is that blue is able to destroy both red aircraft without either of the red aircraft launching a missile.

19.6 Conclusions and Future Work

We have proposed our tactics exploration framework based on genetic programming to evolve behaviour trees. The approach is scalable due to the nature of behaviour trees and does not rely on expert knowledge of existing tactics to run, instead starting

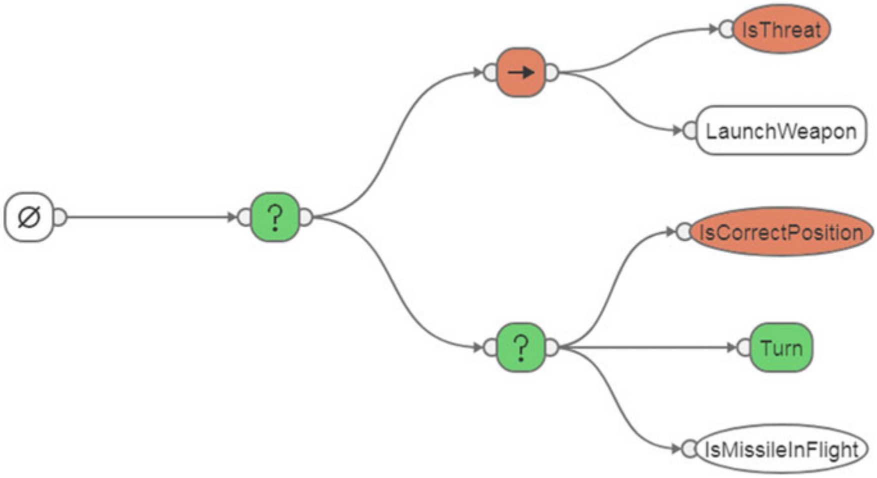


Fig. 19.3 Sub-tree (with some branches and nodes removed to improve clarity) of the evolved behaviour tree for the blue aircraft in the 75% jamming case. Green nodes show success, red failure and uncoloured nodes did not get visited. In this case, which was the only triggered path through the entire scenario, a threat was not detected, the aircraft's position relative to any threat could not be ascertained and so the aircraft performed a turn

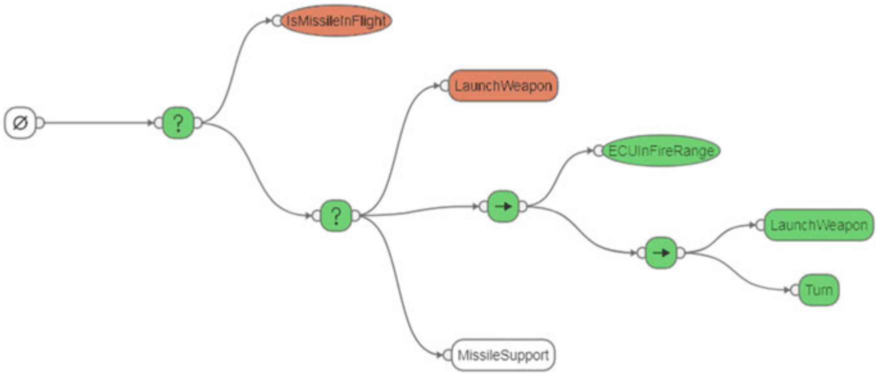


Fig. 19.4 Sub-tree (nodes and branches removed to improve clarity) of the blue aircraft in the 50% jamming case. The green path represents the successfully executed nodes when the blue aircraft detects the first red aircraft, launches a weapon and executes a turn

with a random initial set of behaviour trees constructed from primitive aircraft actions and improving the solutions using an evolutionary process guided by a fitness function. In addition, behaviour trees, capturing the strategy of an evolved behaviour for aircraft, are automatically produced and subsequently, can be incorporated into simulators for analysis and training. The approach was demonstrated on a scenario, where an aircraft with degraded radar capabilities through electronic jamming could

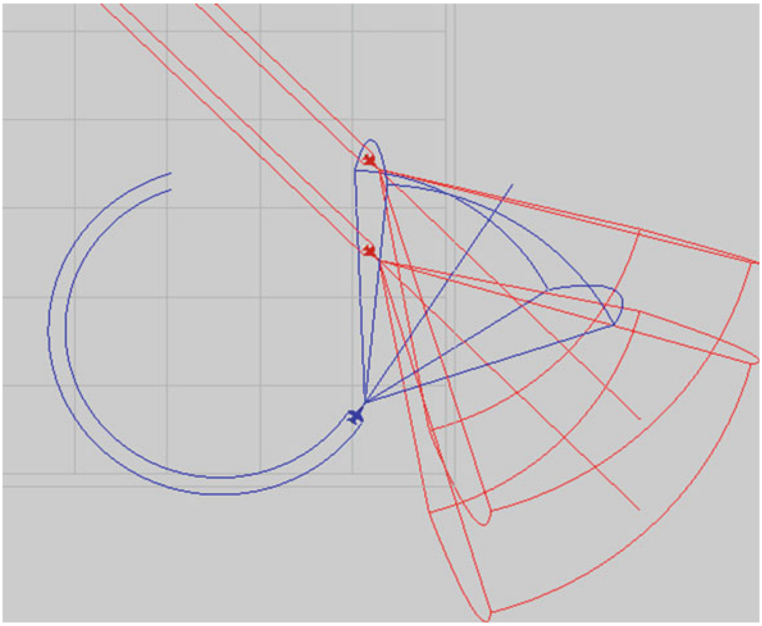


Fig. 19.5 Strategy evolved in response to a scenario with 25% jamming against blue. The blue aircraft circles and approaches the two red aircraft from the side so as to avoid their forward facing radar (radar cones of blue and targeted red shown)

successfully defeat two opponent aircraft using a range of strategies suited to the aircraft's capabilities.

There are several lines of investigation that could be pursued in future work, including further exploring the development of suitable fitness functions, and the mapping of the approach to other combat domains beyond the air combat space.

Acknowledgements The authors would like to acknowledge the work of Richard Brain and Lilia Finkelshtein for work on the ACE 2 Simulation environment and hand-design of an opponent behaviour tree to drive evaluation. This research is supported by the Defence Science and Technology Group, Australia under the Modelling Complex Warfighting Strategic Research Investment. The authors would also like to acknowledge the support of Mr. Martin Cross (DST) who sponsored this research.

References

1. Isla D (2005) Handling complexity in the Halo 2 AI. In: Proceedings of the game developers conference. International Game Developers Association. https://www.gamasutra.com/view/feature/130663/gdc_2005_proceeding_handling_.php
2. Champandard A (2007) Understanding behavior trees. AiGameDev.com 6:328

3. Marzinotto A, Colledanchise M, Smith C, Ögren P (2014) Towards a unified behavior trees framework for robot control. In: 2014 IEEE international conference on robotics and automation (ICRA), May 2014. IEEE, pp 5420–5427
4. Hannaford B (2018) Behavior trees as a representation for medical procedures. arXiv:1801.07864
5. Yao J, Wang W, Li Z, Lei Y, Li Q (2017) Tactics exploration framework based on genetic programming. Int J Comput Intell Syst 10(1):804–814
6. Masek M, Lam CP, Benke L, Kelly L, Papasimeon M (2018) Discovering emergent agent behaviour with evolutionary finite state machines. In: International conference on principles and practice of multi-agent systems, October 2018. Springer, Cham, pp 19–34
7. Ramirez M, Papasimeon M, Lipovetzky N, Benke L, Miller T, Pearce AR, Scala E, Zamani M (2018) Integrated hybrid planning and programmed control for real time UAV maneuvering. In: Proceedings of the 17th international conference on autonomous agents and multiagent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp 1318–1326
8. Koza J (1992) Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge
9. Perez D, Nicolau M, O'Neil, M, Brabazon A (2011) Reactiveness and navigation in computer games: Different needs, different approaches. In: Paper presented at the 2011 IEEE conference on computational intelligence and games (CIG'11), Seoul, South Korea, August 31st–September 3rd 2011. IEEE, pp 273–280
10. Lim CU, Baumgarten R, Colton S (2010) Evolving behaviour trees for the commercial game DEFCON. European conference on the applications of evolutionary computation. Springer, Berlin, pp 100–110
11. DEFCON—Introversion software (2019) https://www.introversion.co.uk/defcon. Accessed 6 March 2019
12. Togelius J, Karakovskiy S, Koutník J, Schmidhuber J (2009) Super Mario evolution. In: 2009 IEEE symposium on computational intelligence and games. IEEE, pp 156–161
13. Yao J, Huang Q, Wang W (2015) Adaptive CGFs based on grammatical evolution. Math Probl Eng, 11pp
14. Toubman A, Roessingh, JJM, Spronck P, Plaat A, Van Den Herik J (2014) Dynamic scripting with team coordination in air combat simulation. In: International conference on industrial, engineering and other applications of applied intelligent systems, June 2014. Springer, Cham, pp 440–449
15. Shaw RL (1985) Fighter combat. Naval Institute Press, Annapolis

Chapter 20
Expanded Basis Sets for the Manipulation of Random Forests

T. L. Keevers

Abstract Random Forests is considered one of the best off-the-shelf algorithms for data mining. However, it suffers from poor interpretability and an opaque decision structure. In this paper, we develop a method for generating an "expanded basis set" for a Random Forest model that captures every possible decision rule and vastly improves the transparency of the classifier. The expanded basis set allows the structure of a Random Forest model to be algebraically manipulated and facilitates a number of operations, including inverse mapping from outputs to the domain of inputs, systematic identification of every decision boundary, and comparison of Random Forest models. The expanded basis set facilitates visualization of the global behaviour of a Random Forest classifier and a data set by combining parallel coordinates with a non-linear binning transformation. The global visualization allows classifier performance to be compared against domain expertise, and areas of underfitting and overfitting to be readily identified. Additionally, the expanded basis set underpins the generation of counterfactuals and anchors—combinations of variables that control the local outputs of a Random Forest model. The basis states can also be used to place bounds on the model stability in response to single or multi-feature perturbations. These stability bounds are especially useful when the model inputs may be uncertain or subject to variation over time.

Keywords Combat simulation · Clustering · Key event identification

20.1 Introduction

Random Forests is one of the most popular machine learning algorithms. It offers strong predictive performance, while requiring minimal performance tuning. However, unlike their constituent decision trees, Random Forest models are difficult to interpret [1]. The decision rules that lead to a particular prediction are usually opaque

T. L. Keevers (✉)
Defence Science and Technology Group, Eveleigh, Australia
e-mail: Thomas.Keevers@dst.defence.gov.au

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_20

and the model structure is difficult to algebraically manipulate. These characteristics can hinder the optimization of a Random Forest classifier and reduce the model's credibility.

A number of model-agnostic methods for understanding model structure have been developed, including partial dependence plots [2], conditional expectation plots [3], measures of feature importance [1] and surrogate models [4]. To the best of our knowledge, the only method developed specifically for interpreting Random Forests is forest floor [5], which uses colour to allow three-dimensional interactions to be captured. While these methods show the aggregate influence of each feature, they don't effectively address the global structure of a Random Forest classifier.

In this paper, we develop a procedure for creating a rule set that describes all possible outputs from a Random Forest model. The key strength of this approach is that it simplifies analysis by reducing a multi-dimensional continuous function to a finite set of points. We extend the method developed for extracting rules from decision trees proposed by Friedman and Popescu as part of their RuleFit algorithm [6]. Combinations of basis states are derived from each feature, which can capture emergent interactions between variables. We call these combinations the *expanded basis set* and the individual node values for each feature the *basis states*.

The expanded basis set representation facilitates the algebraic manipulation of a Random Forest model and is developed in Sect. 20.2. In Sect. 20.3, we show that Random Forests can learn high-order interactions, even from shallow trees, but are unable to capture XOR-type behaviour. The expanded basis set representation is leveraged to create inverse maps from outputs to inputs (Sect. 20.4) and verify whether two Random Forest models are identical (Sect. 20.5). We use the expanded basis set to visualize the global behaviour of a Random Forest model and the data set by combining parallel coordinates with a non-linear transformation, which enables us to identify potential regions over- and underfitting by Random Forests (Sect. 20.6). Lower and upper bounds are placed on the stability of a Random Forest classifier and its predictions (Sect. 20.7). Decision boundaries are identified and used to systematically generate "anchors" [7] and "counterfactuals" [4] given an arbitrary data point (Sect. 20.8). A similar method can be used to generate adversarial examples (Sect. 20.9). These new analysis tools are applied to the *iris* and *digits* data sets from Scikit-learn [8] as proof-of-concept. While the scope of this paper is restricted to Random Forests, the methods can trivially be extended to accommodate other tree ensemble methods like boosting.

We will use the term Random Forests when referring to the algorithm that uses training data to build a model, and the term Random Forest model or Random Forest classifier to refer to a particular instance that has already been trained.

20.2 Expanded Basis Set

Decision trees consist of a number of piecewise linear decision boundaries, with the particular geometry defined by the nodes in the decision tree and the outputs of each decision path [1]. Tree-based descriptions are easy to interpret and are flexible enough to capture a range of non-linear behaviour. However, tree-based descriptions are difficult to algebraically manipulate and contain superfluous information (multiple tree structures can produce globally identical outputs, yet have different topologies). In contrast, it is easy to manipulate rules and verify whether two rule ensembles are equivalent.

Rule ensembles can be extracted from decision trees by tracing a path through the tree and forming the product of indicator functions associated with all the edges of the path [6]. Although decision rules from a single decision tree are easy to manipulate, they prove to be unwieldy when dealing with ensembles of rules, which occurs for ensemble methods like Random Forests. We demonstrate the limitations of the rule extraction method for dealing with Random Forests by considering a simple counter-example consisting of three decision trees (shown in Fig. 20.1).

We consider a Random Forest classifier built from three decision trees, each of which accepts as input a feature vector of length two (x_0, x_1) and outputs a classification of either red or blue. The rules of the decision trees, also shown in Fig. 20.1, are

	Tree 1	Tree 2	Tree 3
Blue	$x_0 > 0$	$x_1 > 1$	$x_1 > -1$
Red	$x_0 \leq 0$	$x_1 \leq 1$	$x_1 \leq -1$

It is easy to show that the subsequent Random Forest classifier consists of separated red and blue regions (Fig. 20.1):

- Blue if ($x_0 \leq 0$ and $x_1 >1$) or ($x_0 > 0$ and $x_1 > -1$)
- Red otherwise

Despite the simplicity of the Random Forest classifier and the child trees, the ensemble demonstrates emergent complexity. All of the individual logical rules involve a single feature, either x_0 or x_1, while the Random Forest classifier is able to learn rules that involve both features simultaneously. This shows that Random Forest models cannot be described by the union of extracted rules from individual trees, so a different approach is required. As an aside, the behaviour observed above suggests that Random Forests outperform decision trees not only because of variance reduction [1], but also because of their ability to learn more complicated basis functions [9].

To increase the power of the extracted decision rules, we introduce an approach in which decision atoms are formed from the ensemble of decision rules, and then formed into different combinations to create decision rules that encompass the expanded basis set (Algorithm 1). We define a *rule atom* as a single logical rule associated with a single feature, but without specifying any output associated with that rule. For our example decision trees, the rule atoms are trivially

$$\textbf{rule atoms} \in (x_0 \leq 0), (x_0 > 0), (x_1 > -1), (-1 < x_1 \leq 1), (x_1 > 1) \quad (20.1)$$

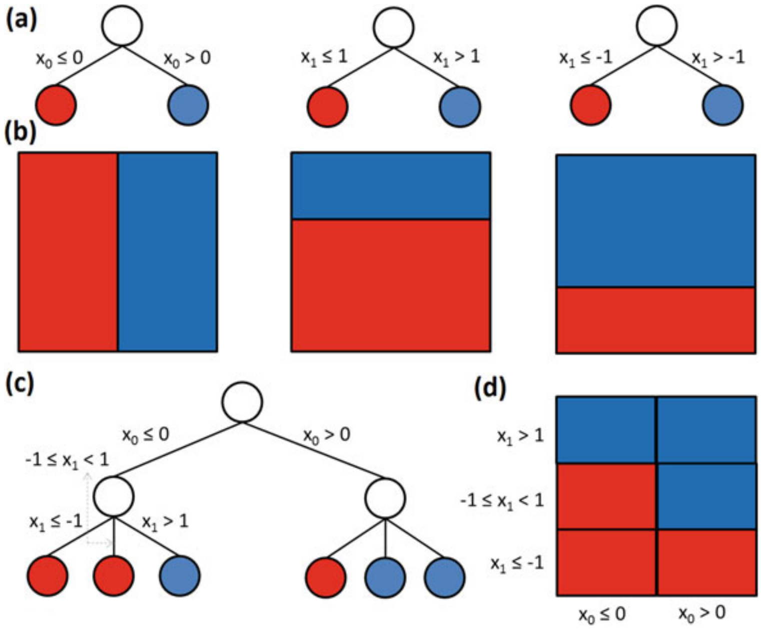


Fig. 20.1 Procedure for generating an expanded rule set for a Random Forest model. **a** Individual decision trees are formed from subsets of features and data points. **b** Graphical representation of the decision trees. **c** A Random Forest model consisting of the three decision trees can also be represented as a single decision tree. **d** Graphical representation of the Random Forest model, see text for more details

The basis states are 0 for x_0, and -1 and 1 for x_1. These rule atoms can be combined to form six rules that form the expanded basis set that can be used to calculate the possible Random Forest model's outputs:

- Blue if $x_0 \le 0$ and $x_1 > 1$
- Red if $x_0 \le 0$ and $-1 < x_1 \le 1$
- Red if $x_0 \le 0$ and $x_1 \le -1$
- Blue if $x_0 > 0$ and $x_1 > 1$
- Blue if $x_0 > 0$ and $-1 < x_1 \le 1$
- Red if $x_0 > 0$ and $x_1 \le -1$

The expanded rule set can be equivalently represented as a single decision tree by iterating through each combination of nodes or as a matrix, as shown in Fig. 20.1.

The rule expansion procedure will reproduce the Random Forest model's behaviour perfectly, but as can be seen above it can produce an unnecessarily long list of rules: 6 rules for the expanded rule set versus the condensed set of 3 rules given above. The rule set could potentially be pruned back to yield a more compact set of rules, although attempting to do so falls outside the scope of this paper.

A practical limitation to creating an expanded rule set is the exponential increase in the number of entries that have to be computed. For example, a Random Forest model consisting of 4 features each with 4 rule atoms has only 256 (4^4) possible outputs, while a classifier with 6 features and 8 rule atoms per feature has 1679616 (6^8)

possible outputs. For high-dimensional problems with tens or hundreds of features, a systematic approach to evaluating each possible output becomes impractical.

A partial solution to the curse of dimensionality is to restrict the range for each feature value so that the number of rule atoms per feature remains small, and is a viable approach when only the local properties of the Random Forest model are of interest. Alternatively, the rule atom framework can be used to develop heuristic methods. For example, we are able to discover adversarial examples by performing a random walk over rule atoms, as described in later sections.

Algorithm 1 Generate a list of all basis states for a Random Forest classifier by decomposing each child decision tree into constituent rules.

```
procedure GENERATE BASIS STATES(random forest)
    list $basisstates = \emptyset$
    for all $tree$ in random forest do
        for all $rule$ in $tree$ do
            for all $ruleatom$ in $rule$ do
                list $basisstates$ ← list of $basisstates \bigcup ruleatom$
    sort list of $basisstates$
    return list of $basisstates$
```

20.3 Emergent Interactions and the Limitations of Learning with Random Forests

Random Forests can learn *emergent interactions* that have greater complexity than any of the individual decision trees. Curiously, Random Forests can learn high-order interactions between features, even if the individual decisions trees are shallow, but are unable to capture XOR-type interactions. This insight could assist data analysts in identifying situations in which Random Forests is likely to be of use, or conversely to design data sets that are resistant to analysis by decision trees and Random Forests for benchmarking or adversarial purposes.

To demonstrate the ability for Random Forests to learn high-order interactions, we consider a sample space with n binary features (0 or 1) and a class that can be -1 or +1. Every decision tree has either no splits or a single split, which inhibits their ability to learn any second-order or higher interactions. However, Random Forests is still able to learn an n-order AND-type interaction, as we show below.

We consider a Random Forest model comprising a total of $2n$+1 decision trees. The first decision tree outputs +1 if and only if the first feature is one, the second tree outputs +1 if and only if the second feature is one, and so forth, which accounts for the first n trees. The following $n - m - 1$ trees output -1, regardless of the input, and the remaining m trees always output +1. As $2n + 1$ is odd, there will always be a strict majority and tied votes do not occur. m controls the order of the interaction.

If m=0, then the Random Forest model will output +1 if and only if all the features are one, thus capturing an n-th order AND-type interaction. If $m = n - 1$, then the Random Forest model becomes an (inclusive) OR-type interaction. For intermediate values of m, the Random Forest model will produce +1 if and only if $n - m$ or more features have a positive value. This shows that even shallow decision trees with a single split are able to learn high-order interactions and generalizes the interaction effects seen in Fig. 20.1.

Despite the ability of Random Forests to learn some high-order feature interactions, it is unable to induce any higher order XOR-type interactions than those provided by the individual decision trees. We once again consider n binary features, but now the decisions trees can have up to $n - 1$ splits, so they can learn an XOR in any $n - 1$ dimensional subspace, but cannot learn a n-dimensional XOR. We show that Random Forests cannot generate an n-dimensional XOR interaction by calculating the difference between the predicted classes of the "even" and "odd" corners of the hypercube, where even and odd refers to the parity of the sum of feature values.

The outputs of the even parity corners of an XOR hypercube are -1, while the outputs of the odd parity corners are +1. Since Random Forest models make their class predictions based on the majority vote, the corners of the hypercube may have a mixture of positive and negative class predictions from the decision trees, noting that the majority vote must be negative on the "even" corners and positive on the "odd" corners. This gives rise to the mathematical identity

$$\sum_{n} \Big(\sum_{x_i = odd} f_n(x_i) - \sum_{x_j = even} f_n(x_j) \Big) > 0, \tag{20.2}$$

where f_n refers to the n-th decision tree and x_i is a feature vector.

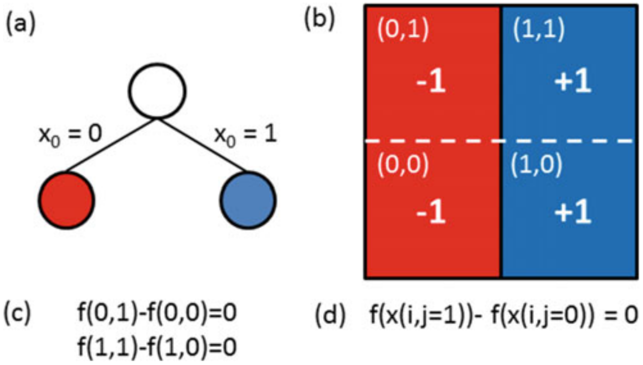


Fig. 20.2 Shallow decision trees have pairs of identical outputs on the unsplit features. **a** A shallow decision tree with binary features that splits only on the value of x_0. **b** The decision tree will produce identical outputs for $x_1 = 0$ and $x_1 = 1$ for each value of x_0 as there was no split involving x_1. **c** A symmetry relationship present for the decision tree shown in panel (**a**). **d** A general symmetry relationship present when the tree depth is less than the number of features. j is a feature that is not split on for a pair of outputs and i provides fixed values for the other features

A decision tree of depth $n - 1$ must generate pairs of identical classes for the unsplit feature. A simple example of a decision tree with a single split and two features is shown in Fig. 20.2. The decision tree has identical outputs for $x(i, 0)$ and $x(i, 1)$ for $i = 0$ and $i = 1$ since the second feature is not used in the split. The same phenomena will occur in deeper decision trees, although the feature(s) not used in the classification will depend on the path taken through the decision tree. Since the pairing always involves one odd and one even corner, this implies

$$\sum_{x_i=odd} f_n(x_i) - \sum_{x_j=even} f_n(x_j) = 0 \tag{20.3}$$

for any decision tree with a depth less than the number of features. Since we only consider decision trees with fewer splits per path than features, there is a contradiction between Eqs. 20.2 and 20.3 that demonstrates that Random Forests cannot induce XOR-type interactions.

20.4 Inverse Decision Trees

Random Forest models accept a vector input and deterministically generate a prediction or decision. This process can be inverted, so that a model is given a potential prediction or decision and every possible feature vector that could generate such an input is provided. We call this an *inverse decision tree*. To the best of our knowledge no method for generating inverse decision trees is described in the literature, although the process can easily be accomplished by hand for small decision trees.

Inverse decision trees generalize the decision boundaries we derived in the previous section. Decision boundaries determine the local effect of perturbations on a single output, while inverse decision trees reproduce the global structure of a Random Forest model. Inverse decision trees are useful for the purpose of interpreting the model structure and identifying when a desired outcome will be generated.

Inverse decision trees can be constructed from an expanded decision tree. The domain of plausible inputs can be generated by fixing the desired output classification and tracing through each decision path for the expanded decision tree (like the one shown in Fig. 20.1), which encapsulates every decision rule and output. As before, multiple tree structures can be generated for a single model so the decision rules, while logically equivalent, may have different topological structures.

20.5 Comparison of Two Random Forest Models

The expanded basis set approach can be used to compare two or more Random Forest classifiers. The basis sets from both Random Forest models can be combined into an even larger expanded basis set. Following earlier reasoning, the combined basis set contains every possible decision rule for both classifiers. The classifiers can be tested for each combination of features. If the classifiers agree at every point, they are predictively identical under all circumstances. Otherwise, a subset of regions that produce different outputs can be identified for further analysis.

20.6 Parallel Coordinates and Visualization

Parallel coordinates enable visualization in more than three-dimensions and are especially useful for understanding the global structure of a data set. A parallel coordinate view of the original *iris data* (extracted from Scikit-learn [8]) is shown in Fig. 20.3. Each colour represents a different species and the features have been linearly rescaled so the minimum value for each feature is zero, and the maximum value is one. From visual inspection, it can be seen that the setosa (blue) class can easily be separated from the versicolor (red) and virginica (black) classes using any of the features, with the final two features being particularly sensitive to the iris species. Parallel coordinates provide a global view of the classifier decision structure.

A second set of parallel coordinates are shown in Fig. 20.3 to represent the data set from the viewpoint of the Random Forest classifier. Each feature has been rescaled to a value of k/n, where k is the number of basis states it exceeds and n is the total number of basis states for that feature. Each feature has a support of [0,1], except in the special case of no nodes when the features are all set to a value of $\frac{1}{2}$. The vertical dots represent the decision boundaries and serve as a guide to the eye.

The rescaled parallel coordinates are useful because they allow us to analyze how the Random Forest classifier is processing multi-dimensional data. We see similar behaviour to the linear scaling case, namely that the last two features clearly provide strong information for classification, while the first two features are less informative. Interestingly, we can see the last two features have a lot of bins, but that some of the bins contain no data points and indicates that Random Forests is overfitting.

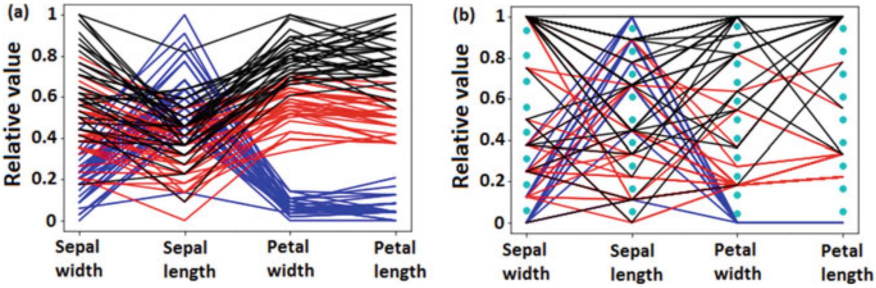


Fig. 20.3 The *iris* data set plotted in parallel coordinates (red is versicolor, blue is setosa, black is virginica). **a** The features have been independently scaled between zero and one. **b** The features have been rescaled such that their values indicate which Random Forest model bin the values fall into. The vertical dots represent the boundaries between adjacent feature bins for the Random Forest classifier. See text for details

We suspect the large number of empty bins is caused by the ambiguous boundary between the versicolor and setosa classes, which cause each of the decision trees to generate multiple decision boundaries.

20.7 Bounds for Stability Analysis

Random Forest classifiers, like decision trees, consist of a number of linear boundaries that demarcate the possible outputs. It is helpful in many situations to be able to explicitly identify the decision boundaries. For instance, a Random Forest classifier may be used to determine the outcome of an applicant's home loan application. Some of the applicant's characteristics may be unknown or subject to variation, such as level of income or living expenses. An unsuccessful application could use the decision boundaries to identify changes that could be used to improve the strength of a future application, or by the creditor to ensure small variation in applicant's status will not materially impact on the chance of a default.

Decision boundaries can, in theory, be identified in a model-agnostic way by generating a grid of points and visually plotting the various model outputs. However, this approach suffers from the curse of dimensionality and may not resolve genuine decision boundaries if the grid is too coarse. An analytic or semi-analytic procedure for identifying decision boundaries is preferable because of gains in computational efficiency and the ability to guarantee identifying any boundaries of interest.

The expanded basis set approach provides an analytic pathway for generating regions in which the classifier output will be stable, depending on whether a single feature or multiple features are perturbed. The key insight needed to generate these bounds is that the classifier output will always remain constant within a hyper-rectangle formed between adjacent basis states. It is possible to identify exact decision boundaries by sequentially sampling points from adjacent hyper-rectangles.

A lower bound for output stability when all features are perturbed can be calculated by taking the two adjacent basis states for each feature, as shown by the outwards pointing arrows in Fig. 20.4. A large minimum bounding box can provide confidence that a prediction or recommendation is robust against mild uncertainty or variability, while a small minimum bounding box is inconclusive, and could be the result of genuine nearby decision boundaries or because of a large number of basis states. The minimum bounding box is non-optimal in the sense that there may exist a larger rectangular region in which the Random Forest model will output a constant value.

In contrast, we are able to establish optimal upper bounds for the case in which only one feature is perturbed. It is calculated by sequentially moving through the basis states of a feature until the classifier output changes, or the list of basis states are exhausted. In practice, the bounds are calculated by observing the classifier output for the midpoint between adjacent basis states and then rounding towards the original value for reasons of numerical stability. The maximum bounding box is optimal in the sense that it provides the exact decision boundaries for each individual feature. The maximum bounding box can be useful in identifying a subset of important variables or providing insight into the local interactions that exist between features. A small

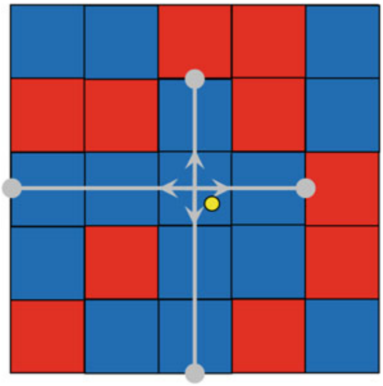


Fig. 20.4 Given a single data point (yellow dot), we can calculate local stability bounds for the classifier output. Optimal bounds for the maximum possible single feature perturbation are calculated by sequentially stepping through adjacent basis states (grey circles) and a lower bound for the stability of a perturbation of an arbitrary number of features is calculated by finding the hyper-rectangle of all adjacent basis states (outwards facing arrows)

maximum bounding box indicates that a particular prediction is unstable, while a large maximum bounding box shows that the output is robust against changes in any single output, but provides no guarantees for multi-feature perturbations.

20.8 Anchors and Counterfactuals

Anchors [7] and counterfactuals [4] play an important role in explainable artificial intelligence. Anchors are sets of features or rules that make model predictions locally independent of other features, while a counterfactual is a feature vector that would have (counterfactually) resulted in the model making a different prediction.

A duality exists between anchors and counterfactuals: If an anchor consists of a features from a possible set of n features, then there must exist a counterfactual that contains at least one feature of the anchor, and it must contain no more than $n - a + 1$ perturbed features. This follows trivially from the observation that if a counterfactual couldn't be generated containing one of the anchor's features, then the anchor would not be the smallest possible set of features that guarantees the original output.

We present algorithms for generating anchors and counterfactuals for Random Forests. We use an L_0 loss function that measures the number of altered features. We choose this loss function because Random Forests produces models that are invariant under monotone transforms. Our algorithms for anchors and counterfactuals are systematic and exhaustive as they find all of the solutions that satisfy our criteria, and can easily be adapted for other loss functions.

Counterfactuals are generated using a forward selection approach. Each individual feature is individually perturbed through all their possible values, and if no change in output is observed for any of the perturbations, then this is repeated for all pairs

Table 20.1 Example points, anchors and counterfactuals for the iris data set with a Random Forest classifier. The * in the anchors indicate values that can be changed without affecting the final classifier output

Point	Anchors	Counterfactuals
(6.1, 2.8, 4.7, 1.2)	(*, *, 4.7, 1.2)	(6.1, 2.8, 4.9, 1.2)
(5.7, 3.8, 1.7, 0.3)	(*, *, 1.7, 0.3) (*, 3.8, *, 0.3)	(5.7, 3.8, 1.7, 0.7)
(7.7, 2.6, 6.9, 2.3)	(*, *, 6.9, 2.3) (7.7, *, *, 2.3)	(7.7, 2.6, 6.9, 0.7)
(6.0, 2.9, 4.5, 1.5)	(*, *, 4.5, 1.5)	(6.0, 2.9, 4.9, 1.5)
(6.8, 2.8, 4.8, 1.4)	(*, 2.8, 4.8, 1.4) (6.8, *, 4.8, 1.4)	(6.8, 2.8, 4.8, 1.6) (6.8, 2.8, 5.0, 1.4)

of features, then all triplets and so forth until a minimum set of features is identified that is able to change the model's output. The set of minimum length counterfactuals consists of all combinations of features that can change the value of the output, given no counterfactual with fewer perturbed features exists.

To generate anchors we follow the opposite approach (Algorithm 2). We initially perturb all but one of the features, gradually stepping down the number of perturbed features until no change in the Random Forest model's output can be produced. Prototype anchors and counterfactuals for iris data set are shown in Table 20.1.

Algorithm 2 Generate all anchors of minimum possible length for a Random Forest model given a local data point.

```
procedure GENERATE LIST OF ANCHORS(random forest, point)
    int $n \leftarrow$ length($point$) – 1
    $anchors \leftarrow$ set of all features $F$
    while $anchors \neq \emptyset$ and $n > 0$ do
        $anchorsold \leftarrow anchors$
        $anchors \leftarrow \emptyset$
        for all Set of features $S$ of length $n$ do
            boolean $stores$
            for all features $F$ not in $S$ do
                for all possible values for $F$ do
                    $newpoint \leftarrow point$
                    $newpoint$[features in $F$] $\leftarrow newvalue$
                    if random forest $(newpoint) \neq$ random forest $(point)$ then
                        $stores \leftarrow$ false
            if $stores$ then
                anchors $\leftarrow$ anchors $\bigcup$ $S$
        $n \leftarrow n - 1$
    return $anchorsold$
```

20.9 Adversarial Examples

Adversarial examples are carefully crafted data points that are able to confuse trained classifiers, despite in many cases being imperceptibly different to correctly classified examples. Adversarial examples have received notable attention in the image recognition domain, where classifiers that attain human-level performance for natural images confidently misclassify adversarial examples [10], and in the cybersecurity domain in which malicious actors may attempt to disrupt or counteract the behaviour of automated systems [11].

The literature on adversarial examples has traditionally focused on gradient-based attacks against deep neural networks [10], although attack methods against support-vector machines [12], logistic regression [13] and linear regression [13] models have also been investigated. The crafting of adversarial examples for Random Forests and decision trees has received comparatively little attention [12]. Only a single method for specifically constructing adversarial examples against decision trees has been reported [12]. None have been reported specifically for Random Forests. Below, we adapt the method for constructing adversarial examples for decision trees to fool Random Forest classifiers using the expanded basis set.

Starting with the original data point, a feature is randomly selected and perturbed into an adjacent basis state until the input becomes misclassified (see Algorithm 3).

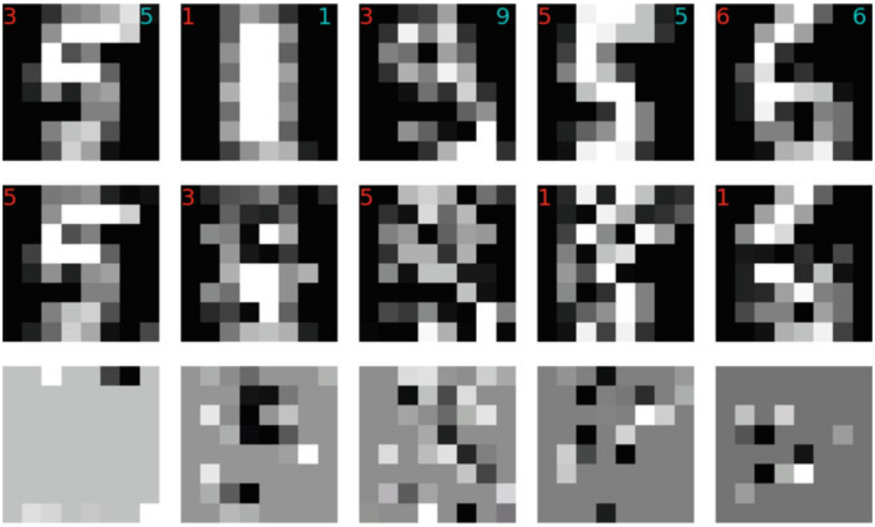


Fig. 20.5 Adversarial examples generated against a Random Forest model trained on the *digits* data set (from Scikit-learn [8]). The original samples are shown along the top row, the adversarial examples are shown in the second row, and the third row shows the perturbation alone. The classification of the image by the Random Forest model is given in red in the upper-left-hand corner for the first and seconds rows, and the ground truth is given in cyan in the upper-right-hand corner in the first row only

We can repeat this process several times, selecting the example with the smallest L_0 loss. This procedure may not discover the optimal counter-examples, like our counter-factual algorithm, but scales more efficiently for large numbers of features.

Adversarial examples for digit recognition are shown in Fig. 20.5. Samples are drawn from the *digits* data set from *Scikit-learn* [8]. The top row shows the original images with their classification by the Random Forest model in red in the upper-left-hand corner and the ground truth in cyan in the upper-right-hand corner of each image. The second row shows a perturbed sample with a changed classification and the third row shows just the perturbation. We can see that the first and last columns have adversarial examples that are close to the originals, while the middle three columns have significant perturbations. The first row is particularly interesting. The Random Forest model originally misclassifies the "5" as a "3", while the adversarial example is assigned the true classification, despite only pixels along the top and bottom of the image being perturbed. This suggests that the Random Forest model has not properly learnt the underlying concepts, which is unsurprising given the small training set of only 100 images. In many cases, the perturbations applied to the image are quite large, which we conjecture is because of the sub-optimal adversarial image finding algorithm, although further work is needed.

As a general trend, we found that adversarial examples tended to perturb pixels close to the centre of the image and perturbed several pixels by a small amount. Our algorithm never found an adversarial example consisting of only a single perturbed pixel, although our search was not exhaustive. This suggests that Random Forests is learning rules consisting of many low-order interactions.

Algorithm 3 Generate an adversarial example against a Random Forest image classifier, given an example image that is classified correctly.

```
procedure GENERATE ADVERSARIAL IMAGE(random forest, $point$)
    $adversarialpoint \leftarrow point$
    while random forest($point$) == random forest($adversarialpoint$) do
        randomly select a feature $f$
        $adversarialpoint$[feature $f$] $\leftarrow$ adjacent basis state value
    return $adversarialpoint$
```

20.10 Conclusion

Random Forests is considered one of the best-off-the-shelf data mining algorithms. However, it suffers from poor interpretability and an opaque decision structure. We have developed a procedure to generate an expanded basis set for a Random Forest model by aggregating the decision rules from all of the constituent decision trees. The expanded basis set allows the Random Forest model to be easily algebraically manipulated, in contrast to the conventional representation that comprises potentially hundreds or thousands of decision trees. The expanded basis set facilities a number of additional operations that were previously infeasible: Inverse mapping of outputs to the domain of potential inputs, comparison of two Random Forest models and generation of adversarial examples. These operations improve the utility of Random Forests and make it more interpretable and transparent for a range of applications.

The global behaviour of a Random Forest classifier and the accompanying data set can be effectively visualized by combining parallel coordinates with a non-linear rank transform that bins feature values using the expanded basis set. Using this approach on the *iris* data set we were able to identify commonalities and differences between the three *iris* species, in addition to observing areas of underfitting and overfitting by Random Forests. For other problem sets, global visualization could facilitate comparison of the classifier's rule structure with domain expertise.

The expanded basis set can be used to evaluate the local and global stability of a Random Forest model. Systematic procedures were developed for creating counterfactuals, local perturbations of features that cause the model output to change, as well as anchors, combinations of features that ensure the model output remains constant when other features are perturbed. These procedures were developed specifically for a Random Forest classifier and can systematically discover all relevant anchors and counterfactuals. The procedures may not scale well for high-dimensional models, so we developed alternative procedures for identifying model stability bounds when a single feature is perturbed and when every feature is perturbed that can easily scale to hundreds or thousands of features. These methods can inform analysts about the stability of a Random Forest model's output, and are particularly valuable in situations where some of the features may be uncertain or vary over time. In the future, the expanded basis set approach could be combined with algorithms like RuleFit to create off-the-shelf data mining algorithms that are accurate, simple to use and interpretable.

Acknowledgements We thank Maria Athanassenas, Tim Dell, David Marlow and Timothy Surendonk for feedback and suggestions.

References

1. Friedman J, Hastie T, Tibshirani R (2001) The elements of statistical learning, vol 1. Springer series in statistics New York, NY, USA
2. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat, pp 1189–1232
3. Goldstein Alex, Kapelner Adam, Bleich Justin, Pitkin Emil (2015) Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. J Computat Graph Stat 24(1):44–65
4. Ribeiro M, Sameer S, Carlos G (2016) Model-agnostic interpretability of machine learning
5. Welling SH, Refsgaard HF, Brockhoff PB, Clemmensen LH (2016) Forest floor visualizations of random forests. arXiv:1605.09196
6. Friedman Jerome H, Popescu Bogdan E et al (2008) Predictive learning via rule ensembles. Ann Appl Stat 2(3):916–954

7. Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence*, 2018
8. Pedregosa Fabian, Varoquaux Gaël, Gramfort Alexandre, Michel Vincent, Thirion Bertrand, Grisel Olivier, Blondel Mathieu, Prettenhofer Peter, Weiss Ron, Dubourg Vincent, Vanderplas Jake, Passos Alexandre, Cournapeau David, Brucher Matthieu, Perrot Matthieu, Duchesnay Édouard (2011) Scikit-learn: Machine learning in python. J. Mach. Learn. Res. 12:2825–2830
9. Thomas G. Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000
10. Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016
11. Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. Adversarial examples for malware detection. In *European Symposium on Research in Computer Security*, pages 62–79. Springer, 2017
12. Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint* arXiv:1605.07277, 2016
13. Goodfellow IJ, Jonathon S, Christian S (2015) Explaining and harnessing adversarial examples. In: Proceedings of the 2015 International Conference on Learning Representations

Chapter 21
Towards the Identification and Visualization of Causal Events to Support the Analysis of Closed-Loop Combat Simulations

Dion Grieger, Martin Wong, Marco Tamassia, Luis Torres, Antonio Giardina, Rajesh Vasa, and Kon Mouzakis

Abstract Analysis of output data from closed-loop combat simulations can provide insights into the relationships between model inputs and outputs. However, analyses that only consider end-of-run output data may not be sufficient to explain why those relationships exist. In the case of stochastic models, where multiple replications of the same scenario are conducted, the presence of outliers and multi-modal results also needs to be accounted for. In this paper, we use a military case study to explore a range of techniques to interrogate the intra-run event data (i.e. trace logs) generated by a combat simulation in order to help address these two issues. Cumulative event plots and geo-spatial visualization techniques which also incorporate the temporal aspects of the simulation appear best suited to explain the presence of outlier replications and multi-modal results. Exploratory work using hierarchical clustering techniques and temporal decision trees provide a promising step towards better explaining causal events within the combat simulation data.

Keywords Combat simulation · Clustering · Key event identification

21.1 Introduction

Closed-loop combat simulations can be employed to provide insights into differences between the operational effectiveness of a set of military alternatives. These alternatives may be physical in nature, such as different equipment, platforms or terrain conditions, or behavioural, such as tactics, techniques or procedures. Analysis of these closed-loop models is typically conducted using end-of-run output data [7].

D. Grieger (✉) · M. Wong
Joint and Operations Analysis Division, Defence Science and Technology Group, Edinburgh, Australia
e-mail: dion.grieger@dst.defence.gov.au

M. Tamassia · L. Torres · A. Giardina · R. Vasa · K. Mouzakis
Applied Artificial Intelligence Institute (A2I2), Deakin University, Geelong, Australia
e-mail: antonio.giardina@deakin.edu.au

A. T. Ernst et al. (eds.), *Data and Decision Sciences in Action 2*,
Lecture Notes in Management and Industrial Engineering,
https://doi.org/10.1007/978-3-030-60135-5_21

A limitation of this approach is the inability to explain the reason for a particular finding in the end-of-run data.

There are two primary challenges in this space for the simulation data analyst. The first is to identify and explain the presence of outliers within a set of replications and determine if those outliers are genuine results, an artefact of the model or potentially invalid replications that should be removed prior to analysis. The second is to explain the key differences between groups of replications, either different groups within the same scenario option or groups of replications from different scenario options. Ideally, approaches to address these issues should be automated in order to avoid the need for analysts to manually view multiple simulation replays. This paper presents research conducted through collaboration between DST Group and Deakin University to address these challenges. We use a case study to provide context for alternative analytical techniques and also propose some alternative methods to further explore this problem in the future.

For stochastic models, multiple replications are used to enable statistical comparisons between alternatives to be made. Inherent to stochastic models is the chance that some replications will produce results that are far from the mean or median of all replications. Such outlier replications may contain vital information in our analysis of combat simulation runs because, while improbable, their occurrence may drastically affect the findings and require further investigation. While methods to detect outliers already exist [5, 16], the analysis of intra-run data (i.e. trace logs) within a simulated replication may hold the key to explaining such outliers. Previous research on the analysis of military simulation intra-run data is limited, although there is a growing interest by researchers in using statistical modelling on intra-run data to find hidden insights. Luotsinen and Bölöni [19] used Hidden Markov Models (HMMs) to annotate the actions of moving agents, with the aim of recognizing key behaviours in military operations. They were able to achieve high accuracy using noisy data and claim that the annotation process can be performed in real time. Acay et al. [1] discussed the suitability of Dynamic Bayesian Networks (DBNs) for causal analysis in military simulations, presenting theoretical arguments on why DBNs are a better choice over HMMs at achieving a higher efficiency. In particular, they argued that DBNs have equal expressive power as HMMs but have better computational complexity on average while equal to HMMs only in the worst case. O'May et al. [21] used simulation intra-run data at three different stages of the simulation to train decision trees in order to predict the simulation outcome achieving high accuracy even though they do not analyze the predictive power of the various features used.

Team-based sports share many similarities to military battles in the sense that they are both team invasion games characterized by the presence of sophisticated attack and defence tactics. Markov models are also a common modelling approach to team-based sport matches where states represent possession of the ball by a team on a particular portion of the field. Hirotsu and Wright [12] modelled games using an HMM with four states while Forbes [22] used an HMM with 18 states, improving its performance by using the additional information in the model. The transition probabilities learned by these models were then used to compare the different playing styles across teams. Positional data is also widely used in the analysis of team-based sports. O'Shaughnessy [22] created heat maps of the AFL field to evaluate the value of possession conditioned upon ball position and team possession using intra-match data. Cervone et al. [6] proposed "Expected Possession Value" (EPV) models in basketball, which also uses intra-match data, such as the position of the players and the ball, to form a description of the ball-player interactions. This enabled the identification of situations where players are well positioned to score off a basketball rebound. Similarly, Le et al. [18] proposed an advanced approach, called "Data-Driven Ghosting" which uses a type of neural network called long short-term memory (LSTM) to predict where the defensive players should have been as opposed to where the players actually were in order to identify players out of formation.

Research on analyzing intra-run data has also been found in the video game scene particularly on Multiplayer Online Battle Arena (MOBA) games.[1] In MOBAs, two teams of "heroes" controlled by human players (usually five per side) must collaborate to invade and take control of the opponents' home base. Drachen et al. [10] used intra-run data to analyze how players of different skill levels behave during a game when playing Defense of the Ancients 2 (DotA2).[2] Using spatio-temporal distance measures and the identification of key early player decisions, they were able to highlight differences in the behaviours between expert and novice players. Yang et al. [28] used player prior statistics and intra-run data from a large number of DotA2 games to predict the outcome of the games, achieving high accuracy. They observed that the predictive power of prior statistical features will drop as the match lasts longer, while intra-run data becomes progressively more useful for the prediction. Finally, work still under review by Demediuk et al. [9] looks at DotA2 using intra-run games data (i.e. movement, game stats and abilities used) to understand the patterns of how heroes can be used differently in order to complement the team.

The literature demonstrates that intra-run data has been analyzed in both the sport and video game fields with some level of detail, while it has only been marginally explored in the military context. This is in part due to the difference between data produced by military simulations and the data recorded from sport and video game matches. In particular, sport and video game data tend to include a limited number of entities and can significantly vary between matches due to human involvement. On the other hand, military simulation data involves a large number of entities, but this added complexity is mitigated by the algorithmic nature of the simulation events that take place which, while stochastic, tend to produce recognizable patterns across repetitions.

The work presented in this paper is a first step towards the complete analysis of intra-run data in the context of military simulations and presents an initial set of approaches an analyst can use to gain insights on the evolution of a simulation. These techniques are not limited to the military context but can also be applied to other simulation models that are stochastic and event-driven in nature, such as emergency evacuation simulations. The study of how the presented methods can be used in other

[1] https://en.wikipedia.org/wiki/Multiplayer_online_battle_arena.

[2] www.dota2.com.

simulation contexts is not an aim of this work, but we hope this paper fosters further investigation in this direction.

21.2 Case Study Overview

A case study was used to help highlight the challenges associated with combat simulation data analysis, particularly the two specific areas of interest described earlier. The case study compared the performance of three military vehicle options, A, B and C, in a specific scenario and across a range of performance metrics.

The mission for the Blue force (the side for which the different vehicle options were being assessed) was the rapid seizure of an urban fringe area. The Blue force was a Combat Team (CT) consisting of three infantry platoons, a Tank troop and two direct fire support vehicles. The Red (enemy) force was approximately one third the size of Blue and its mission was to delay Blue by 24 h. Other key aspects of this scenario were

- The CT assembly area was located approx. 10 km from the objective within a heavily vegetated area;
- There was only one viable road-based approach to the objective;
- Red had some indirect fire support options and
- An ambush was planned by Red under the assumption that Blue would be approaching via the roads.

A number of metrics were used to provide an assessment of each option's lethality, protection, signature and knowledge acquisition capabilities. Protection was assessed by comparing casualty rates of infantry and various levels of vehicle damage. Similarly, the lethality-based metrics considered the same effects on enemy platforms and infantry. The signature metrics measured the total number of unique Blue force entities that were detected during the simulation and also the range at which each Platoon was first acquired. The knowledge metrics captured the same information in relation to acquisitions of Red entities made by the Blue force. In addition, a binary mission success metric was used to determine whether Blue achieved its mission. In this case, mission success was based on the successful seizure of its objective within 24 h and before reaching certain attrition thresholds to critical assets.

Subject matter experts (SMEs) were consulted in order to extract the appropriate plans and tactics for both the Blue and Red forces to undertake their respective missions. These plans were then encoded in the COMBAT XXI simulation developed by the United States Army [2]. COMBAT XXI is a closed-loop, stochastic, discrete event entity-based model focused on land-force tactical combat. COMBAT XXI is designed to model operations at brigade and below and is a high-resolution model that allows for detailed behaviours of units and individual entities. In this case study, the simulation was replicated 201 times for each of the three vehicle options in order to allow for statistical analysis of the results. Each replication terminated either when Blue successfully completed its mission or it reached one of the defined failure thresholds described earlier.

This particular scenario, and analytical question, was chosen for the case study as it is indicative of the types of studies usually conducted by DST Group using COMBAT XXI. Therefore, it is expected that the methods employed will be generalizable to future COMBAT XXI studies.

21.2.1 Case Study Insights

The analysis of the end-of-run data identified a number of questions that we would like to be able to further explore using methods that interrogate the simulation event data. The aim of this type of analysis is to help provide the context between model inputs and outputs and to explain key differences between replications or options in terms of important causal events within the simulation. For this case study, we used the following two questions as exemplars of the analytical questions we typically seek to address using these types of analyses:

1. There appear to be some outliers in the results for some metrics. In particular, replications 23, 39, 124 and 138 are far from the median result relative to other replications for a number of metrics for option A. An example is provided in Fig. 21.1 where the four replications identified earlier are those at the bottom left portion of the plot. Does an explanation exist within the simulation event data that could explain why these outliers are occurring for these particular metrics?

2. A statistical comparison of the three options was conducted for each metric. A discussion of all the results of these analyses is beyond the scope of this paper. Instead we will focus on the metric which considers the percentage of enemy (Red) infantry killed by Blue to illustrate the analytical challenge. The results were that option B has the highest mean $\mu = 0.368$, with C and A following in this order, having means of $\mu = 0.354$ and $\mu = 0.336$, respectively. To confirm this from a statistical standpoint, for each pair of options we run the Wilcoxon signed-rank test [27] with

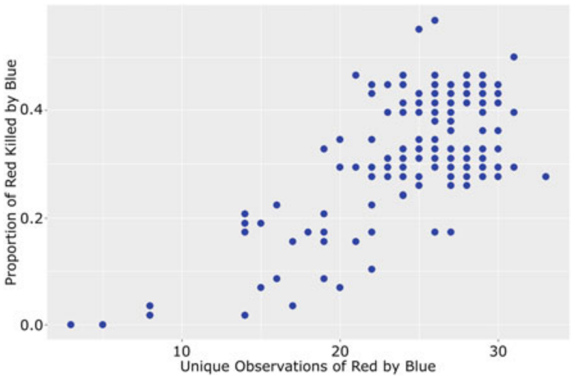


Fig. 21.1 Proportion of enemy killed by Blue vs. number of detections of enemy made by Blue for option A

Bonferroni correction [3] and calculate the Cohen's d coefficient [8], which gives an estimate of the "impact" of the difference between the two options. Our threshold for statistical significance is alpha = 0.05 (corrected alpha = 0.008). We chose a paired test because the same 201 random seeds were used for each of the three options, running on the same otherwise deterministic simulation software.

- The test between options A and B yields $p < 0.001$ ($W = 6419.5$), with the effect size being $d = -0.316$: this indicates a statistically significant, medium effect size in favour of B.
- The test between options B and C yields $p = 0.109$ ($W = 7435$), with the effect size being $d = +0.143$: this indicates a non-statistically significant, small effect size in favour of B.
- The test between options A and C yields $p = 0.064$ ($W = 8459$), with the effect size being $d = -0.171$: this indicates a non-statistically significant, small effect size in favour of C.

The statistical analysis confirms that B outperforms option A with a medium effect size. However, there is not enough evidence to confirm that B outperforms C, nor that C outperforms A: the small effect sizes found suggest that it is possible that there was no difference, as opposed to the cause being an underpowered study. Ideally, the analyst would like to be able to further explain these results, namely, why does option B perform statistically better than option A but not better than option C.

21.3 Case Study Methods and Analysis

In this section, we describe a number of methods to explore the simulation event data in order to assess their utility to provide further insights to the questions of interest, both in the context of the case study, and of typical simulation analysis tasks.

21.3.1 Exploratory Data Analysis

Prior to analyzing any simulation event data, an exploratory data analysis (EDA) is typically conducted to explore the end-of-run metrics. The purpose of the EDA is to look for any anomalies in the data that may warrant further investigation or to identify replications that may be invalid and should be excluded from statistical tests. The hypothesis is already established prior to the EDA, i.e. the options are all the same, and therefore there is little risk of data dredging. A range of scatter plots, histograms, box plots and their derivatives are used to explore different metrics and various groupings of the data. For the case study data, a violin plot that groups the data according to mission success provides some additional insights into the outliers seen within the Red infantry casualty results (Fig. 21.2). In this instance, the data is split using the mission success metric (metric code PA_S001), which represents

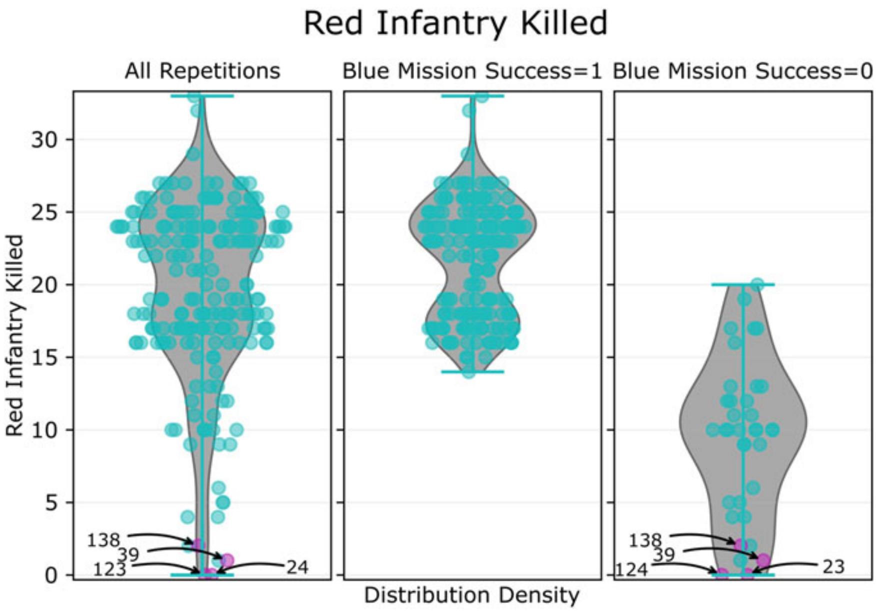


Fig. 21.2 Violin plot for the number of Red infantries killed by Blue for option A (y-axis) for all replications (left), Blue mission success (middle) and Blue mission failure (right)

whether the mission was a success (PA_S001 = 1) or failure (PA_S001 = 0) for Blue. A useful finding in relation to our questions of interest is that the outliers we are seeking to explain all lie within the group of replications in which Blue failed to achieve its mission. Violin plots comparing options A, B and C for the same metric (not shown) also provide some insight into the second question. Specifically, while the median results for options B and C are the same (39.7%) and larger than A (34.5%), the shape of the distribution of option B is actually more similar to that of option A. These results are useful for presenting hypotheses that provide a reason for why some replications are outliers and why we are seeing certain statistical differences between options. However, we are no closer to understanding the events within the simulation that actually caused these results. For this reason, we will now focus the analysis on events occurring during the simulation rather than only looking at end-of-run data.

21.3.2 Cumulative Plots

The proposed cumulative plot analysis techniques are based on the cumulative count over time of detection and engagement events. Engagements could potentially result in casualties to personnel and damage of various magnitudes to vehicles. The visualization of these variables as a function of time can provide insights about the

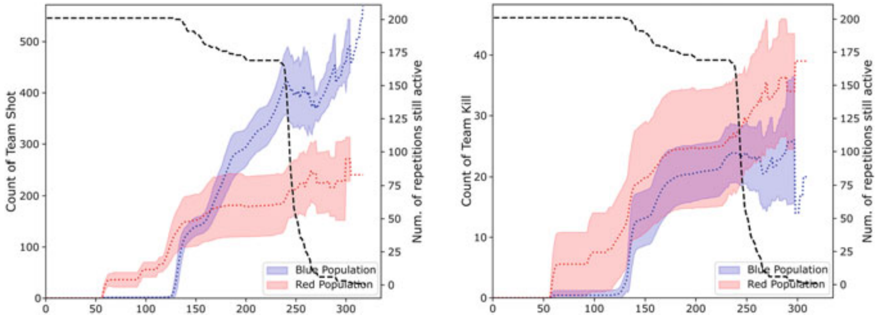


Fig. 21.3 Cumulative count of shots (left) and kills (right) for option A. Time (mins) is shown on the x-axis

progression of certain metrics during the missions. To generate these plots, a series of regularly spaced time-bins (typically 1 min) was created for every detection and engagement event within each replication. For every time-bin and replication, the number of events that occurred from the start of the mission until the end of the time-bin was counted. Mean values (dotted line) and standard deviations (shaded regions) were calculated across all replications (see Fig. 21.3). As each replication can finish at different times, the black dashed line was added to record how many replications were still active at each point in time (scale on the right Y-axis). As a result of some replications being removed from the calculations, the plots are not strictly monotonically increasing as might be expected in a cumulative plot. When replications terminate, they are dropped so that the statistics shown in the plots at successive times are representative only of the replications still active at those times. If these were not done, the mean would be lower due to the presence of smaller numbers in the sample, belonging to replications that did not have the "time" to increase those numbers. A replication is terminated either if Blue achieves its mission and reaches a certain point on the map before the 24 h mark, or if Blue exceeds some pre-determined casualty rates for one of its units.

There are two significant points of interest in these plots for option A (Fig. 21.3). The first is that there is a sudden rise in the number of shots and kills for both sides between the 130 and 140-minute mark. This suggests that there may be a significant set of engagements in that time period. The second point of interest is that despite Blue having a fairly constant and continuous rates of fire between the 140 and 250-minute mark, there appears to be a decreasing return on those engagements in terms of the number of kills inflicted on Red. The synthesis of these two findings suggests that the events occurring between the 130 and 140-minute mark should be subject to further investigation given the effect they appear to have on the remainder of the simulation. Note that the mean and standard deviation values beyond 250-minute mark are quite volatile as a result of these being very few replications which extend beyond that time point.

We also explored alternative approaches to visualize and group these cumulative plots. To provide an example, we again use the binary success measure to compare both the difference between successful and unsuccessful missions and also the difference between Red and Blue within those groups (Fig. 21.4). We clearly see a decisive point around the 130 to 140-minute mark in which Blue is not detected by Red nearly as much during successful missions as it is when the mission fails (bottom right of Fig. 21.4). There are some second-order effects to be considered when examining detection data, such as the effect that casualties have on the number of entities available to be detected. Nonetheless, the synthesis of the analyses presented so far indicates a point in time within the simulation that is of significant interest.

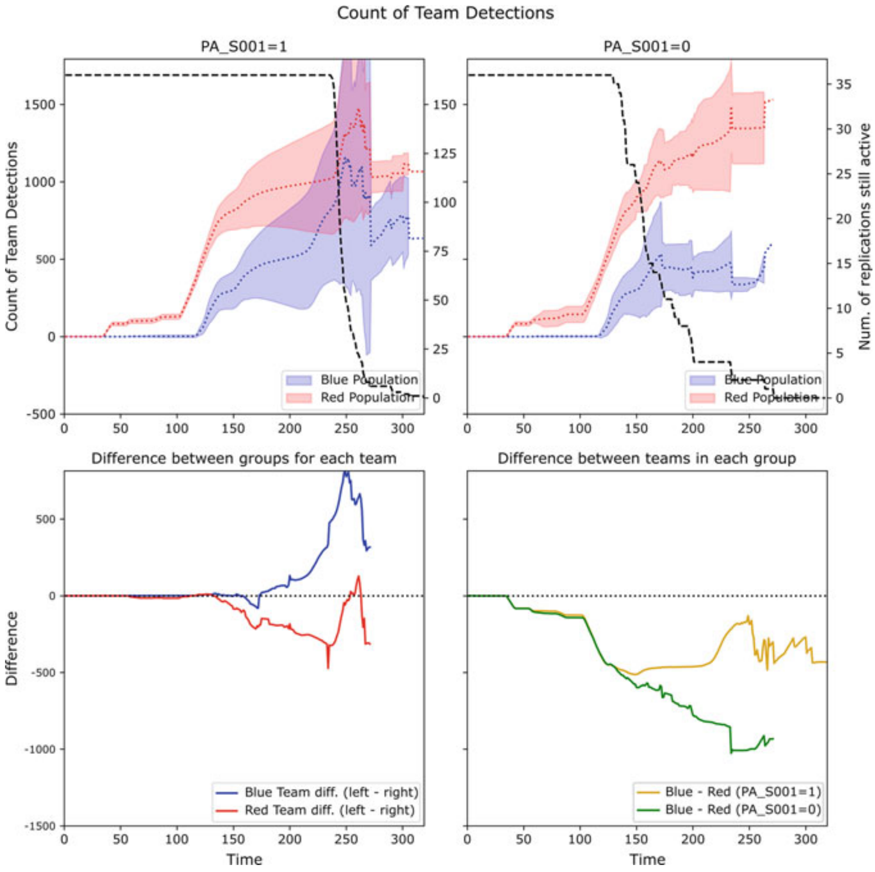


Fig. 21.4 Cumulative count of detections for mission success (top left) and failure (top right) and differences between detection counts of same side (bottom left) and between opposing sides (bottom right)

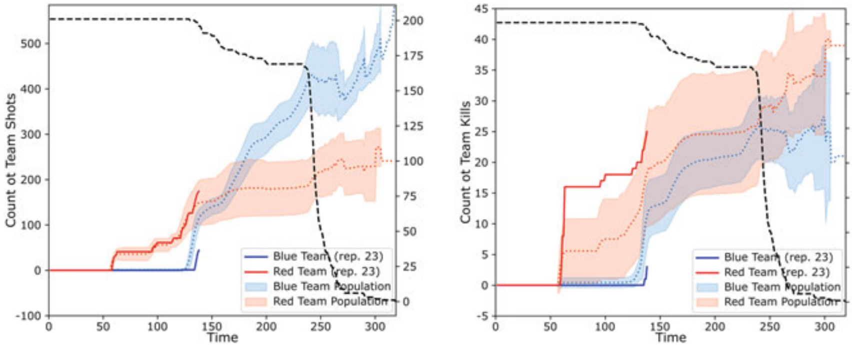


Fig. 21.5 Cumulative count of shots and kills for replication 23 shown against population trends for option A

21.3.2.1 Replication Versus Population

One of the primary objectives of this paper is to understand why certain replications appear different compared to the majority. To this end, it can be useful to compare the cumulative count of events of a replication against the mean trend of the population. Figure 21.5 shows the evolution over time of these cumulative counts for replication 23, one of the nominated outliers in option A of the case study data. The first noticeable feature is that replication 23 is shorter than the majority of all other replications. Interestingly, it terminates around the 140-minute mark, which is within the time period identified previously as being “of interest”.

The other salient point about the cumulative data for replication 23 is that there is a sudden increase in kills by Red beyond the standard deviation range around the 60-minute mark. There does not appear to be a similar increase in the number of shots taken which suggests that some low probability kills may have occurred. A hypothesis is that the early damage suffered by Blue may have had a subsequent effect in the major engagement that takes place around the 140-minute mark and caused the Blue attrition threshold to be met and, in turn, terminate the replication.

21.3.3 Geo-spatial Plots

The analysis presented thus far has identified some time periods of interest but has not yet been able to link the results to the context of the military scenario or mission. We now present the utility of using geo-spatial plots to further analyze intra-run data and to put geographical context around the time periods of interest. For the particular example relevant to the case study, we computed the average latitude and longitude of the target of all engagement events within the time period from 130 to 140-minute mark (Fig. 21.6). The replications tend to cluster into two geo-spatial groups, as far as average location is concerned. These figures, in fact, show a large group of

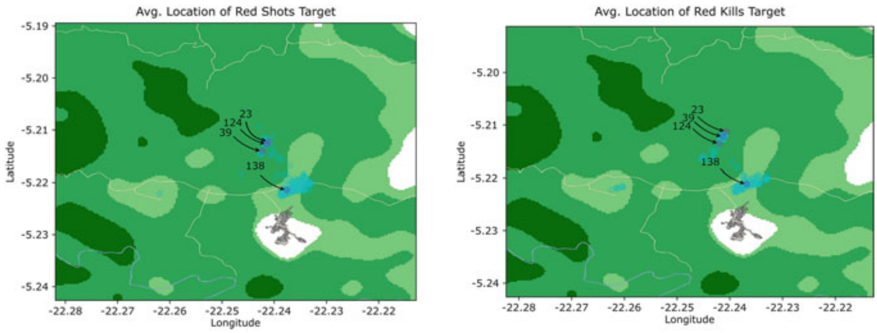


Fig. 21.6 Red targets average location for the shot and kill events for each replication in option A for the time period between the 130 and 140-minute mark

replications near the urban area (the dark spot in the map) and a smaller group North-West from there which contains three of the four outliers identified in the end-of-run data, including replication 23 which we have examined in detail already. Given that Blue is moving from North-West to South-East, there is an implication here that the approach of Blue, in those replications grouped with number 23, has been delayed for some reason and that they are being engaged, on average, from further out.

21.4 Other Methods Explored

Using the methods described in Sect. 3, we can tell if a particular replication falls outside of the normal bounds in terms of the number of detection and engagement events and also identify potential temporal and spatial points of interest within a scenario. However, we require methods that are able to further assist with explanatory theories on the reasons that gave rise to the results observed, preferably using an automated approach. As future events of a simulation are affected by the outcomes of previous events, there may be key events or sequences of events that determine how a simulation replication unfolds and the end-of-run metric that is achieved. It may be possible to detect these key events or sequences by comparing the event sequences in outliers to their more normative counterparts. This section describes proposed approaches to address the issue.

21.4.1 Hierarchical Clustering for Similarity Analysis and Key Event Identification

The hierarchical clustering approach produces clusters by either breaking down or aggregating items based on their similarity measures. The end result is a tree of clusters where similar items form sub-trees in the hierarchy. The hierarchical clustering analysis technique adopted in this research uses the agglomerative strategy with a complete linkage criterion, using either the Hamming or Euclidean distance depending on the nature of the data [26]. The raw event data was processed in a number of ways to produce features that would yield good results when used to generate the hierarchy:

1. Labelling events as detections (d), shots (s) or kills (k) for either the Blue (B) or Red (R) teams and using these as categorical features;
2. Using the pairwise Hamming distance between the event vectors calculated in [1] as features;
3. Time-binning the events and using the count of events for each type (e.g. b-d, b-s, r-k, etc.) as features;
4. Space-clustering and time-binning the events and proceeding as in [3];
5. Space-time clustering the events and proceeding as in [3].

These transformations allow the technique to show different aspects of the data. In our analysis the space-time clustered events produced the most useful results. An example of the result obtained is presented in Fig. 21.7. While the technique looks promising in identifying key groups of clusters based on intra-run events, a notable extension would look at whether the clusters formed by the dendrogram analysis can be contrasted in any meaningful way such that key events and sequences can be identified.

21.4.2 Space-Time Cubes for Visualizing Military Simulation Trace Data

In tactical analysis, location and time are crucial to understand the evolution of combat events in a single replication. To aid in the data analysis of such aspect, we adopted a visualization type, common in geo-spatial analysis, known as "space-time cube". Kraak provides an overview of the space-time cube and analyzes it from a geo-spatial visualization perspective [15]. The technique has been used to visualize accessibility in urban environments [20], movement differences among genders [17] and radiological hazard exposure [11]. A space-time cube facilitates the analysis of a time series of geo-spatial information, by showing both space and time data in a single visualization. This is achieved through a 3D plot where the two horizontal axes represent longitude and latitude, while the vertical axis represents time. In such way, the movement through space and time of an entity is represented as a rising line in 3D space. Different colours can be used for different lines in the same visualization to highlight groups of information, and a map of the terrain can be drawn as a plane at time 0. An example of a space-time cube using an alternative simulation data set (not the case study data) is shown in Fig. 21.8. In this scenario, there were eight red vehicles advancing towards six blue vehicles who were attempting to delay the advance. Long periods of static positioning can be clearly seen by lines that are almost

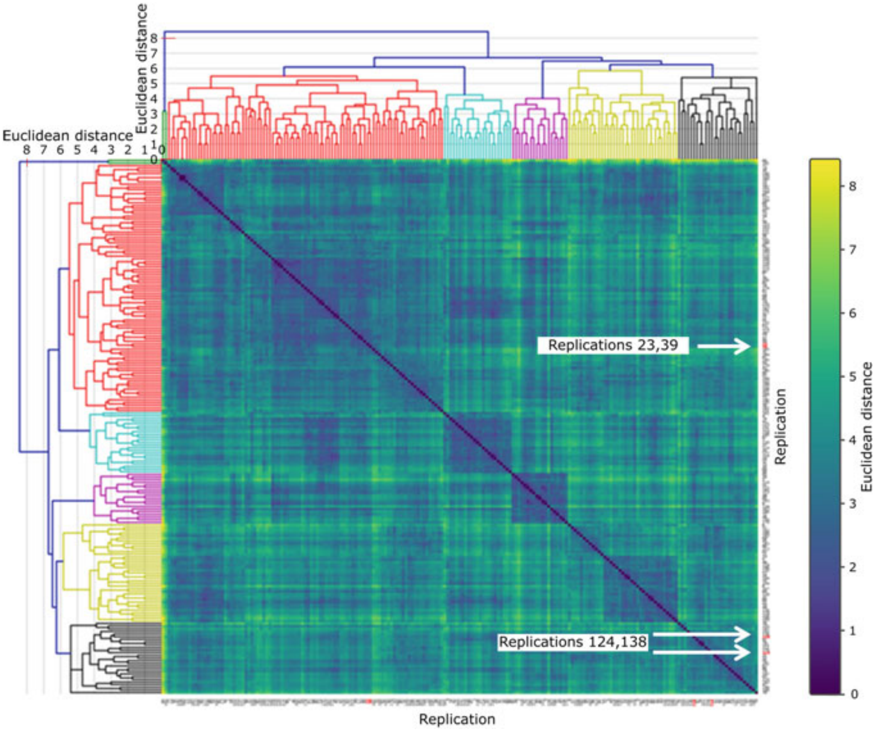


Fig. 21.7 Dendrogram and distance matrix of the result of the agglomerative clustering. The dendrogram (the tree-like structure) is repeated on both the X- and Y-axes and shows the "distance" between replications. The distance matrix is represented as a heatmap and shows the "distance" between every pair of replications. Note that the dendrogram analysis managed to cluster the case study outliers (replications 23, 39, 124 and 138) into neighbouring subgroups

vertical, while differences between the paths and rates of advance of the red vehicles are also visible. We plan to build upon the base space-time cube by introducing new visualization elements that can help differentiate the evolution of combat events between two replications.

21.4.3 Temporal Machine Learning Approaches for Key Event Identification

Another potential approach for identifying and explaining key events which we have explored is to apply various supervised learning algorithms to the simulation event data. The concept is to train several classifiers/regressors to predict some outcome metric (e.g. mission success, number of victims, etc.). Each predictor is trained on data from a certain time interval. It is expected that, as time progresses, there will

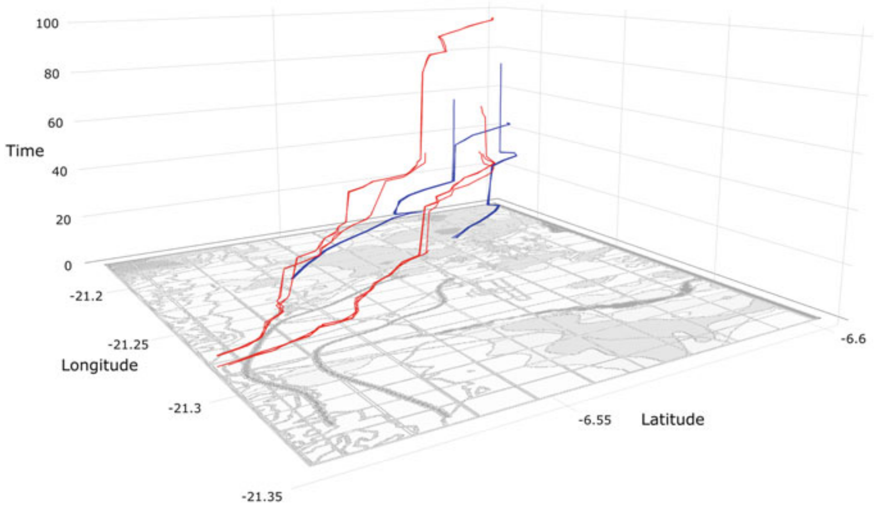
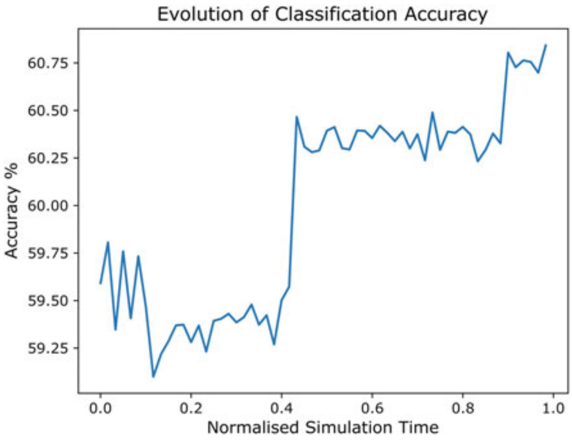


Fig. 21.8 A 3D visualization of movements of units over time; the two horizontal axes represent space, while the vertical axis represents time. Each line portrays the movements of a Red/Blue unit; the location of a unit through time is given by projecting the unit line on the terrain; the time of each movement is indicated by the vertical distance of the line from the terrain. The terrain is shown at time t = 0 for reference

be less room for changes in the evolution of combat. Therefore, it is reasonable to expect that the accuracy of later predictors will generally be higher than that of earlier predictors. Points in time when events occur that shift the balance of combat are likely to cause sudden changes in accuracy. Figure 21.9 shows an example of a temporal predictor using a decision tree [4] applied to the same alternative combat simulation data set as for the space-time cube. The improvement in accuracy is not linear in this example which implies that there might be some critical events occurring

Fig. 21.9 Classification accuracy as a function of normalized time using decision trees. Sudden accuracy increases denote the presence of significant events in time



within certain time periods that could help explain the outcome of the simulation. For example, the sudden increase in accuracy around simulation time t = 0.4 could point to the presence of an important event within the simulation.

Decision trees were initially chosen for this approach as they have the advantage of being more "explainable" compared to most other supervised learning methods. That is, if one wants to investigate the reasons why a decision tree produced a certain output, the answer is relatively easy to find; one would just need to descend the tree from the root and take note of which branches are traversed or analyze the differences between two successive trees. However, other approaches such as Random Forests and Artificial Neural Networks could also be applied here. Choosing the right feature space as inputs into these predictive models is also a critical aspect of this problem. One possible feature space to consider would be the space-time clusters used in the hierarchical clustering analysis in Sect. 4.1.

21.5 Future Work

Other methods that could be applied to this problem space but have not yet been investigated include Markov Chains and probabilistic graphical models (PGM).

A Markov Chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event [14]. It is possible to calculate transition probabilities between states from observed data in order to capture an approximation of the simulation behaviour in the form of a Markov Chain. As the states are logged by the simulation, the transition probabilities can be calculated empirically rather than inferred. Possible directions for such investigations include

- Verifying whether movements before a shot are strategically more likely to lead to a kill;
- Spatially segregating the data into individual models for each support-by-fire position, thus allowing analysis of individual sub-conflicts without the analysis being polluted by the concurrent actions of other troops acting elsewhere on the battlefield;
- Introducing time as part of the state, allowing the investigation of how behaviour changes over the course of the battle. This would allow detailed endgame analysis of time-critical situations to investigate whether a given strategy is rational.

PGMs are graphs where nodes (vertices) represent random variables while edges represent statistical dependencies between these random variables. PGMs can be of two types:

- Directed PGMs (edges represented as arrows and called Bayesian Networks) are widely used in ML and statistics, and are usually used to explain the causality relationship between variables (for this reason, they are also known as Causal Networks);

- Undirected PGMs (called Markov Random Fields) are used in computer vision and physics to refer to correlation between random variables.

Bayesian Networks can be used to find or infer the relationship between the inputs and outputs of the simulation, learned from simulation data over multiple replications. Do-calculus allows one to infer, from a Bayesian Network, whether it is possible to identify causal effects from non-experimental data [23]. This provides a principled approach for using these models to answer "what-if" questions. Do-calculus is part of a larger class of methods, called Structural Equation Modelling, which allow the analysis of random variables whose inter-dependencies can be specified in a graphical model [13].

DBNs introduce time information to the Bayesian Network model: it is similar to the HMM and, in fact, it applies the same constraints. DBNs are also used in time series analysis. DBNs can represent time as either a point (instance) or as a timer-interval. DBNs can then be used to learn the dependencies between the state of the simulation at two different time points, so we can infer the expected flow of the simulation and how it affects the final outcomes. In the context of the analysis conducted in this research, this could refer to the state of both forces' movement, detection, shot and kill events.

Poropudas and Virtanen introduce a simulation meta-modelling approach. They use DBNs as a compact representation to understand the evolution of the simulation, and also to run "what-if" analyses [24]. In a previous study they also applied DBNs to analyze air combat simulation data [25]. We plan to extend on this work in the context of analyzing land combat simulation data.

21.6 Conclusion

In this paper, we present a set of techniques used to provide insights into the data generated from combat simulations like COMBATXXI. In particular, unlike most work that has been done in this area, we focus on intra-run events data as opposed to end-of-run data. We believe that the intra-run events data may have the power to expose the causes leading to the observed simulation outcomes.

We suggest that, by using the cumulative count of events technique, the general evolution of the battle can be shown and, when replications are split based on success criteria, the differences in evolution between successful and unsuccessful replications become evident. This technique can also be applied to contrast a single replication against the population in order to help understand where and how this replication differs from the rest. The geo-spatial visualization is another powerful technique that can help demonstrate the differences between outlier and non-outlier replications in terms of geographical locations. The evidence presented in the case study suggests that the outliers for this specific scenario are indeed valid, although further exploration is still required to isolate their cause.

Furthermore, we present other explored methods that can complement the work presented. We showcase the benefits of adding a third dimension to geo-spatial visualizations to display time. We also demonstrate how Hierarchical Clustering can be used to group replications with similar sequences of intra-run events and how temporal machine learning approaches could potentially be used to identify points in time where events shift the balance of the battle. The application of these statistical and machine learning techniques looks promising in enabling us to further elicit the underlying factors driving the evolution of a particular replication or option. We hope that this study highlights a need to augment the current techniques used to analyze combat simulations, and other stochastic event-driven models and promotes more work in this direction.

References

1. Acay LD (2010) Semi-automated causal analysis in simulation based military training. In: Proceedings of the simulation conference and exhibition (SimTecT'10), pp 353–358
2. Balogh I, Harless G (2003) An overview of the COMBAT XXI simulation model: a model for the analysis of land and amphibious warfare. In: Proceedings of 71st military operations research society symposium
3. Bonferroni C (1936) Teoria statistica delle classi e calcolo delle probabilita. Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze 8:3–62
4. Breiman L (1984) Classification and regression trees. Routledge
5. Breunig MM, Kriegel H-P, Ng RT, Sander J (2000) LOF: identifying density-based local outliers. In: ACM SIGMOD Record, Vol 29, pp 93–104. ACM
6. Cervone D, D'Amour A, Bornn L, Goldsberry K (2014) POINTWISE: Predicting points and valuing decisions in real time with NBA optical tracking data. In Proceedings of the 8th MIT sloan sports snalytics conference, Vol 28, p 3. Boston, MA, USA
7. Chau W, Grieger D (2013) Operational synthesis for small combat teams: exploring the scenario parameter space using agent-based models. In 22nd national conference of the Australian society for operations research, pp 196–203
8. Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Measur 20(1):37–46
9. Demediuk S, York P, Walker J, Block F, Drachen A (2019) Role identification for accurate analytics In: DotA 2: A Journey in Applied Cluster Analysis and Unbiased Dataset Creation. In Currently under review (p. TBA). TBA
10. Drachen A, Yancey M, Maguire J, Chu D, Wang IY, Mahlmann T, Klabajan D (2014) Skill-based differences in spatio-temporal team behaviour in defence of the ancients 2 (DotA 2). In: 2014 IEEE games media entertainment, pp 1–8. IEEE
11. Hedley NR (1999) Hagerstrand revisited: interactive space-time visualizations of complex spatial data. Informatica 23(2):155–168
12. Hirotsu N, Wright M (2002) Using a Markov process model of an association football match to determine the optimal timing of substitution and tactical decisions. J Oper Res Soc 53(1):88–96
13. Kaplan D (2008) Structural equation modeling: foundations and extensions, vol 10. Sage Publications
14. Karlin S (2014) A first course in stochastic processes. Academic press
15. Kraak M-J (2003) The space-time cube revisited from a geovisualization perspective. In: Proceedings of the 21st international cartographic conference, pp 1988–1996. Citeseer
16. Kriegel H-P, Kröger P, Schubert E, Zimek A (2009) Outlier detection in axis-parallel subspaces of high dimensional data. In: Pacific-Asia conference on knowledge discovery and data mining, pp 831–838. Springer

17. Kwan M-P (1999) Gender, the home-work link, and space-time patterns of nonemployment activities. Econ Geogr 75(4):370–394
18. Le HM, Carr P, Yue Y, Lucey P (2017) Data-driven ghosting using deep imitation learning. Proceedings of the 11th MIT sloan sports analytics conference
19. Luotsinen LJ, Fernlund H, Bölöni L (2007) Automatic annotation of team actions in observations of embodied agents. In Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, p 9. ACM
20. Miller HJ (1991) Modelling accessibility using space-time prism concepts within geographical information systems. Int J Geogr Informat Syst 5(3):287–301
21. O'May JF, Heilman EG, Bodt BA (2005) Battle command metric exploration in a simulated combat environment. Army Research Lab Aberdeen Proving Ground MD Computational And Information Sciences DIR
22. O'Shaughnessy DM (2006) Possession versus position: strategic evaluation in AFL. J Sports Sci Med 5(4):533
23. Pearl J (1995) Causal diagrams for empirical research. Biometrika 82(4):669–688
24. Poropudas J, Virtanen K (2011) Simulation metamodeling with dynamic Bayesian networks. Eur J Oper Res 214(3):644–655
25. Poropudas J, Virtanen K (2007) Analyzing air combat simulation results with dynamic Bayesian networks. In: 2007 winter simulation conference, pp 1370–1377. IEEE
26. Sasirekha K, Baby P (2013) Agglomerative hierarchical clustering algorithm—a review. Int J Scientif Res Publ 83:83
27. Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics Bull 1(6):80–83
28. Yang Y, Qin T, Lei Y-H (2016) Real-time esports match result prediction