– Proceedings –

48th Annual Conference of the Operations Research Society of South Africa

8–11 September 2019
The Vineyard Hotel, Cape Town, South Africa

ISBN: 978–1–86822–701–3

Editorial Board

Editorial

From the Editorial Board, a warm welcome to the 48th Annual Conference of the *Operations Research Society of South Africa* (ORSSA). The proceedings contain accepted manuscripts over a wide range of OR-related topics. Twenty-two (22) prospective manuscripts were received with fifteen (15), 68%, published in the proceedings. Each manuscript underwent a rigorous double-blind peer-review process by at least two experts in the OR community around the globe. The proceedings will be published online at

`http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf`

Congratulations on all who got their manuscripts accepted. Thank you for every author and reviewer's hard work. A special thanks to the Associate Editors for their willingness to help whenever called upon and more specifically, Susan Campher, who was the pillar during the whole process from start to finish. Regarding the typesetting we thank Celine Jansen van Rensburg for all her efforts. For those unsung heroes contributing to the success of the proceedings, we, as the Editorial Board, thank you.

On behalf of all the conference delegates we thank both the Local Organizing Committee for arranging this event and the support from all the sponsors.

We wish you a pleasant stay in the beautiful City of Cape Town.

Best wishes,

Philip Venter

(e) `Philip.Venter@nwu.ac.za`
(t) +27 18 285 2220
Editor-in-chief: ORSSA Proceedings 2019
Operations Research Society of South Africa

Reviewers

The Editorial Board would like to thank the following reviewers:

RA Bennetto	Icepack, Ireland
DG Breed	ABSA, South Africa
J Bührmann	University of the Witwatersrand, South Africa
D Clark	Chalcid, South Africa
C de Kock	Stellenbosch University, South Africa
AP de Villiers	Stellenbosch University, South Africa
JV du Toit	North-West University, South Africa
MJ Fisher	Independent consultant (Canada)
HW Ittmann	HWI Consulting, South Africa
N Jacobs	Stellenbosch University, South Africa
R Koen	Council for Scientific and Industrial Research, South Africa
HA Kruger	North-West University, South Africa
KM Malan	University of South Africa, South Africa
GS Nel	Stellenbosch University, South Africa
JH Nel	Stellenbosch University, South Africa
WC Pelser	Council for Scientific and Industrial Research, South Africa
PD Pretorius	North-West University, South Africa
MJ Puttkammer	North-West University, South Africa
T Schmidt-Dumont	Stellenbosch University, South Africa
SP Silal	University of Cape Town, South Africa
SE Terblanche	North-West University, South Africa
A van der Merwe	North-West University, South Africa
JH van Vuuren	Stellenbosch University, South Africa
PVZ Venter	North-West University, South Africa
EJ Willemse	Singapore University of Technology and Design, Singapore

Table of contents

Proceedings of the 48th ORSSA Annual Conference
pp. 1–9
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3
©2019

Biases and debiasing of decisions in ageing military systems

WC Pelser*

Abstract

Many of the administrative decisions that must be made in a military environment are complex and rely on a rational analysis of situations. Decisions within the domain of ageing systems are particularly difficult and often riddled with different biases. This paper investigates why rational thinking is not always the norm, and suggests possible ways to assist decision making. A few biases are identified, and available debiasing techniques are discussed. It was found that research in this field is limited and must be expanded in order to ensure optimal decision.

Key words: Cognitive bias, motivational bias, debiasing.

1 Introduction

Military leaders are confronted with decisions from strategic to tactical levels. They encounter uncertainty, lack of information and forced abstraction on the different levels. Defence forces face an operating environment characterized by volatility, uncertainty, complexity, and ambiguity. Military decision makers must make sense of this paradoxical and chaotic setting. Succeeding in this environment requires decision makers that are willing to embrace improvisation and reflection. Decisions within the domain of ageing systems are particularly difficult and often riddled with different biases.

Carl von Clausewitz's metaphoric description of the condition of war is as accurate today as it was when he wrote it in the early 19th century [2]: "Their judgement was based more on wishful thinking than on sound calculation of probabilities; for the usual thing among men is that when they want something they will, without any reflection; leave that to hope, while they will employ the full force of reason in rejecting what they find unpalatable. – Thucydides, History of the Peloponnesian War".

This paper starts with a review of literature on the areas of risk management, cognitive biases, and motivational bias. The paper then applies the insight to the life cycle management of ageing military systems to identify typical risks and biases and suggest ways to mitigate such biases.

[*] Council for Scientific and Industrial Research, Meiring Naude Road, Pretoria, Republic of South Africa, email: wpelser@csir.co.za

2 Literature study

Kahneman & Tversky [4] confirmed the presence of common cognitive biases in the professional judgements of laypersons and experts. They were dissatisfied with the discrepancies of classical economics in explaining human decision making, and developed the initial tenets of a discipline now widely known as behavioural economics.

"Cognitive biases are departures from purely rational thought. They are systematic errors in thinking that prevent us from being entirely rational. There are a number of causes. One common cause is complexity. The human mind is not equipped to deal with the sheer number of factors and their relationships in many situations found in a modern, technologically-complex society. In order to counter this, we commonly use heuristics (rules of thumb) to help assess complex situations." Examples of heuristics may be rules of thumb, educated guesses, gut reaction or common sense [9].

Since Kahneman & Tversky's ground-breaking work, behavioural decision researchers have identified a large number of biases in human judgement and decision making, each bias showing a deviation from a normative rule of probability or utility theory. Montibeller & von Winterfeldt [5] focused on biases that are relevant for decision and risk analysis because they can significantly distort the results of an analysis and are difficult to detect and correct.

Montibeller & von Winterfeldt [5] define a cognitive bias as a systematic discrepancy between the "correct" answer in a judgemental task, given by a formal normative rule, and the decision maker's or expert's actual answer to such a task. Montibeller & von Winterfeldt regard motivational biases, which include conscious and unconscious distortions of judgements and decisions, made within an organizational context and because of self-interest, fear and social pressure, as equally important. Montibeller & von Winterfeldt point out that the methods for reducing motivational biases is fundamentally an unexplored research field.

Motivational bias is the adjustment of response motivated by perception of reward or penalty. Motivational bias is different from cognitive bias, in which a discrepancy, usually subconscious, is introduced by the manner in which the individual processes information [5]. All motivational biases are hard to correct [5]. An example of a motivational bias given by Montibeller & von Winterfeldt [5] is the underestimation of the cost of a project to provide more competitive bids. This is definitely the case in decisions regarding life cycle management of ageing military systems.

According to the *United States* (US) Army [7], research suggests that consequences of intuitive decision making, and therefore of relying on heuristics and succumbing to cognitive biases, become more prevalent in situations of greater complexity or uncertainty. Insight into the nature of human decision making has important implications for the US Army, its mission and the decisions military professionals make. The US Army [7] provides awareness of existing research on decision making in general and, more specifically, cognitive biases, that may inform US Army efforts to prepare its soldiers and leaders for the environment of the future.

3 Reduction or mitigating of biases (debiasing)

Debiasing is the reduction of bias, particularly with respect to judgement and decision making. According to Montibeller & von Winterfeldt [5], debiasing refers to attempts to eliminate, or at least reduce, cognitive or motivational biases. Early attempts showed the limited efficacy of debiasing tools, *i.e.*, to which degree they reduced the bias and brought judgments close to the required normative standard, but recently researchers have become somewhat more optimistic about overcoming biases, particularly with the use of adequate tools and methods [5].

Bias training can result in debiasing at a general level in the long term. Morewedge *et al.* [6] found that training provides mitigating strategies that can reduce some biases. For example, a person may learn or adopt better strategies by which to make judgements and decisions.

Debiasing can occur as a result of changes in external factors, such as changing the incentives relevant to a decision or the manner in which the decision is made. Debiasing can play an important role to improve decisions regarding life cycle management of ageing military systems.

4 Application: Decision biases in the life cycle management of ageing military systems

The following common scenario provides a practical example to discuss biases that influence decisions regarding ageing military systems.

Multiple factors conspire against optimal decision-making in the life cycle management of ageing military systems. It is often the case that system deterioration develops due to sustained under-resourcing and under-investment, especially in financially constrained defence environments. This leads to perceptions of performance obsolescence when compared to the latest state-of-the-art competitors. These deficiencies are often not as relevant when it comes to the "simple" work-horses of defence, such as transport fleets. The typical tendencies are numbered for ease of reference when discussing the possible biases.

1. Multiple operational and support challenges arising from a lack of proper investment (**underinvestment**).
2. **Pessimism and "challenge fatigue"** among operational, maintenance and management staff.
3. Resulting **perceptions of technological obsolescence**, of being unable to logistically support the system into the future, and of imminent end-of-system-life.
4. **Inadequate expertise to analyse** the **cost and effectiveness** of all system lifecycle alternatives. These options include the following:

    (a) Continue as-is;

    (b) Logistic re-capitalisation;

    (c) Life extension;

    (d) Upgrade;

(e) Phase-out with replacement;
(f) Phase-out without replacement (*i.e.*, doing without a capability); and
(g) Phase-out and replacement with a non-materiel solution (*i.e.*, delivering the capability in some other way such as doctrine change, or via another existing system).

5. The so-called **"conspiracy of optimism"** — that a new acquisition project will be successful, within schedule and budget, and that a new system will be more effective, efficient and economical despite frequent evidence to the contrary.
6. **Ingrained preferences** for specific systems, suppliers or countries of origin.
7. **Vested personal interests**, such as the opportunity to manage large programs, often with attractive foreign deployments.
8. **Political preferences**, such as changing alliances.

4.1 Bias 1: Myopic problem representation

Myopia is also known as near-sightedness.

Description: This bias occurs when an oversimplified problem representation is adopted, based on an incomplete mental model of the decision problem [5].

Evidence: This bias focuses on one option — to acquire a new system — regardless of implications, that defies all logic. It focuses on a small number of objectives — a single future state of the world — such as unfounded expectations of lower operations, acquisition and support cost [5].

Relevant tendencies are the following:

- 1 — Under-investment; and
- 4 — Inadequate expertise to analyse cost and effectiveness.

Debiasing techniques are the following:

- Explicitly encourage thinking about more objectives [5];
- Encourage evaluating viable new alternatives [5]; and
- Encourage thinking of other possible states of the future [5].

4.2 Bias 2: Availability or ease of recall

Description: The bias occurs when the probability of an event that is easily recalled, is overstated [8].

Evidence: In this case, the eagerness of having a new system is overstated. The expectation of a new system without problems overshadows the train of thought. All elements of life cycle costs of systems are often not taken into account.

Relevant tendencies are the following:

- 5 — Optimism conspiracy.

New equipment tends to have a "honeymoon period" in which few problems are experienced; institutional memory of this tendency provides an easily available metaphor.

Debiasing techniques are the following:

- Encourage thinking about other possible states of the future;
- Apply devils advocacy; and
- Provide facts and statistics [5].

4.3 Bias 3: Affect influenced

Description: This bias occurs when there is an emotional predisposition for — or against — a specific outcome or option that taints judgements [5]. This is probably triggered when facing a difficult decision and a "gut feeling" is used.

Evidence: Several studies [5] that assess the role of affect causing an inverse perceived relationship between positive and negative consequences related to pandemics and human-caused hazards. Affect influences the estimation of probabilities of events [5].

Relevant tendencies are the following:

- 3 — Perceptions of technological obsolescence;
- 7 — Vested personal interests; and
- 8 — Political preferences.

Debiasing techniques are the following:

- Avoid loaded descriptions of consequences in the attributes [5];
- Cross-check judgements with alternative elicitation protocols when eliciting value functions, weights, and probabilities [5];
- Use multiple experts with alternative points of view [5]; and
- Enforce formal decision making methods.

4.4 Bias 4: Confirmation

Description: Confirmation bias is the tendency to find, interpret, favour and remember information so that it confirms pre-existing beliefs [1]. This may lead to the unconscious selective use of facts.

Evidence: Confirmation bias is evident in several experimental settings, such as in information gathering, selection tasks, evidence updating, and own-judgement evaluation [5].

Relevant tendencies are the following:

- 3 — Perceptions of technological obsolescence;
- 7 — Vested personal interests; and
- 8 — Political preferences.

Debiasing techniques are the following:

- Use multiple experts with different points of view [5];

- Generate and test alternative hypotheses;
- Challenge assessments with counter-facts [5]; and
- Probe for evidence for alternative propositions [5].

4.5 Bias 5: “Conspiracy of optimism” or optimism bias

Description: The bias occurs when the desirability of an outcome leads to an increase in the extent to which it is expected to occur. It is also called “wishful thinking” or “desirability of a positive event or consequenc” [5].

People tend to overestimate the probability of positive events and underestimate the probability of negative events happening to them in the future. A number of factors can explain unrealistic optimism, such as perceived control [8].

Evidence: This bias is observed when people’s subjective confidence in their own ability is greater than their objective (actual) performance. It is frequently measured by having experimental participants answer general knowledge test questions; they are then asked to rate on a scale how confident they are in their answers [1]. The UK Treasury now requires that all ministries develop and implement procedures for megaprojects that will curb so-called “optimism bias” [3].

Relevant tendencies are the following:

- 5 — Optimism conspiracy.

Debiasing techniques are the following:

- Use formal methods of decision making;
- Provide facts and statistics [5];
- Use multiple experts with alternative points of view [5]; and
- Use decomposition and realistic assessment of the facts [5].

4.6 Bias 6: Desirability of options or choice

Description: This bias leads to over- or underestimating probabilities, consequences, values, or weights in a direction that favours a desired alternative [5].

Evidence: Only anecdotal evidence, such as the biased estimates of probabilities and impacts in risk assessment [5].

Relevant tendencies are the following:

- 6 — Ingrained preferences.

Debiasing techniques are the following:

- Use analysis by multiple stakeholders with different backgrounds providing different value perspectives; and
- Use multiple experts with different opinions for evaluation [5].

4.7 Bias 7: Omission of important variables

Description: This is the tendency to overlook important aspects when making decisions.

Evidence: Important facts and decision variables are omitted to enhance a particular choice.

Relevant tendencies are the following:

- 4 — Inadequate expertise to analyse cost and effectiveness;
- 5 — Optimism conspiracy;
- 6 — Ingrained preferences;
- 7 — Vested personal interests; and
- 8 — Political preferences.

Debiasing techniques are the following:

- Explicitly encourage thinking about more objectives [5];
- Use formal methods of decision making;
- Defining balanced expert groups;
- Encourage evaluating viable new alternatives; and
- Encourage thinking of other possible states of the future.

4.8 Bias 8: Pessimism

Description: The pessimism bias is a cognitive bias that causes people to overestimate the likelihood that bad things will happen. This bias distorts people's thought process, and can be detrimental to your emotional wellbeing, which is why it is strongly associated with various mental health issues, and most notably with depression.

Evidence: Undesirability of a negative event or outcome (precautionary thinking, pessimism) [5]. Pessimism will unquestionably influence the thought processes of decision makers.

Relevant tendencies are the following:

- 1 — Under-investment;
- 2 — Pessimism and "challenge fatigue"; and
- 3 — Perceptions of technological obsolescence.

Debiasing techniques are the following:

- Use formal methods for the analysis;
- Use uninvolved, independent teams to provide cross checks; and
- Provide facts and statistics to inform decision makers.

Table 1 summarises typical tendencies, and possible biases underlying these tendencies.

		Biases							
		1	2	3	4	5	6	7	8
	Tendencies	Myopic	Availability	Affect	Confirmation	Optimism	Choice	Omission	Pessimism
1	Under-investment	X							X
2	Pessimism								X
3	Obsolescence perceptions			X	X				X
4	Inadequate analysis	X						X	
5	Optimism conspiracy		X			X		X	
6	Ingrained preferences						X	X	
7	Personal interests			X	X			X	
8	Political preferences			X	X			X	

Table 1: *Summary of tendencies and possible biases.*

5 Conclusion

This paper summarised literature on the topic of cognitive biases. Some typical biases applicable to this scenario of decision-making in the life cycle management of ageing military systems were identified and defined. Evidence to substantiate each bias was discussed, followed by the specific relevant tendencies within the scenario of decision-making in the life cycle management of ageing systems. Lastly, debiasing techniques that could be of value in combating these biases and tendencies were suggested.

The article highlights the existence of typical erroneous tendencies and biases in decision-making on life cycle management of military systems, and provides a conceptual basis for a systematic approach to bias detection and mitigation. The conceptual basis can serve as foundation for further analysis. It will ensure that all tendencies and biases are covered and broadened to other decisions in a specific military environment, for example research in support of developmental training.

References

[1] Dekuyer J, 2018, *Instructor bias at military training institutions*, [Online], Available from `https://groundedcuriosity.com/instructor-bias-at-military-training-institutions/`.

[2] Dobson-Keeffe N & Coaker W, 2015, *Thinking more rationally: Cognitive biases and the joint military appreciation process*, Australian Defence Force Journal, **197**, pp. 5.

[3] Flyvbjerg B, 2014, *What you should know about megaprojects and why: An overview*, Project management journal, **45(2)**, pp. 6–19.

[4] Kahneman D & Tversky A, 1974, *Judgment under uncertainty: Heuristics and biases*, Science, **185(4157)**, pp. 1124–1131.

[5] Montibeller G & von Winterfeldt D, 2015, *Cognitive and motivational biases in decision and risk analysis*, Risk analysis, **35(7)**, pp. 1230–1251.

[6] Morewedge CK, Yoon H, Scopelliti I, Symborski W, Korris JH & Kassam KS, 2015, *Debiasing decisions: Improved decision making with a single training intervention*, Policy insights from the behavioral and brain sciences, **2(1)**, pp. 129–140.

[7] Rodman, J, 2015, *Cognitive biases and decision making: A literature review and discussion of implications for the US Army*, White Paper, US Army Mission Command - Capabilities Development and Integration Directorate, Fort Leavenworth (KS).

[8] Sharot, T, 2011, *The optimism bias*, Current biology, **21(23)**, pp. R941–R945.

[9] Williams, BS, 2010, *Heuristics and biases in military decision making*, Military Review, US Army, **90(5)**, pp. 40–52.

Proceedings of the 48th ORSSA Annual Conference
pp. 10–18
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3
©2019

Comparing the efficiency of two picking policies in a retail warehouse *via* simulation

P Cilliers[*] JH van Vuuren[†]

Abstract

A well-known logistics company is faced with the problem of deciding on the most efficient stock picking policy of two possible strategies for the picking of multiple frozen stock items in the freezer section of one of its retail warehouses. Both picking strategies are aimed at picking frozen stock items for multiple customers along a single delivery vehicle route at a time, and repeating this for various vehicles. The picking strategies are referred to as Picking Policies A and B, where Picking Policy A prioritises the picking of frozen stock items on a customer-by-customer basis, while Picking Policy B prioritises the picking process on a stock item by stock item basis. In order to compare the relative efficiency and practical workability of these picking policies, an agent-based simulation modelling approach is adopted. The aim in the stock picking simulation model developed for this purpose is to record the total time spent by order pickers in the system when picking (according to the two picking policies), packing and transporting stock items from the freezer section of the retail warehouse to the dispatch area for each picking policy over time, based on historical demand data. Due to stochastic elements involved in customer orders and order picker behaviour, the study is carried out at a specific level of statistical significance. The results show with 99% confidence that Picking Policy B outperforms Picking Policy A. This result is intended as decision support for the investigated logistics company, assisting it in making an informed decision in respect of the implementation of a preferred picking policy.

Key words: Logistics, warehouse, picking policies, agent-based modelling, simulation.

1 Introduction

Modern logistics has become more and more customer-centric in recent years [3]. Retail warehouses, therefore, continually strive to improve their responsiveness to customer orders. Any underperformance in respect of order picking in a retail warehouse can lead to

[*] Department of Industrial Engineering, University of Stellenbosch, Private bag X1, Matieland, 7602, Republic of South Africa, email: 18969771@sun.ac.za

[†] **(Fellow of the Operations Research Society of South Africa)**, Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602, Republic of South Africa, email: vuuren@sun.ac.za

inefficiency in the operations of the warehouse, and consequently the entire supply chain, potentially causing major customer dissatisfaction and/or lost opportunity cost. In order to achieve the level of responsiveness expected by customers, retail warehouses primarily aim to shorten the time required to consolidate orders from warehouse *stock keeping units* (SKU). The travel time of an order picker when consolidating orders is typically an increasing function of the travel distance covered by the picker within the warehouse. Consequently, warehouse managers tend to pursue minimisation of expected order picker stock picking duration as the primary objective in warehouse management optimisation. A critical problem experienced in retail warehouses is to decide on an order picking policy aimed at pursuing this objective [2]. The development of high-quality order picking policies for retail warehouses by exact or heuristic optimisation means has therefore been among the top retail research topics in recent years [1].

The focus in this project is on a real-world case study of the picking policies within a retail warehouse managed by an industry partner attached to the project. There is currently uncertainty among the management of the industry partner's Cape Town warehouse as to which of the two proposed policies should be implemented (*i.e.,* is superior in terms of the objective mentioned above). An agent-based simulation model is therefore developed in this paper for quantifying the effects of two different picking policies so that their relative efficiencies can be compared within the frozen food section of the industry partner's Cape Town warehouse.

According to the first picking policy, henceforth referred to as *Picking Policy A*, order pickers are required to pick all the stock destined for delivery to customers along a single delivery vehicle route, but pick per customer along that route at a time. Order pickers thus only pick what is required of a specific product for a single customer at a time and may hence have to return later to a specific SKU to pick stock for a separate customer by the same vehicle. According to *Picking Policy B*, on the other hand, order pickers are also assigned to pick all the products for delivery along a single vehicle route, but are not required to perform this picking process separately per customer. Order pickers therefore pick all the necessary products required for the customers assigned to the entire route in one go, and deposit the stock in a dispatch area where the orders then have to be separated into customer consignments.

A comparison is carried out in this project of the aforementioned two picking scenarios in an extensive computer simulation experiment. The objective of this experiment is to gather statistical evidence according to which a decision support recommendation can be made to the industry partner in respect of which policy to implement.

2 Simulation model

An agent-based modelling approach is adopted in this paper for implementation within the AnyLogic software suite. AnyLogic offers continuous, discrete and, importantly, GIS animation and mobility functionalities, which are useful in the context of this study [4].

Several components of the AnyLogic software suite are utilised to transform the underlying dynamics and behaviour of the operations within the retail warehouse of the industry partner attached to this project into the computer simulation environment. The simulation model components are object classes which represent distinct agents, or entities, in the simulation model. Agents are viewed as the building blocks of the simulation model and may represent physical or abstract entities. Picking trolleys, reach trucks, SKUs, materials, and pallet batches are all instances of object classes employed in the simulation model developed as part of this project. In order to translate the dynamics and behaviour of the orders picked by the picking trolleys and the replenishment operations associated with the empty SKUs into executable code in AnyLogic, several components of agents, variables, parameters, functions and scheduled event triggers are adopted to construct code sequences which define agents that mimic the behaviour of the order picking process of the industry partner.

2.1 General model description

The primary entities responsible for the execution of the underlying simulation model consist of agents referred to as picking trolleys, picked batches, material batches and reach trucks. A *picking trolley*, which is manned by an order picker, is responsible for the consolidation of customer orders. The flow chart constructed for the agent *picking trolleys* in Figure 1 dictates the behaviour of a *picking trolley. Picked batches* function as queues which facilitate storage of a pre-specified number of product units, while a *picking trolley* travels from point to point with the *picked batch* present on it. Consequently, the *picked batch* acts as a container in which the order picker can place product units. Once the *picked batch* reaches its capacity or the end of an order consolidation list, it is programmed to travel to the depot area where the *picked batch* is released. *Material batches*, on the other hand, are used to store and batch each type of available product unit into the corresponding batch sizes in which the product units are stored in the SKUs of the retail warehouse. Once the *material batch* is empty, a *reach truck* is responsible for the removal and replenishment of an identical *material batch* in the SKU.

Upon initiating a simulation run, the *graphical user interface* (GUI) is visually presented upon execution to allow the user to enter input parameters for a desired comparison. For both picking policies, the number of picking trolleys responsible for the order picking operations and the number of reach trucks responsible for the SKU replenishment operations are specified *via* the GUI so as to allow the user to observe and compare the effects of various combinations of picking trolleys and reach trucks introduced.

Several assumptions were made with respect to the development of the simulation model. These assumptions are as follows:

- An order picker's travel speed is uniform at 3.5 m/s. This speed remains constant regardless of the number of product units that are placed in the picked batch (which is located on the picker's picking trolley).
- A reach truck's travel speed is uniform at 5 m/s. This speed also remains constant regardless of whether it is travelling empty or carrying a material batch.
- Order pickers always remove the correct products from each picking bin.
- A material batch's content decreases by the same number of product units as that by which the order picked by an order picker increases.

- The time taken by an order picker to pick a product unit from a picking bin and to place it in the picking trolley's picked batch is constant for each picking bin.
- Picking trolleys do not experience any warehouse traffic as a result of congestion within SKU aisles.
- In cases where order pickers are required to pick a quantity of product units larger than the current number of product units present in the associated material batch, the material batch is removed from the SKU and is replenished, whereafter the order picker picks the remaining product units from the replenished material batch. Hereafter, the order picker continues with its picking operations as dictated by the shipment order list.

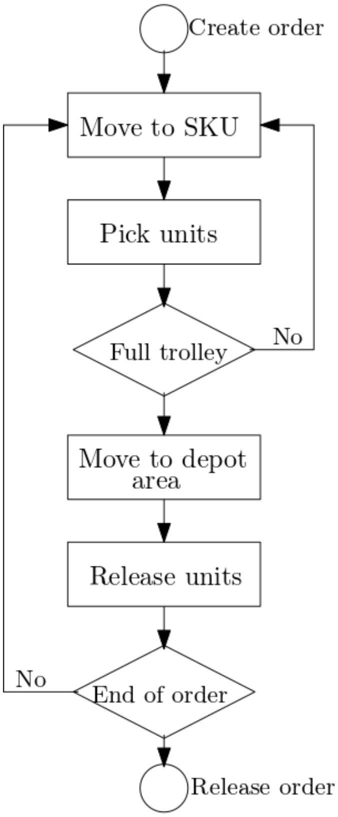
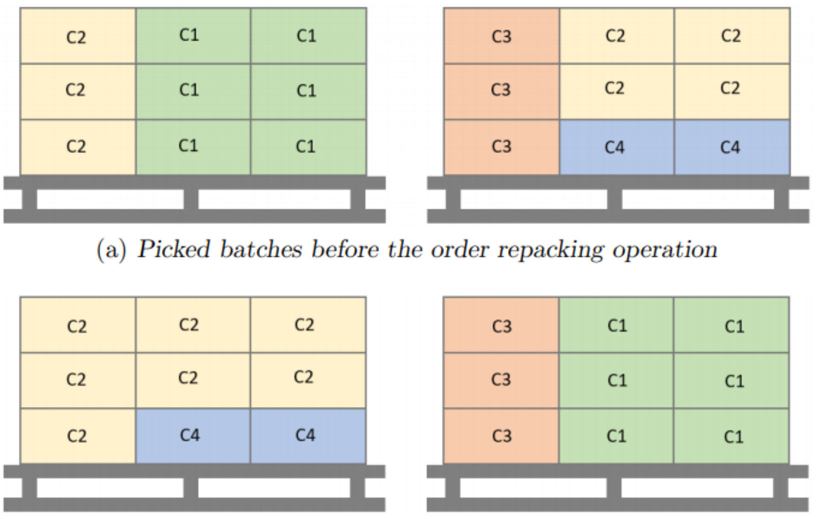


Figure 1: *Flowchart of the behaviour of a picking trolley when consolidating an order.*

2.2 The dynamics of the model components

The order repacking operation is applied to sort all picked batches associated with a single order consolidation list so that all product units destined for a single customer order is located on a single picked batch. Although a customer's order can be separated in multiple picked batches, as illustrated in Figure 2(a), a customer order is preferably located on a single picked batch if feasible according to the picked batch capacity constraint, as illustrated in Figure 2(b).

As Picking Policy A is picked per customer, the order repacking process only occurs in this case when a new customer's product units are located in different picked batches due to the behaviour adopted by an order picker. In cases where the product units of a customer order are located in two different picked batches due to the order volume being larger than the capacity available in a picked batch, as illustrated by Figure 3, no order repacking is applied to these picked batches.



(a) *Picked batches before the order repacking operation*

(b) *Picked batches after the order repacking operation*

Figure 2: *Nature of the repacking process. Different colours denote different customer orders (all destined for delivery by a single vehicle).*

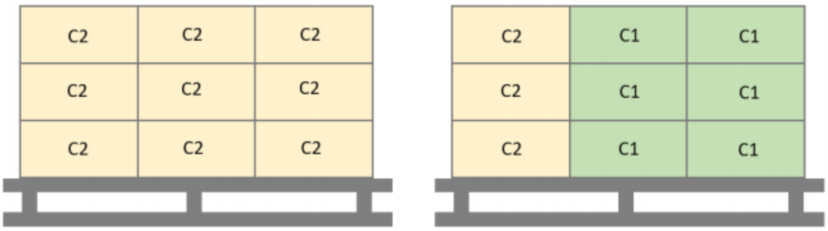


Figure 3: *Picked batches constituting a large customer order when picked according to Picking Policy A. Different colours denote different customer orders (all destined for delivery by a single vehicle).*

The occurrence of repacking is thus smaller for Picking Policy A than for Picking Policy B, as 100% of picked batches are required to be repacked when adopting Picking Policy B (because consecutive entries in the shipment order list may be associated with different customers). In cases where a customer order's product units are located in multiple picked batches, such a multi-picked batch is assumed to be repacked at a rate of 600 product units per hour at the depot area, as six active employees are responsible for repacking. In the simulation model, the order repacking process is only executed once all picked batches have been completed (*i.e.*, at the end of the simulated period, the length of which is one week for the purpose of this paper)[1]. Once all shipment orders have been completed for the given time period, the model scans each picked batch and determines the number of picked batches associated with each customer order. Any customer's product units that are located in multiple picked batches, cause those batches to be added to the repacking list. Once the simulated repacking duration has been calculated, this time is added to a total duration tally associated with the respective picking policy.

2.3 Picking policy simulation

The order picking list presented to each order picker prior to commencing picking operations conforms to the tabular format illustrated in Table 1. Picking operations carried out according to Picking Policy A are picked per customer. All customer orders are therefore picked separately (*i.e.,* a customer order is completed before commencing picking operations for the next customer in the order picking list). Picking operations according to Picking Policy B are not picked per customer. Instead, the product units required by all customers associated with a single shipment are picked together, one product unit at a time, so that order pickers never have to return to the same picking bin during the picking operations of a shipment. While the same simulation model is applicable to both picking policies, a modelling distinction between the two policies is made by presenting the model with differently sorted picking lists, as illustrated in the table below.

Original order picking list (Picking Policy A)			
Pick sequence	Customer	Product unit	Picking bin
1	C1	P1	a
2	C1	P2	b
3	C1	P3	c
4	C1	P4	d
5	C2	P1	a
6	C2	P4	d
7	C2	P5	e
8	C2	P6	f
9	C3	P1	a
10	C3	P7	g
11	C3	P8	h
12	C4	P4	d
13	C4	P7	g
14	C4	P9	i

Picking Policy B			
Pick sequence	Customer	Product unit	Picking bin
1	C1	P1	a
2	C2	P1	a
3	C3	P1	a
4	C1	P2	b
5	C1	P3	c
6	C1	P4	d
7	C2	P4	d
8	C4	P4	d
9	C2	P5	e
10	C2	P6	f
11	C3	P7	g
12	C4	P7	g
13	C3	P8	h
14	C4	P9	i

Table 1: *The original format of a picking list (that is, for Picking Policy A) and a picking list according to Picking Policy B.*

3 Simulation results

The *key performance indicator* (KPI) employed in this paper to measure the relative efficiency of the two picking policies, is the total time it takes order pickers to complete both

[1] In the retail warehouse of the industry partner, vehicles despatch consolidated orders from the depot daily. However, as the frequency of despatches, for the purpose of this comparison study, do not have an effect on the outcome of this paper, vehicles are modelled as despatching consolidated orders weekly.

a week's shipment picking and repacking operations associated with actual historical demand data, measured over multiple weeks. As customer demand and the sizes of customer orders fluctuate over time, the number of order shipments to customers per week varies. This results in a comparison of the relative efficiency of the picking policies over weekly numbers of shipments ranging from 11 shipments (during week 5) to 68 shipments (during week 7).

Because all batches picked according to Picking Policy B have to be repacked (at a rate of 600 product units per hour), the total duration (measured in seconds) of consolidating orders according to Picking Policy B is calculated as

$$\text{Total Duration} = \text{Simulated Duration} + 6(\#\text{ of Picked Batches}).$$

In contrast, only a proportion ρ_i of batches picked according to Picking Policy A that have to be repacked is calculated separately for the simulation run of week i. The repacking rate remains constant for both picking policies (600 product units per hour), and so the total duration of order consolidation according to Picking Policy A is calculated as

$$\text{Total Duration} = \text{Simulated Duration} + 6\rho_i(\#\text{ of Picked Batches}).$$

Picking Policies A and B were simulated over the sixteen weeks for which demand data were provided by the industry partner for a combination of three order pickers and two reach trucks being available to carry out order consolidation for both picking policies (these numbers were specified by the industry partner). The results obtained by applying the simulation model to the aforementioned data are presented in Figure 4. The results are reported separately for the sixteen weeks for which data were made available by the industry partner. The fluctuations present in the results are due to varying customer demand.

The results of Figure 4 are further consolidated in the form of box plots in Figure 5. From these boxplots, the following conclusions may be drawn:

1. The variance of order consolidation durations for each picking policy is largely due to fluctuating customer demand per week.
2. The quickest time to consolidate a week's shipment orders was achieved via Picking Policy B during week 5.
3. The longest time to consolidate a week's shipment orders was achieved via Picking Policy A during week 7.
4. The mean order consolidation time according to Picking Policy B is 27% shorter than that of Picking Policy A.

The Friedman statistical test was employed to test whether the relative performances of the two picking policies could be considered to be statistically distinguishable at a 99.9% level of confidence (*i.e.,* for a statistical significance of $\alpha = 0.001$). The null hypothesis H_0 of the test was taken as "there is no significant difference between the mean order consolidation time according to Picking Policy A and that according to Picking Policy B." The test statistic was computed as $F = 16$, while the relevant χ^2-critical value was found to be $\chi^2_{1,0.001} = 10.828$. The ρ-value associated with the Friedman test was $\rho = 0.00006$. The null hypothesis H_0 was therefore rejected (*i.e.,* it was shown with 99.9% confidence that there is indeed a significant difference between the mean order consolidation time according to Picking Policy A and that according to Picking Policy B).

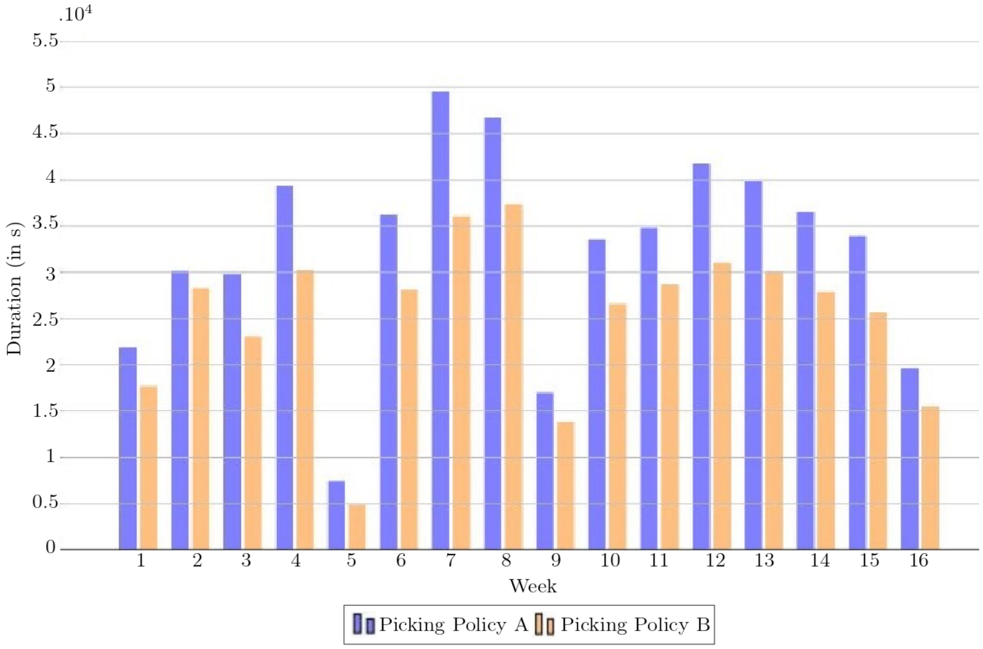


Figure 4: *Order consolidation durations for three order pickers and two reach trucks.*

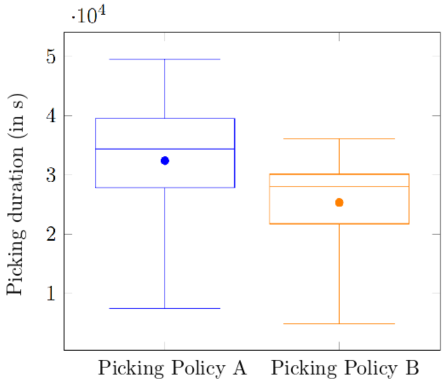


Figure 5: *Comparison of the weekly order consolidation durations over sixteen weeks according to Picking Policies A and B, for three order pickers and two reach trucks in the modelled system.*

4 Conclusion

The results and insights generated in this paper will allow the industry partner's managers to compare the two policies on equal terms. This ability is expected to bring valuable additional facts to the discussion table, which will aid the industry partner in implementing the superior picking policy. Furthermore, based on the level of detail incorporated into the simulation model, as well as the fact that regular interaction was maintained with a representative of the industry partner in respect of providing model input data and parameters, the manner of execution of this project caused the industry partner to develop confidence in the model and the results that it yielded.

References

[1] Elbert RM, Franzke T, Glock CH & Grosse EH, 2017, *The effects of human behavior on the efficiency of routing policies in order picking: The case of route deviations*, Computers and Industrial Engineering, **111**, pp. 86–92.

[2] Gagliardi JP, Renaud J & Ruiz A, 2007, *A simulation model to improve warehouse operations*, Proceedings of the 39$^{\text{th}}$ Winter Simulation Conference, Montreal, pp. 2012–2018.

[3] Richards G, 2017, *Warehouse management: A complete guide to improving efficiency and minimizing costs in the modern warehouse*, Kogan Page Publishers, London.

[4] XJ Technologies Company, 2010, *AnyLogic 7 user's guide*, [Online], [Cited March, 2018], Available from `http://www.xjtek.com/`

Proceedings of the 48th ORSSA Annual Conference
pp. 19–26
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3

Comparing the performance of different metaheuristics when solving a stochastic bi-objective job shop scheduling problem

S Snyman*   JF Bekker†

Abstract

Scheduling is, at present, still considered a recurring problem in modern manufacturing environments. The *Job Shop Scheduling Problem* (JSP) is one of the most difficult scheduling problems to solve and is known to be an NP-hard problem. To solve the problem, researchers have implemented several dispatching rules, heuristics and metaheuristics in the single and multi-objective domain. Typically, metaheuristics have performed better than common rules and heuristics, and have proven to be more effective in finding good schedules. In the past, researchers used weakly conflicting objectives to find good solutions for the multi-objective JSP. This paper provides a comparison of different metaheuristics for solving a stochastic bi-objective JSP, where the makespan and total overtime objectives are to be minimised. The makespan and total overtime objectives are shown to be conflicting. A novel ranking and selection technique, *Procedure MMY*, is implemented to determine the approximate Pareto set and guarantee the probability of correct selection of the best scenarios.

Key words: Job shop scheduling, Procedure MMY, ranking and selection.

1 Introduction

According to Yang & Gu [20] the scheduling problem is viewed as one of the most important and hardest combinatorial optimisation problems due to its complexity and frequency in practical applications. Furthermore, machine scheduling problems can be divided into, but are not limited to, single machine, parallel machines, flow shop, flexible flow shop, job shop, open shop, *etc.* This paper will focus on the *Job Shop Scheduling Problem* (JSP) which is an NP-hard problem, when more than three resources are present [6, 12]. This problem is not only NP-hard, but it also has the well-earned reputation of being one of the most computationally stubborn combinatorial problems considered to date [2].

The JSP has previously been modelled and evaluated in the multi-objective optimisation domain, by minimising several common performance indicators (*e.g.,* makespan, average

*Department of Industrial Engineering, University of Stellenbosch, Private bag X1, Matieland, 7602, Republic of South Africa, email: snymans@sun.ac.za

†Department of Industrial Engineering, University of Stellenbosch, Private bag X1, Matieland, 7602, Republic of South Africa, email: jb2@sun.ac.za

job tardiness, average flow time, *etc.*) as seen in [11]. These performance indicators are, however, weakly conflicting. This paper therefore proposes a bi-objective optimisation problem that minimises the makespan and total overtime objectives, which are conflicting objectives.

This paper reports on part of a research project that has the objective to develop a prototype real-time simulation scheduler for a sensorised job shop, which is to serve as a decision support tool so that unexpected disturbances in the shop can be overcome by the generation of new schedules with real-time data from the shop. The paper will focus on the expansion of the developed system, as described in [15, 16, 17], from the single-objective to the multi-objective domain. Several metaheuristics will be compared based on their performance when solving the stated bi-objective JSP. The best performing metaheuristic will then be used in conjunction with the recently developed MMY ranking and selection algorithm [22] that ensures the probability of correct selection of alternatives.

A summary of the relevant literature is provided in §2, which is followed by a discussion of the experiments that were conducted with the results detailed in §3. The outcome of the ranking and selection is discussed in §4, and finally, a conclusion is provided in §5.

2 Literature review

In this section, the required literature to understand the JSP, as well as the methods that were implemented to solve the bi-objective JSP, is provided. The previous use of the stated objectives is also mentioned and discussed.

According to McKay *et al.* [10], management scientists became interested in the scheduling problems faced by manufacturers during the 1950s and early 1960s. The JSP is one of these scheduling problems and it is still attracting the attention of researchers due to its complexity. The JSP can be defined as follows [5, 7, 8, 9, 24]:

- There is a set J of n jobs $\{J_1, J_2, \ldots, J_n\}$ that must be processed on a set M of m machines $\{M_1, M_2, \ldots, M_m\}$.
- Each job i consists of a finite and predetermined sequence of j operations, $O_i = \{o_{i,1}, o_{i,2}, \ldots, o_{i,j_i}\}$, where the operation order is fixed.
- An operation may only be assigned to an available machine forming part of the set M.
- A machine can process at most one operation at a time, and no preemption is allowed.
- Each operation $o_{i,j}$ has a processing time $p_{i,j}$.
- The aim is to find a schedule for processing these n jobs on the m machines.

Dispatching rules are typically used to generate the schedules for processing these n jobs. The dispatching rules indicate which job the operator must selected next, to start its processing step on the machine. Typical dispatching rules that have been used to solve the JSP include, but are not limited to, shortest processing time first, earliest due date, first-come-first-served, most-important-job-first, *etc.* [4, 19]. Although these dispatching rules are quite simple to implement, they are consistently outperformed by metaheuristics, which was observed in [17]. Due to this outcome, the study will focus on implementing metaheuristics to find sufficiently high quality solutions to the stated JSP.

Due to the large amount of research that has gone into implementing different metaheuristics to solve the JSP, as observed in a survey by Chaudhry & Khan [3], it is difficult to choose a single best approach to solve any given JSP. For this reason, it was decided to implement four different metaheuristics to solve the JSP presented in this research study. The four metaheuristics that were implemented are the *Multi-objective Simulated Annealing* (MOSA) algorithm, which is a trajectory-based algorithm, the *Non-dominated Sorting Genetic Algorithm II* (NSGAII) and the *Multi-objective Genetic Algorithm* (MOGA) — which are biologically inspired population-based algorithms — and lastly, the *Multi-objective Optimisation Cross Entropy Method* (MOOCEM), which is another population-based algorithm with a stochastic basis.

After the selection of the metaheuristics, the objectives that need to be optimised had to be determined. The objectives that have typically been used throughout literature to solve the JSP, include the makespan, total tardiness and total idle time of machines, as observed in [1, 11, 14]. These objectives are, however, not in conflict with each other. If the makespan of a schedule is minimised, it would be expected that the total tardiness of jobs and the idle time of machines would also be minimised, and *vice versa.* This assumption was tested and the output of the test confirmed that when the makespan of a schedule is minimised, the tardiness will also be minimised, as seen in Figure 1. The correlation coefficient was determined to be 0.738 by conducting a correlation test. The positive correlation between these two objectives, illustrated by the blue trend line in Figure 1, therefore suggests that they are strongly related to each other and that they should not be used in conjunction in a multi-objective JSP.

Due to the assumption that preemption is not allowed, the processing step of a job must be finished without interruption. This can result in overtime if the processing step is still ongoing at the closing time of the shop. It was therefore decided to use makespan and total overtime as objectives. When the makespan of a schedule is minimised, it is expected that more overtime is required and that the total overtime will increase. On the other hand, if the total overtime is minimised, the makespan of the schedule will increase. This assumption was also tested by conducting a correlation test. The correlation coefficient was determined to be -0.066, which indicates that there is low correlation between these two objectives. The correlation between the objectives are illustrated by the blue trend line in Figure 2.

Overtime has previously been used as an objective in a multi-objective JSP, as seen in [13, 23]. Rohaninejad *et al.* [13] studied a multi-objective flexible JSP with machine capacity constraints. In the study, a hybrid genetic algorithm based on the ELECTRE method was used. Zhang *et al.* [23] determined a robust schedule for a flexible job shop scheduling problem with flexible workdays, by using either a goal-guided multi-objective tabu search or a goal-guided multi-objective hybrid search. Both these studies did, however, assume deterministic processing times, which may not adequately mimic a real world job shop, as processing times are often stochastic. Therefore, what differentiates this study from the previous studies, is the use of stochastic processing times. Both these studies also focus on smaller instances for testing by limiting the number of jobs to sixteen in [13] and fifty in [23]. This study will, however, use three different test scenarios, *i.e.*, 50, 100 and 200 jobs, that must be processed on eight machines. In the next section, the test scenarios and their results are discussed in further detail.

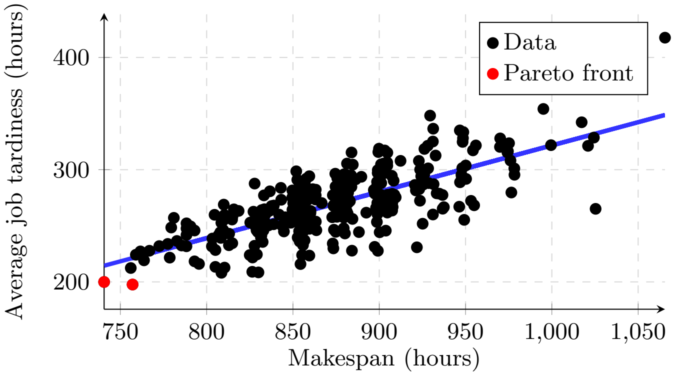


Figure 1: *Correlation of makespan and average job tardiness as objectives.*

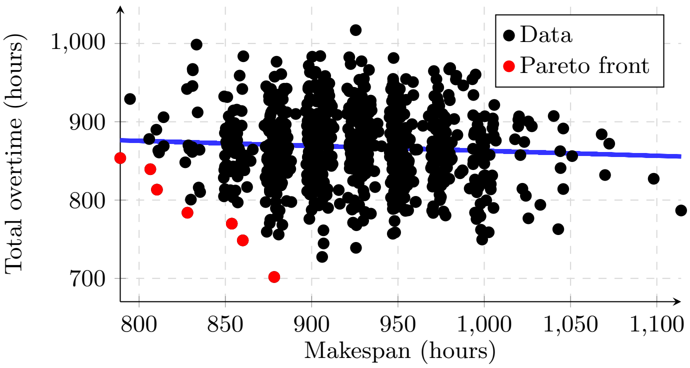


Figure 2: *Correlation of makespan and total overtime as objectives.*

3 Experiments and results

A discussion of the test scenarios that were conducted, as well as the results that were observed, is provided in this section. First, three test scenarios were created for the purpose of testing the metaheuristics. Weber *et al.* [18] state that the standardisation and systematisation of test data for the JSP is still lacking. This is also true for the stochastic JSP, and, therefore, randomly generated test data for the three test scenarios containing 50, 100 and 200 jobs, respectively, were used. In order to ensure that the data would not favour a single method, careful attention was given when the data were randomly generated. The processing times were sampled from a log-normal distribution to introduce stochastic processing times. The log-normal distribution was chosen in this study to ensure that the processing times that were sampled are all positive, as negative processing times are not allowed.

Each of the chosen metaheuristics was then applied to the stated test scenarios, by conducting 500 experiments per metaheuristic for each scenario. To determine the best performing metaheuristic, performance metrics needed to be selected in order to compare the generated approximate Pareto sets. The metrics selected for the comparison tests are the *generational distance* (GD), *hyper-area ratio* (HR) and *maximum spread* (MS), as presented in [21]. The results were compared with the paired t-test because the experiments are independent and normality was assumed through the central limit theorem, as well as unknown and unequal variances for the results. The mean performance metric value for each metaheuristic and test scenario is provided in Table 1.

	50 jobs			100 jobs			200 jobs		
Metaheuristic	GD	HR	MS	GD	HR	MS	GD	HR	MS
MOGA	7.232	0.937	0.515	96.002	1.064	0.421	129.163	1.116	0.406
NSGAII	3.788	0.909	0.647	54.052	1.020	0.543	77.466	1.080	0.561
MOOCEM	9.336	0.921	0.458	119.291	1.060	0.374	182.124	1.132	0.291
MOSA	5.738	0.939	0.586	77.555	1.048	0.479	97.240	1.136	0.530

Table 1: *Mean performance metric values for each metaheuristic and the different test scenarios.*

An example of a p-value table that was generated from the paired t-tests that were conducted can be seen in Table 2. Because of a p-value of 0.744, it is evident that the performance of the MOSA and MOGA metaheuristics for the HR performance metric of the 50 job test are not statistically different. The performance of MOOCEM and NSGAII are also not statistically different (with a p-value of 0.06). All the other comparisons indicate that the metaheuristics are statistically different from the others for the HR performance metric. Due to space limitations, the other p-value tables will not be provided.

	MOGA	NSGAII	MOOCEM	MOSA
MOGA	–	0	0.015	0.744
NSGAII	0	–	0.060	0
MOOCEM	0.015	0.060	–	0
MOSA	0.744	0	0	–

Table 2: *HR p-value table for 50 job test scenario.*

The comparison tests that were conducted for all three test scenarios indicated that the NSGAII consistently performed the best of all the metaheuristics. The NSGAII was therefore chosen to be used in conjunction with the recently developed ranking and selection algorithm for stochastic systems, called *Procedure MMY*. The implementation of Procedure MMY and results observed for the procedure are discussed next.

4 MMY integration and results

The MMY ranking and selection algorithm for discrete stochastic simulation problems was proposed by Yoon [22]. There are three variations of the MMY algorithm, namely *MMY*, *MMY1* and *MMY2*. The MMY procedure was designed to find a relaxed Pareto set, while MMY1 and MMY2 were devised to seek the true Pareto set without the *indifference-zone*

(IZ) concept and the true Pareto set with the IZ concept, respectively. For this study, the MMY procedure was selected for generating the relaxed Pareto set. This procedure was selected because both the MMY1 and MMY2 procedures are very strict with the inclusion of solutions into the Pareto set, while the MMY procedure would rather include the solutions where indifference is present.

Before the MMY procedure could be applied, the NSGAII was again applied to the three test scenarios created previously. The NSGAII then determined a set of fifty solutions, consisting of dominated and non-dominated solutions, for each test scenario. For each of these solutions the simulation model was run independently 1 000 times as a preliminary step in order to estimate the unknown true means of the two objectives as best as possible.

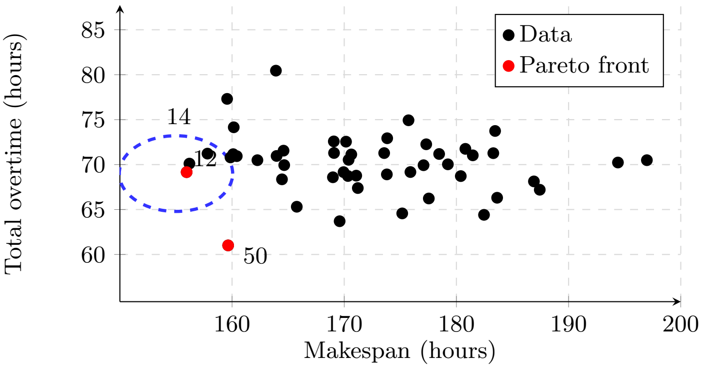


Figure 3: *The estimated true means of the 50 solutions of the 50 job test scenario.*

The estimated true means of the fifty solutions for the 50-job test scenario are presented in Figure 3 (as an example). When the MMY procedure was applied to the generated data, the IZ value for each objective was set to $\delta_1^* = 12$ hours and $\delta_2^* = 7$ hours, respectively, for the 50-job test scenario. The IZ values, δ_1^* and δ_2^*, are the ranges that are defined for each objective by the user of the system. If the objective values of two solutions are within the IZ value of that objective, then the solutions are considered to be indifferent to each other. The probability of correct selection $P(\text{CS})$ was chosen to be 0.95, and the first stage sample size was set to be $n_0 = 10$. When the true means in Figure 3 are examined, true Pareto set $Q = \{12{,}50\}$ is obtained. However, the MMY procedure identified that the Pareto set with IZ is $Q_{IZ} = \{12{,}14{,}50\}$. The relaxed Pareto set can therefore be defined as $Q_R = \{12{,}14{,}50\}$ for the 50-job test scenario. The total number of simulation replications assigned to the systems in Q_R were $N_{12} = 95$, $N_{14} = 95$ and $N_{50} = 75$. The observed members of the Pareto set were selected with a guaranteed probability of correct selection, due to Procedure MMY. The same process was repeated for the other test scenarios.

Table 3 provides the mean performance metric values for the different test scenarios after the MMY procedure was applied. The table also contains the number of simulation replications required to find the relaxed Pareto set while still fulfilling the probability of correct selection. The sets that were compared were the true Pareto set and the relaxed Pareto set. Considering these performance metrics, the GD value should be zero and the HR and MS values should be one for the relaxed Pareto set to be considered equal to the true Pareto set. The MMY procedure did perform well in all three test scenarios, as it

Test scenario	GD	HR	MS	Simulation replications
50 jobs	0.670	1.020	0.990	$N_{12} = 95$
				$N_{14} = 95$
				$N_{50} = 75$
100 jobs	3.744	1.000	0.990	$N_{1} = 408$
				$N_{4} = 406$
				$N_{32} = 408$
				$N_{43} = 392$
				$N_{46} = 408$
200 jobs	1.218	1.000	0.910	$N_{3} = 713$
				$N_{14} = 713$
				$N_{27} = 713$
				$N_{43} = 713$

Table 3: *Mean performance metric values for the MMY procedure and the different test scenarios.*

achieved small GD values compared to those in Table 1, as well as HR and MS values close to one in all three test scenarios. The procedure also determined the relaxed Pareto set while using considerably less simulation replications compared to the true Pareto set that required 1 000 replications per member.

5 Conclusion

The comparison of the performance of different metaheuristics to solve a stochastic bi-objective JSP were discussed. The results of these comparisons illustrated that the NSGAII performed best of the selected metaheurstics. This paper also contributes to the limited research that has previously been conducted to solve the JSP using strongly conflicting overtime and makespan as objectives. Furthermore, this paper discusses the first implementation of the MMY ranking and selection procedure on a stochastic JSP, to find the relaxed Pareto set.

References

[1] Adibi MA, Zandieh M & Amiri M, 2010, *Multi-objective scheduling of dynamic job shop using variable neighborhood search*, Expert Systems with Applications, **37**, pp. 282–287.

[2] Baykasoğlu D, Hamzadayi H & Köse SY, 2014, *Testing the performance of teaching–learning based optimization (TLBO) algorithm on combinatorial problems: Flow shop and job shop scheduling cases*, Information Sciences, **276**, pp. 204–218.

[3] Chaudhry IA & Khan AA, 2016, *A research survey: Review of flexible job shop scheduling techniques*, International Transactions in Operational Research, **23**, pp. 551–591.

[4] Dominic PDD, Kaliyamoorthy S & Kumar MS, 2004, *Efficient dispatching rules for dynamic job shop scheduling*, The International Journal of Advanced Manufacturing Technology, **24(1)**, pp. 70–75.

[5] French S, 1982, *Sequencing and scheduling: An introduction to the mathematics of the job shop*, Ellis Horwood, Chichester.

[6] Ganesh K & Anbuudayasankar SP, 2012, *International and Interdisciplinary Studies in Green Computing*, IGI Global, Hershey (PA).

[7] Gen M, Gao J & Lin L, 2009,*Multistage-based genetic algorithm for flexible job shop scheduling problem*, pp. 183–196, Springer, Berlin.

[8] Jensen MT, 2001, *Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measures*, Applied Soft Computing, [Online Serial], Available from `https://doi.org/10.1016/S1568-4946(01)00005-9`

[9] Kuhpfahl J, 2015, *Job shop scheduling with consideration of due dates: Potentials of local search based solution techniques*, Springer, Wiesbaden.

[10] McKay KN, Safayeni FR & Buzacott JA, 1988, *Job-shop scheduling theory: What is relevant?*, Interfaces, **18(4)**, pp. 84–90.

[11] Moritz RLV, Reich E, Schwarz M, Bernt M & Middendorf M, 2015, *Refined ranking relations for selection of solutions in multi-objective metaheuristics*, European Journal of Operational Research, **243(2)**, pp. 454–464.

[12] Pardalos PM, 2013, *Approximation and complexity in numerical optimization: Continuous and discrete problems*, Springer, New York (NY).

[13] Rohaninejad M, Khierkhah A, Fattahi P & Vahedi-Nouri B, 2015, *A hybrid multi-objective genetic algorithm based on the ELECTRE method for a capacitated flexible job shop scheduling problem*, International Journal for Advanced Manufacturing Technologies, **77**, pp. 51–66.

[14] Sha DY & Lin HH, 2010, *A multi-objective PSO for job-shop scheduling problems*, Expert Systems with Applications, **37**, pp. 1065–1070.

[15] Snyman S & Bekker J, 2017, *Real-time scheduling in a sensorised factory using cloud-based simulation with mobile device access*, South African Journal of Industrial Engineering, **28(4)**, pp. 161–169.

[16] Snyman S, Bekker J & Botha JP, 2018, *A real-time scheduling system for a sensorised job shop using cloud-based simulation with mobile device access*, Proceedings of the 29th SAIIE Annual Conference, Stellenbosch, pp. 165–177.

[17] Snyman S & Bekker J, 2019, *A real-time scheduling system in a sensorised job shop*, Proceedings of the 7th International Conference on Competitive Manufacturing (COMA'19), Stellenbosch, pp. 99–104.

[18] Weber E, Tiefenbacher A & Gronau N, 2019, *Need for standardization and systematization of test data for job-shop scheduling*, Data, **4(1)**, pp. 1–21.

[19] Wisner JD, 2016, *Operations management: A supply chain process approach*, SAGE Publications, New York (NY).

[20] Yang Y & Gu X, 2014, *Cultural-Based Genetic Tabu Algorithm for Multiobjective Job Shop Scheduling*, Mathematical Problems in Engineering, [Online Serial], Available from `https://doi.org/10.1155/2014/230719`

[21] Yen GG & He Z, 2014, *Performance metric ensemble for multiobjective evolutionary algorithms*, IEEE Transactions on Evolutionary Computation, **18(1)**, pp. 131–144.

[22] Yoon M, 2018, *New multi-objective ranking and selection procedures for discrete stochastic simulation problems*, PhD Thesis, University of Stellenbosch, Stellenbosch.

[23] Zhang J, Yang J & Zhou Y, 2016, *Robst scheduling for multi-objective job-shop problems with flexible workdays*, Engineering Optimization, **28(11)**, pp. 1973–1989.

[24] Zhang CY, Li PG, Rao YQ & Guan ZL, 2008, *A very fast TS/SA algorithm for the job shop scheduling problem*, Computers & Operations Research, **35**, pp. 282–294.

Proceedings of the 48th ORSSA Annual Conference
pp. 27–35
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3
©2019

Decision support for effective team selection in Fantasy Premier League

R Dykman[*] T Schmidt-Dumont[†]

Abstract

Fantasy Premier League is a popular online sports prediction game, based on the English Premier League, in which a user forms an imaginary team composed of real-world soccer players. In this paper, a decision support system capable of accurately predicting player performances and subsequently suggesting team selections such that a manager's cumulative points score may be maximised is proposed. This decision support system is based on an integer programming approach. The effectiveness of the proposed system is evaluated in the context of a real-world case study, comparing system performance to that of real-world players. With the aid of the system, a player would have finished in the top 6.7% of players worldwide during the 2017/18 season.

Key words: Integer programming, forecasting, sports.

1 Introduction

A fantasy sports game is a form of competition that is usually played in an online manner in which each competitor forms an imaginary team composed of real-world players. Each competitor, usually referred to as a *manager*, receives points based on the real-world performances of the players in his/her team. Fantasy sports games have become an immensely popular pastime activity, and it is common for a group of friends or co-workers to compete against one another in a league that is privately run. *Fantasy Premier League* (FPL), which is based on the English Premier League, is the largest fantasy competition worldwide boasting more than 5.9 million players in the 2017/18 season. Due to the immense popularity of FPL, cash and prize incentives are offered to players that perform well in the competition.

In order to be able to compete at the highest level in FPL, effective team selection decisions must be taken week after week. These decisions are highly constrained and must take into account both short- and long-term rewards in terms of the cumulative points gain.

[*]Department of Industrial Engineering, Stellenbosch University, Private bag X1, Matieland, 7602, Republic of South Africa, email: rudi.dykman@gmail.com

[†]Department of Industrial Engineering, Stellenbosch University, Private bag X1, Matieland, 7602, Republic of South Africa, email: thorstens@sun.ac.za

Currently there are 533 players in the English Premier League that can be selected in more than 10^{25} different team configurations.

Each player is assigned one of four playing positions, based on their real world position, together with a monetary value in *Great British Pounds.* The positions are *Goalkeeper, Defender, Midfielder* and *Forward.* At the start of each season, each FPL manager must select a squad of 15 players, consisting of exactly 2 Goalkeepers, 5 Defenders, 5 Midfielders and 3 Forwards, with the total monetary value of the squad not exceeding £100 million [3]. Furthermore, no more than three players from any one club may be selected.

All matches in the current FPL season are split into thirty-eight rounds called *gameweeks* (GWs). During any specific GW, each team typically plays one match, although there may be exceptions. Between GWs, FPL managers are free to purchase and sell players. Managers are granted one free transfer per GW, while any additional transfers are penalised with a four-point subtraction from the manager's cumulative score [3]. If the free transfers are not used by a manager, these can accumulate to a maximum of two.

In order to succeed in FPL, it is of utmost importance to select the right players at the right time. Players are subject to form, which refers to the current level of performance of a player. In FPL, *form* is defined as a player's average score per match, calculated over the past thirty days [3]. The difficulty of an upcoming fixture may also have a significant effect on the points score achieved by a player, as a striker is more likely to score, and thus achieve a high score, when playing against a weak opponent. These factors thus have to be taken into account when making effective team selection decisions.

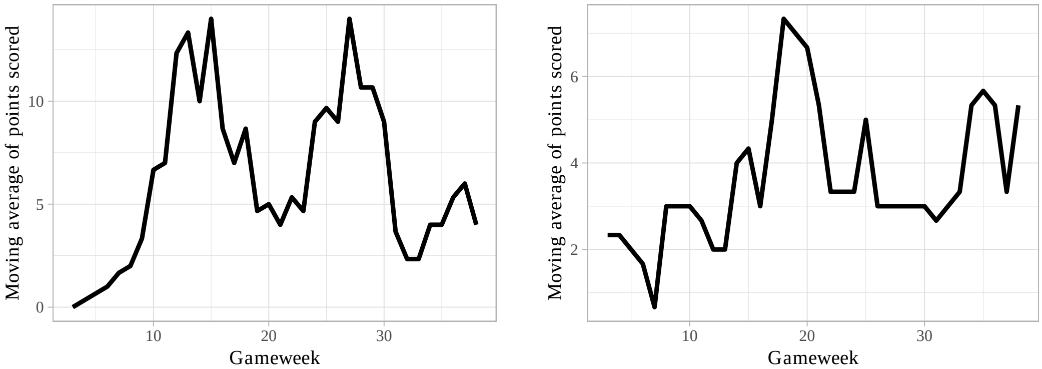
2 Literature review

Fantasy sports leagues have drawn the attention of operations researchers around the world in pursuit of optimal team selections. Limited amounts of work have, however, focused on FPL specifically. Although not directly aimed at FPL, Bonomo *et al.* [2] developed a mathematical programming approach towards optimal team selection in *Gran DT*, a fantasy league based on the Argentinian professional soccer league. They did not, however, detail the processes whereby player points forecasting was performed.

In a rare attempt at playing FPL specifically, Matthews *et al.* [9], developed a *reinforcement learning* (RL) methodology with the aim of learning optimal team formations. In their implementation, a belief state Markov decision process — based on a Bayesian belief model of player abilities — was employed in order to simulate games and their outcomes. Player points were estimated based on the simulated outcomes. The RL algorithm would then be employed to suggest sets of players and possible transfers with the aim of maximising the points return. As a result of employing this approach, reported performance placed them in the top 2.5% of players worldwide. The authors of this paper, however, believe that similar performance may be achieved with a significantly simpler modelling approach based on points forecasting and integer programming for team selection.

3 Player points forecasting

As may be seen in Figure 1, no clear trend or seasonality emerges when a moving average of the number of points scored per GW is plotted for a random sample of two FPL players. This lack of seasonality was confirmed by performing the Webel-Ollech test [13] for seasonality, yielding an average p-value of 4.8392×10^{-1} when performed for all players. This characteristic poses a serious challenge, as many forecasting techniques, such as Holt & Winter's method [4], cater specifically for these characteristics. It was therefore decided to aim to achieve the best possible results with simple forecasting techniques.



Eden Hazard (Midfielder) Dejan Lovren (Defender)

Figure 1: *A plot of the moving average of points scored in the last three games by two players in the 2017/18 EPL season, not showing any sign of seasonality over consistent time periods.*

3.1 Time-series forecasting

Three time-series based approaches were considered in this study [8]. These are (1) the *average* of points achieved in all prior GW's, (2) the *moving average* of points achieved by a player during the last k GWs, and (3) *exponential smoothing* in order to attain a points prediction [8]. The best performing value of k in the moving average implementation was empirically determined as 5. The points prediction for player i is therefore simply the average number of points scored during the previous five GWs. In the exponential smoothing implementation, the points prediction is given by

$$F_{i,T} = F_{i,T-1} + \alpha e_{i,T-1}, \tag{1}$$

where α denotes the smoothing factor, and $e_{i,T-1}$ denotes the forecast error during the previous GW, $T-1$. The best-performing value of α was determined empirically as 0.8.

3.2 Explanatory forecasting

Various regression approaches were employed for determining the relationship between various independent variables and the points scored by each player. These variables include the *average* of points scored in all GWs to date, the *moving average* of points achieved in the past five GWs, the points score achieved in the previous GW, a player's so-called

Influence, Creativity and Threat (ICT) index, the upcoming *Fixture Difficulty Rating* (FDR), and the net number of transfers of a player into or out of squads by all human managers. The ICT index and the FDR are metrics published by the FPL organisers and pertain to a player's ability to influence the outcome of a game, provide assistance to teammates, and a player's threat on goal, while the FDR provides a measure of the difficulty of the upcoming fixture based on the opponent [3].

To take multiple variables into account, *multiple linear regression* (MLR) [7] was implemented. Employing a backward elimination [11] approach, it was determined that all variables do, in fact, have a statistically significant influence on the points prediction for a player. The points forecast by MLR is then given by

$$F_{i,T} = b_0 + b_1 a_{i,T} + b_2 \mathrm{ma}_{i,T} + b_3 p_{i,T-1} + b_4 \mathrm{ICT}_{i,T} + b_5 \mathrm{FDR}_{i,T} + b_6 \mathcal{T}_{i,T}, \tag{2}$$

where $a_{i,T}$ denotes player i's average number of points until time T, $\mathrm{ma}_{i,T}$ denotes player i's moving average of points over the past five GWs leading up to time T, $p_{i,T-1}$ denotes player i's points score during the previous GW, $\mathrm{ICT}_{i,T}$ denotes player i's ICT index at time T, $\mathrm{FDR}_{i,T}$ denotes player i's FDR for the upcoming GW, and $\mathcal{T}_{i,T}$ denotes the number of net transfers of player i into and out of managers' teams between GWs $T-1$ and T. A summary of the best-performing constants, $b_0, \ldots, b_6$, may be found in Table 1.

	Beta value	Std. Error	t-statistic	p-value
(Intercept)	9.6551×10^{-1}	4.9233×10^{-2}	1.9611×10^{1}	6.7128×10^{-85}
Moving Average	3.1400×10^{-1}	1.5591×10^{-2}	2.0140×10^{1}	2.1360×10^{-89}
Average	7.2384×10^{-2}	1.9961×10^{-2}	3.6262×10^{0}	2.8825×10^{-4}
FDR	-2.1717×10^{-1}	1.5265×10^{-2}	-1.4226×10^{1}	1.0101×10^{-45}
Previous GW Points	7.8639×10^{-2}	8.4542×10^{-3}	9.3018×10^{0}	1.5077×10^{-20}
ICT Index	2.4846×10^{-1}	1.2923×10^{-2}	1.9227×10^{1}	1.0499×10^{-81}
Net Transfers	1.5088×10^{-6}	4.2180×10^{-7}	3.5771×10^{0}	3.4819×10^{-4}

Table 1: *The results of the ANOVA of the multiple linear regression model using all indicators. (A p-value smaller than 0.05 indicates statistical significance.)*

Apart from MLR, *Decision Tree Regression* (DTR) and *Random Forest Regression* (RFR) were also implemented [10]. For these implementations, the standard implementations as in the R packages rpart [12] and randomForest [6] were employed. For the RFR, it was found that increasing the number of trees to above 500 did not yield statistically significant improvements at a 5% level of significance with respect to the forecasting accuracy achieved. Therefore, 500 trees were employed in this study.

A statistical comparison of the different forecasting techniques was performed using the player points data from the 2017/18 season. As may be seen in Figure 2, exponential smoothing achieved the best accuracy. Its performance was, however, found to be statistically similar at a 5% level of significance to that of MLR, DTR and RFR. Exponential smoothing, MLR, DTR and RFR were, however, able to outperform both the average and moving average methods for forecasting player points values at a 5% level of significance.

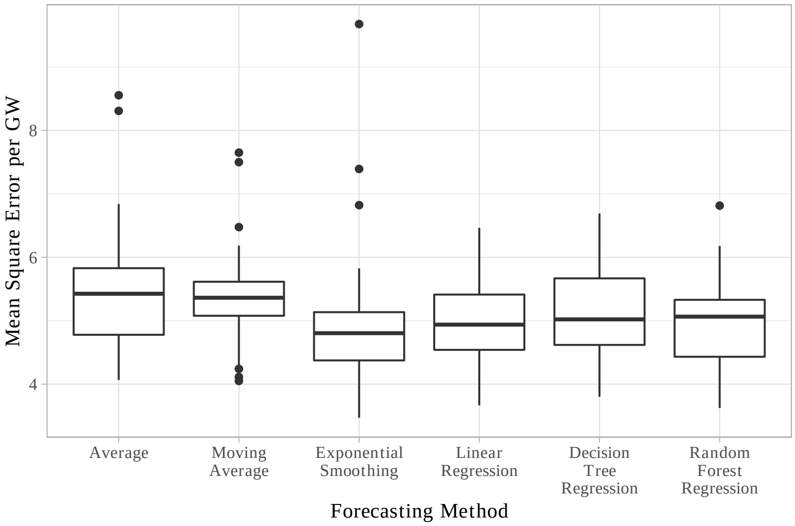


Figure 2: *A box plot illustrating the difference in performance between the various forecasting methods using data from the 2017/18 FPL season.*

4 A mathematical approach towards team selection

A three pronged modelling approach is employed within the DSS. The first *integer program* (IP) is solved in order to generate an initial squad at GW = 1 with which to start the season. A second IP is then solved periodically in order to suggest valid transfers for each GW $\in [2, \ldots, 38]$. Finally, a third IP is solved in order to select the starting team for each GW. The R package lpSolve [1], which employs a Branch and Bound method for solving IPs to optimality, is employed to solve the IPs. When solving the IPs, a single forecast points value is employed for each player for each GW to solve the IP to optimality. At the start of an FPL season, a manager is required to select his/her entire squad, filling all positions in the process.

The objective when selecting the initial squad is to

$$\text{maximise } f(x) = \sum_{i=1}^{\mathcal{I}} p_i x_i, \tag{3}$$

where

$$p_i = \begin{cases} \text{total score for player } i \text{ in the previous season if GW} = 1, \text{ or} \\ \text{the forecasted points value for player } i, \end{cases} \tag{4}$$

$$x_i = \begin{cases} 1 & \text{if player } i \text{ is selected, and} \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

and $\mathcal{I}$ denotes the set of all players i. The maximisation is performed subject to the constraints

$$\begin{aligned}
\sum_{i=1}^{\mathcal{I}} c_i x_i &\leq 100, && && (6)\\
\sum_{i=1}^{\mathcal{I}} x_i &= 2 && \text{where } i \in \mathcal{G}, && (7)\\
\sum_{i=1}^{\mathcal{I}} x_i &= 5 && \text{where } i \in \mathcal{D} \text{ and } i \in \mathcal{M}, && (8)\\
\sum_{i=1}^{\mathcal{I}} x_i &= 3 && \text{where } i \in \mathcal{F}, && (9)
\end{aligned}$$

and

$$\sum_{j=1}^{\mathcal{J}} \sum_{i=1}^{\mathcal{I}} t_{i,j} x_i \leq 3, \tag{10}$$

where $\mathcal{J}$ denotes the set of all teams j, $\mathcal{G}, \mathcal{D}, \mathcal{M}$ and $\mathcal{F}$ denote the sets of Goalkeepers, Defenders, Midfielders, and Forwards, respectively, and c_i denotes the cost of player i. Finally,

$$t_{i,j} = \begin{cases} 1 & \text{if player } i \text{ plays for team } j \text{, and} \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Constraint (6) is included to ensure that no more than the initial budget of £100 million is spent. To ensure that all required positions are filled according to the FPL guidelines set out in §1, constraints (7)–(9) are included. Finally, constraint (10) ensures that no more than three players from any one club are included in the team.

The objective function for the transfer suggestion model is modified so as to

$$\text{maximise } f(x) = \sum_{i=1}^{\mathcal{I}} p_{i,g} x_{i,g} - q_g, \tag{12}$$

where q_g denotes the points penalty deducted in GW g if the number of transfers made exceeds the number of available free transfers. In addition to the constraints (7)–(10), the following constraints are implemented:

$$-q_g + 4(t_g - f_g) \leq 56y, \tag{13}$$

$$t_g - f_g \leq 56(1 - y), \tag{14}$$

$$15 - \sum_{i=1}^{\mathcal{I}} x_{i,g-1} x_{i,g} = t_g, \tag{15}$$

$$f_g, t_g \in \mathbb{Z}, \tag{16}$$

$$q_g, f_g, t_g \geq 0, \quad \text{and} \tag{17}$$

$$f_g \leq 2, \tag{18}$$

where t_g denotes the number of transfers made in GW g, f_g denotes the number of free transfers available during GW g, and y denotes a binary variable employed to implement the if-then constraints (13)–(14). Constraint (15) is included to determine the value of t_g, while f_g is an input parameter read from the FPL database. Constraint (15) may give the illusion of non-linearity, but as the model is solved periodically, $x_{i,g-1}$ is, in fact, an input parameter during GW g.

Finally, the model for selecting the starting eleven during each GW is presented. The objective now is to

$$\text{maximise } f(s)\sum_{i=1}^{\mathcal{I}} p_{i,g}x_{i,g}s_{i,g}, \tag{19}$$

where

$$s_{i,g} = \begin{cases} 1 & \text{if player } i \text{ is in the starting eleven during GW } g\text{, and} \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

The set of players included in the current squad $x_{i,g}$ is now available as input parameter. The maximisation is constrained by

$$\sum_{i=1}^{\mathcal{I}} s_{i,g} = 11, \tag{21}$$

$$\sum_{i=1}^{\mathcal{I}} G_i, s_i = 1, \tag{22}$$

$$\sum_{i=1}^{\mathcal{I}} D_i s_i \geq 3, \tag{23}$$

and

$$\sum_{i=1} F_i, s_i \geq 1 \tag{24}$$

to ensure all positions are filled. Finally, the manager may assign captaincy and vice-captaincy to two of his/her players. In any GW, the captain's points are doubled, and in cases where the captain did not play, the vice-captain's points are doubled. The captain and vice-captain are simply those players with the highest and second-highest points forecasts.

5 Numerical results

The data from the completed 2017/18 season were employed in order to assess the performance of the developed system. These data may be extracted directly from the FPL database using the fplR package [5]. The results achieved are summarised in Table 2.

Adjusting the time-series predictions with the FDR resulted in significant improvements in respect of the points achieved over the course of a season. The forecast points values are adjusted according to

$$\rho_{i,g} = \frac{p_{i,g}}{a(FDR_{i,g})}, \tag{25}$$

where a denotes a weighting factor to control the impact that the FDR has on a forecast points value. This formulation, as well as the best-performing values for the weighting factor a, was empirically determined. It was found that the best-performing values for a were 1.4, 0.7 and 1.4 for the Average, Moving Average and Exponential Smoothing methods, respectively. With the adjusted points forecasts, these methods achieved scores of 2 021 points, 2 028 points and 2 119 points, respectively. The score of 2 119 achieved with Exponential Smoothing would have resulted in the manager finishing in 393 762th place, finishing in the top 6.7% worldwide.

Forecasting method	Score	Rank	Percentile
Average	1 847	2 446 932	41.40%
Moving Average	1 820	2 653 841	44.90%
Exponential Smoothing	2 028	960 971	16.26%
Multiple Linear Regression	2 109	445 014	7.53%
Decision Tree Regression	1 788	2 906 664	49.18%
Random Forest Regression	1 756	3 145 659	53.22%

Table 2: *The final score and rank obtained using the different forecasting techniques for the 2017/18 season.*

6 Discussion

Although the performance of the system is not quite on par with that achieved by Matthews *et al.* [9], the results achieved are promising, with the points forecasting proving the problematic area. If the ICT index achieved by a player during a specific GW were available before the GW, and thus could be incorporated in the forecast (drastically improving the prediction accuracy), the system could have achieved a score of 2 512, which would have resulted in the manager comfortably winning the FPL season. This provides validation of the mathematical modelling approach while highlighting that forecasting accuracy is critical for system performance improvement. Improving on the forecasting accuracy is the focus in an ongoing project on decision support for FPL at Stellenbosch University.

References

[1] Berkelaar M, 2019, *lpSolve: Interface to 'Lp_solve' v. 5.5 to solve linear/integer programs*, R package version 5.6.13.1, [Online], [Cited July 22nd, 2019], Available from `https://cran.r-project.org/web/packages/lpSolve/index.html`

[2] Bonomo F, Durán G & Marenco J, 2014, *Mathematical programming as a tool for virtual soccer coaches: A case study of a fantasy sports game*, International Transactions in Operations Research, **21**, pp. 399–414.

[3] Fantasy Premier League, [Online], [Cited April 5th, 2019], Available from `https://fantasy.premierleague.com/`

[4] Gardner ES, 2006, *Exponential smoothing: The state of the art — Part II*, International Journal of Forecasting, **22(4)**, pp. 637–666.

[5] Henderson E, 2018, *fplR: An R package that provides a compendium of tools for working with-Fantsasy Premier League (FPL) data in R*, R Package version 0.0.0.9, [Online], [Cited July 22nd, 2019], Available from `https://github.com/ewenme/fplr`

[6] Liaw A & Wiener M, 2002, *Classification and regression by randomForest*, R News, **2(3)**, pp. 18–22.

[7] McDonald JH, 2009, *Handbook of biological statistics*, Sparky House Publishing, Baltimore (MD).

[8] Makridakis SG, Wheelwright SC & Hyndman RJ, 1998, *Forecasting: Methods and applications*, John Wiley & Sons, New York (NY).

[9] Matthews T, Ramchurn S & Chalkiadakis G, 2012, *Competing with humans at fantasy football: Team formation in large partially-observable domains*, Proceedings of the 26th AAAI Conference on Artificial Intelligence, Toronto, pp. 1394–1400.

[10] Speybroeck A, 2016, *Classification and regression trees*, International Journal of Public Health, **57(1)**, pp. 243–246.

[11] Sutter JM & Kalivas JH, 1993, *Comparison of forward selection, backward elimination, and generalized simulated annealing for variable selection*, Microchemical Journal, **47(1–2)**, pp. 60–66.

[12] Therneau T & Atkinson B, 2018, *rpart: Recursive partitioning and regression trees*, R package version 4.1–13, [Online], [Cited April 12th, 2019], Available from `https://CRAN.R-project.org/package=rpart`

[13] Webel K & Ollech D, *An overall seasonality test based on recursive feature elimination in conditional random forests*, Proceedings of the 4th International Conference on Time Series and Forecasting, Granada, pp. 20–31.

Proceedings of the 48th ORSSA Annual Conference
pp. 36–44
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3
©2019

Decision support for the design of bus routes in a public transit system

G Hüsselmann*   JH van Vuuren†

Abstract

The problem considered in this paper is that of providing decision support for aiding a bus company in deciding upon efficient bus routes. The design criteria adopted are the simultaneous pursuit of minimising the average passenger travel time and minimising the system operator's cost (measuring the latter as the sum total of all route lengths in the system). This bi-objective minimisation problem is solved approximately in three distinct stages — a solution initialisation stage, an intermediate analysis stage and an iterative metaheuristic search stage during which high-quality trade-off solutions are sought. The model takes as input an origin-destination demand matrix for a fixed set of bus stops, along with the corresponding road network structure, and returns as output a set of potential bus route solutions. The decision maker can then select one of these route sets based on the desired trade-off between the aforementioned transit system design criteria. The model is applied to a special case study involving the Matie Bus service provided by Stellenbosch University to its students.

Key words: Simulated annealing, strategic planning, transportation, vehicle routing.

1 Introduction

Transport is increasingly becoming an urgent problem in urban areas. The need for, and possibilities of, transport seems to be growing, and this results in increasing travel times for commuters [11]. There are different approaches towards alleviating traffic congestion problems. One such approach is to encourage the use of public transport — the situation where more people are transported in fewer vehicles.

In the context of Stellenbosch University, a very unique problem arises in this general context. The town was originally designed for a small population, but with the ever-increasing size of the university population and the town's increasing attractiveness from a tourism perspective, transport demand has also increased in recent years [9]. This

*Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602, Republic of South Africa, email: email: 17832020@sun.ac.za

†**(Fellow of the Operations Research Society of South Africa)**, Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602, Republic of South Africa, email: vuuren@sun.ac.za

problem induces considerable frustration in terms of traffic congestion, parking deficits, illegal parking, long walking distances to and from parking spaces and the time delays associated with each of these. These challenges may, of course, be addressed to some extent by the introduction of adequate public transport [9]. The *Matie Bus* is a service initiative that has been launched by Stellenbosch University for this purpose [15]. The service caters for students' transportation needs. Some questions have, however, arisen in respect of the effectiveness of this service.

The problem considered in this paper is that of providing decision support to the Matie Bus service in terms of deciding upon a redesign of the bus routes in its transit system. A mathematical model is derived for this purpose and takes as input an *origin-destination* (OD) travel-pairs demand matrix of prospective passengers, and produces as output a set of potential routes for the buses. The decision maker is able to specify the desired trade-off pursued between minimising passenger travel time and operator cost in route set recommendations produced by the model.

The three remaining sections of this paper are organised as follows. Section 2 is devoted to a brief literature review. In §3, a mathematical model of the *urban transit routing problem* (UTRP) is derived, validated and verified. Finally, §4 is a special case study in which the model of §3 is applied to the route design problem for the Matie Bus service.

2 Literature review

The global transit planning process requires certain inputs, such as passenger demand, the area underlying the transport network (including its topological characteristics), the available transport vehicles and the drivers of these vehicles. The end goal is to establish a set of transport routes or lines together with their associated timetables [8]. According to Ceder and Wilson [3], the planning process for transit networks can, in general, be partitioned into a sequence of five distinct steps. These steps are network design, frequency setting, timetable development, bus scheduling and driver scheduling. The UTRP pertains to the first of these steps only.

In the UTRP, the goal is to find a configuration of various transit routes that minimises or maximises a pre-selected set of objective functions subject to a set of constraints [2]. The UTRP requires design choices to be made from a set of alternatives, and can often be modelled in the form of integer or mixed integer programming models [8]. Moreover, various trade-offs typically have to be considered that mainly affect the passengers and operators of the transport network. Finding a suitable trade-off between the design objectives poses a significant challenge [8].

In the literature, however, mathematical models of the UTRP have primarily been concerned with minimising an overall cost function, typically in the form of a linear combination of passenger and operator costs [2]. According to Baaj and Mahmassani [1], the most popular objective function considered in UTRP models takes the form of the weighted sum of the passenger *average travel time* (ATT) and operator *total route time* (TRT), both measured in units of time.

The UTRP is a variation on the well-known vehicle routing problem in the operations research literature [16]. Furthermore, the UTRP has been established as an NP-hard, mixed combinatorial optimisation problem [1]. For this reason, the exact solution of problem instances is unattainable in almost all cases according to Baaj and Mahmassani [1]. Solution approaches applied to instances of the UTRP in the literature can be partitioned into three classes: Practical guidelines or *ad hoc* procedures, analytical optimisation methods for idealised situations, and metaheuristic approaches for more practical problems [1, 7].

When dealing with transit networks of realistic sizes, where many variable values have to be determined, the aforementioned practical guidelines and analytical approaches do not work well. In such cases, the inherent complexity involved in instances of the UTRP call for methods that can deal effectively with the larger sizes of realistic transit networks. For this reason, metaheuristics have often been applied to UTRP instances [7, 8].

(Meta)heuristic solution methods applicable to the UTRP can be classified into four main families according to Guihaire and Hao [8]. These are specific and *ad hoc* heuristics that often follow a greedy construction principle, neighbourhood searches (such as *simulated annealing* (SA) and tabu search), evolutionary searches (such as genetic algorithms), and hybrid searches in which two or more solution methods are combined.

3 Mathematical model for the UTRP

Each vehicle is assumed to traverse back and forth along its assigned route, reversing its direction each time a terminal vertex is reached [10]. The inconvenience caused to passengers when having to transfer from one vehicle (route) to another is represented by a fixed constant transfer penalty of 5 min for each transfer incurred. This value is in line with assumptions in previous studies [4].

The following assumptions, due to Mumford [12], are also made in this paper. The passenger demand, expected travel time and distance matrices associated with the UTRP are assumed to be symmetric. The passenger demand exhibits a many-to-many pattern, but is fixed over time. The total travel time of a passenger comprises only in-vehicle travel time and transfer penalties. Each passenger's route choice is assumed to be based on the shortest expected travel time, regardless of the number of transfers required. The vehicle stops have already been determined, as is the case in most UTRP instances, and the aim is only to design a vehicle route set servicing these stops. An OD demand matrix is specified in terms of these vehicle stops. It is assumed that zero, one or two transfers between bus routes are acceptable for passengers, and whenever a passenger needs to make more than two transfers in order to travel to his/her destination, the situation is considered as unmet demand.

The ATT per passenger and the total route traversal time are the transport system performance measures minimised simultaneously within the modelling approach adopted in this paper, because the aim is to vary bus routes so as to reduce the cost of inconvenience to passengers and the route maintenance cost to the operator. John *et al.* [10] proposed a mathematical model formulation for the UTRP based on graph theoretic concepts that is more intuitive than earlier integer programming model formulations for the UTRP. Their general modelling paradigm is adopted in this paper.

Suppose the road network in respect of which the UTRP must be solved is modelled by an edge-weighted graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is a set of vertices representing the required transit vehicle stops and $\mathcal{E} = \{e_1, \ldots, e_m\}$ is a set of edges, each of which is incident with two vertices in $\mathcal{V}$. These edges represent shortest-time direct road links between pairs of vertices in G (*i.e.*, not containing any intermediate transit vehicle stops). The weights of the edges of G are specified in the form of an $n \times n$ *weight matrix* $\boldsymbol{W}$ whose entry in row i and column j denotes the expected travel time along the direct road link represented by an edge joining v_i and v_j if there is such an edge present in G, or ∞ otherwise. An $n \times n$ *demand matrix* $\boldsymbol{D}$ is also associated with G, which contains as entry in row i and column j the passenger demand between vertices v_i and v_j.

A *transit route* is defined as a simple path in G, represented by an ordered sequence of distinct vertices of G, each successive pair of which are adjacent in G. A transit route therefore contains no loops or repeating vertices. Let $G_R = (\mathcal{V}_R, \mathcal{E}_R)$ be the subgraph induced in G by the vertices in a transit route R. A set $\mathcal{R}$ of transit routes is considered an infeasible solution to the UTRP unless each vehicle stop is contained in at least one route $R \in \mathcal{R}$ [10]. Stated mathematically, the transit route set $\mathcal{R}$ is considered infeasible unless

$$\bigcup_{R\in\mathcal{R}} \mathcal{V}_R = \mathcal{V}. \tag{1}$$

A minimum and maximum number of vehicle stops is also specified per route in the transit route set $\mathcal{R}$ [10]. That is, it is required that

$$m_{\min} \leq |\mathcal{V}_R| \leq m_{\max}, \qquad R \in \mathcal{R}. \tag{2}$$

It is, of course, also required that the subgraph of G induced by the vertices of all the routes in a feasible set of transit routes be a connected graph [10]. That is,

$$G_{\mathcal{R}} = \left(\bigcup_{R\in\mathcal{R}} \mathcal{V}_R, \bigcup_{R\in\mathcal{R}} \mathcal{E}_R \right) \text{must be connected.} \tag{3}$$

The cardinality of the required set of transit routes is finally also specified. That is, a requirement of the form

$$|\mathcal{R}| = r \tag{4}$$

is specified for some natural number r.

In general, passengers prefer to travel to their destinations in the shortest possible time while avoiding the inconvenience of having to make transfers if this is possible. The length of a shortest path for an OD pair $v_i, v_j \in \mathcal{V}$ along a transit route set $\mathcal{R}$, denoted by $\alpha_{v_i,v_j}(\mathcal{R})$, is measured in units of time and can be computed by applying Dijkstra's algorithm to the subgraph $G_{\mathcal{R}}$ of G. This shortest path length includes both transport time and transfer time (if transfers between different routes in $\mathcal{R}$ are required). Mumford [12] measured the passenger (time) cost associated with a route set $\mathcal{R}$ as the ATT for all passengers, measured in units of time, which leads to the ojective of

$$\text{minimising } f_1(\mathcal{R}) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{D}_{v_i,v_j}\alpha_{v_i,v_j}(\mathcal{R})}{\sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{D}_{v_i,v_j}}, \tag{5}$$

where the numerator represents the total travel time of all passengers and the denominator represents the total number of passengers in the system.

A simple proxy was proposed by Mumford [12] for operator cost. This is the sum of the TRT along each route (in one direction), and results in the objective of

$$\text{minimising } f_2(\mathcal{R}) = \sum_{R \in \mathcal{R}} \sum_{(v_i,v_j) \in R} \boldsymbol{W}_{v_i,v_j}. \tag{6}$$

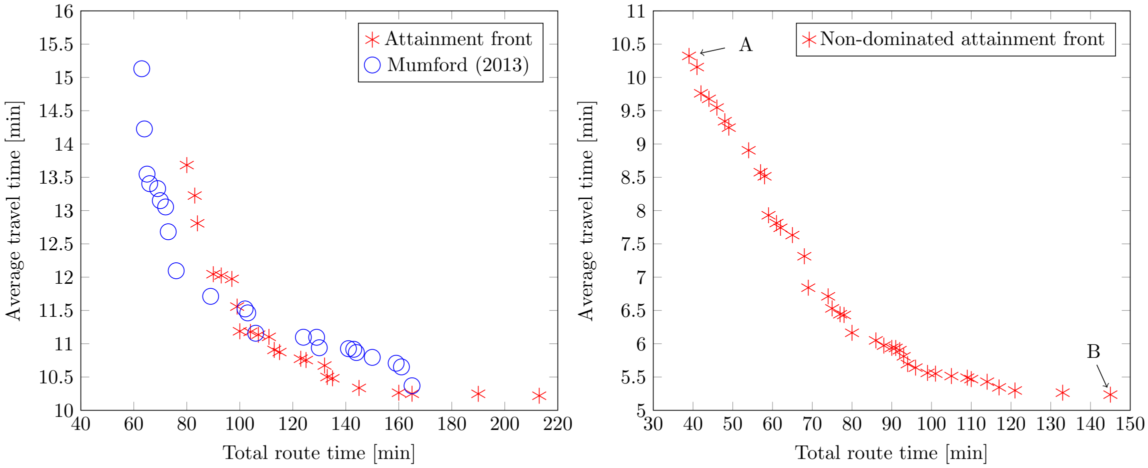
When solving the above model for the UTRP, both objective functions (5)–(6) are minimised simultaneously in order to establish a Pareto front representing effective trade-offs between minimising passenger costs and minimising operator costs. The UTRP solution methodology proposed by Fan and Machemehl [6, 7] consists of three main components, namely an *initial candidate route set generation procedure* (ICRSGP), a *network analysis procedure* (NAP), and an SA algorithm-based meta-heuristic search that combines the ICRSGP and NAP. Fan and Machemehl [7] have found that SA outperformed a genetic algorithm with respect to the quality of route sets achieved in the UTRP. SA was therefore selected as the solution method implemented in this paper due to its simplicity and the high-quality results that it reportedly achieves, providing near-optimal results within very reasonable time frames [6].

The ICRSGP iteratively generates r random routes, containing between $m_{\min}$ and $m_{\max}$ (inclusive) nodes each, until constraints (1)–(4) are satisfied, yielding an initial feasible solution $\mathcal{R}$. This feasible solution to the UTRP model (1)–(6) is then taken as the starting point for the NAP, during which the objective function values $f_1(\mathcal{R})$ and $f_2(\mathcal{R})$ associated with a transit route set $\mathcal{R}$ are assessed [7]. For this purpose, the graph $G_{\mathcal{R}}$ has to be analysed so that values for $\alpha_{v_i,v_j}(\mathcal{R})$ may be determined for all OD pairs $v_i, v_j \in \mathcal{V}$ in terms of the weight matrix $\boldsymbol{W}$. The SA algorithm employed here incorporates the ICRSGP and NAP, and guides the candidate solution generation procedure within an iterated search framework [6, 7]. The particular version of SA implemented in this paper is the dominance-based multi-objective SA algorithm proposed by Smith *et al.* [13]. After extensive parameter evaluation, the geometric cooling schedule was selected with a cooling rate of 0.95. The minimum number of accepted moves per search epoch was set to four, and the maximum number of epochs that may elapse without the acceptance of any new solutions was also set to four (the stopping criterion). The maximum allowable number of iterations per epoch was set to 700.

Mumford [12] published results for Mandl's well-known Swiss network UTRP benchmark problem obtained by an evolutionary algorithm. The Pareto set approximation of Mumford is presented graphically in Figure 1(a) together with the attainment front obtained by twenty different initialisations of our SA search process. As may be seen in the figure, the SA search procedure produces competitive results with the evolutionary algorithm of Mumford.

4 Matie Bus case study

Problems that arise with the implementation of the fixed Matie Bus routing system, shown in Figure 2, include the fact that the routes do not meet students' destination requirements



(a) Model and Mumford [12] comparison

(b) Matie Bus case study attainment front

Figure 1: *(a) Attainment front for sets containing six routes together with the 2013 results of Mumford [12] for the Swiss network UTRP benchmark instance of Mandl [11], and (b) the newly proposed approximate Pareto set of solutions for the Matie Bus case study.*

sufficiently, that the buses do not have enough capacity to satisfy transport demand, and that the buses are not at the required locations at the required times. It has been suggested that an effective redesign of the routes of the bus service may lead to a larger portion of students potentially being serviced. This section is devoted to a case study on the Matie Bus service with a view to propose an alternative set of high-quality bus routes that may be implemented in service of a larger portion of the student population.

The accuracy of OD demand data is crucial for the design of a transit system aimed at meeting the demand of passengers effectively. These data are usually estimated, but with the emerging technology of Bluetooth and Wi-Fi sensors [5], it is nowadays possible to generate accurate OD demand matrices. The *Stellenbosch Smart Mobility Lab* [14] has launched a project to install such sensors across the campus of Stellenbosch University that will enable this type of data gathering, within ethical boundaries. The data are expected to be available during 2019 and may then be used in conjunction with the model of §3 to design bus routes that can effectively meet the demand of the students. Due to these data not being available at present, an estimate of the OD demand matrix was made which maps out the anticipated demand patterns of Stellenbosch University students across campus. The route sets uncovered by the model of §3 are displayed in objective function space in Figure 1(b).

The extremal points, depicted by A and B in Figure 1(b), represent respectively the best results obtained by the model from the operator perspective, denoted by A, and from the passengers' perspective, denoted by B. The objective function values, route decomposition and the graphical depiction of these two route sets superimposed on a map of Stellenbosch can be seen in Figure 3. Figure 3(a) represents the best route set from the operator perspective, due to the TRT being a minimum (therefore requiring to maintain shorter routes), but this comes at a trade-off in that passengers will incur a longer ATT due to there being fewer routes to choose from for their respective trips. Figure 3(b), on the

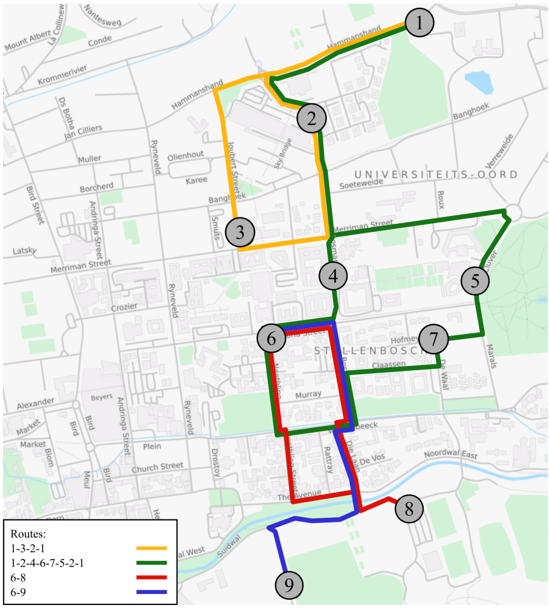
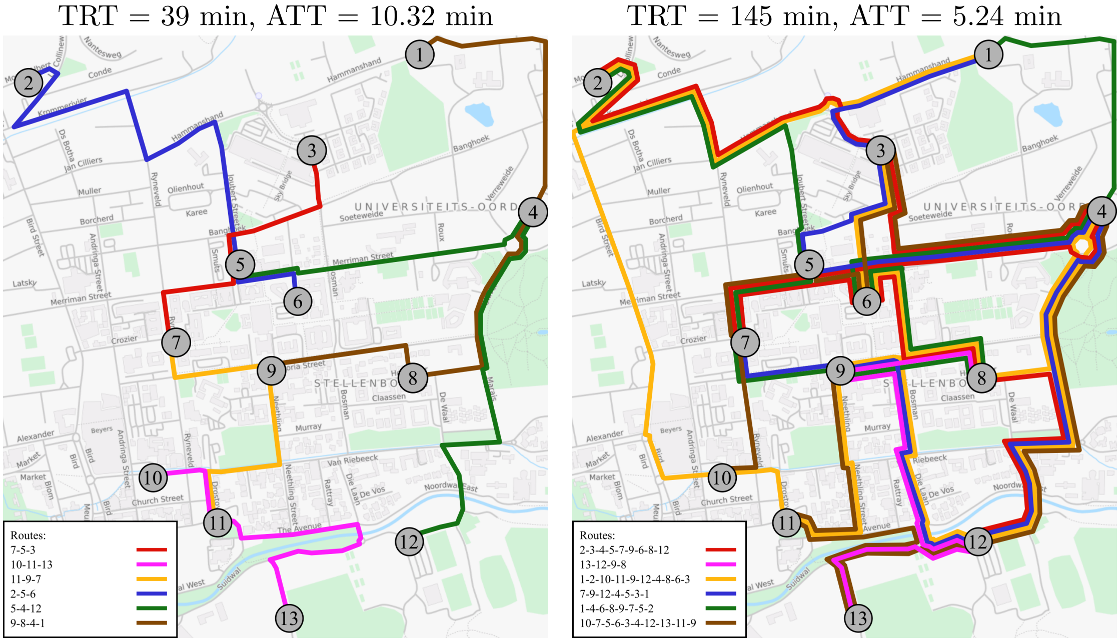


Figure 2: *The current Matie Bus routes for the day shuttle service [15].*



(a) Solution A

(b) Solution B

Figure 3: *The best route sets from the perspective of (a) the operator and (b) passengers returned by the SA algorithm of §3 for the Matie Bus case study.*

other hand, represents the best route set from the passengers' perspective, yielding the minimum ATT, due to more route accesibility and the presence of more direct routes without incurring transfers. This gives more flexibility to the transit system, but at the trade-off of longer routes for the operators to maintain. The new route sets and the current Maties Bus routes' transportation networks cannot be compared directly with each other. This is due to their transportation network orders not being the same, and therefore the potential unmet demand of the current Matie Bus network, due to its underlying graph having fewer vertices, would distort the values of the objective funtions in such a comparison. It can, however, be noted that it may hold considerable value for the students of Stellenbosch University if a larger portion of the student population could potentially be served by the new routes.

References

[1] Baaj MH & Mahmassani HS, 1991, *An AI based approach for transit route system planning and design*, Journal of Advanced Transportation, **25(2)**, pp. 187–209.

[2] Baaj MH & Mahmassani HS, 1995, *Hybrid route generation heuristic algorithm for the design of transit networks*, Transportation Research — Part C: Emerging Technologies, **3(1)**, pp. 31–50.

[3] Ceder A & Wilson NH, 1986, *Bus network design*, Transportation Research — Part B: Methodological , **20(4)**, pp. 331–344.

[4] Chakroborty P & Wivedi T, 2002, *Optimal route network design for transit systems using genetic algorithms*, Engineering Optimization, **34(1)**, pp. 83–100.

[5] Dunlap M, Li Z, Henrickson K & Wang Y, 2016, *Estimation of origin and destination information from Bluetooth and Wi-fi sensing for transit*, Transportation Research Record: Journal of the Transportation Research Board, **2595**, pp. 11–17.

[6] Fan W & Machemehl RB, 2004, *Optimal transit route network design problem: Algorithms, implementations, and numerical results*, (Unpublished) Technical Report, Center for Transportation Research, University of Texas, Austin (TX).

[7] Fan W & Machemehl RB, 2006, *Using a simulated annealing algorithm to solve the Transit Route Network Design Problem*, Journal of Transportation Engineering, **132(2)**, pp. 122–132.

[8] Guihaire V & Hao JK, 2008, *Transit network design and scheduling: A global review*, Transportation Research — Part A: Policy and Practice, **42(10)**, pp. 1251–1273.

[9] Huurne D & Andersen J, 2014, *A quantitative measure of congestion in Stellenbosch using probe data*, Proceedings of the 1st International Conference on the use of Mobile Informations and Communication Technology (ICT) in Africa, Stellenbosch, pp. 62–67.

[10] John MP, Mumford CL & Lewis R, 2014, *An improved multi-objective algorithm for the urban transit routing problem*, Evolutionary Computation in Combinatorial Optimisation, Lecture Notes in Computer Science, **8600**, pp. 49–60.

[11] Mandl CE, 1979, *Applied network optimization*, Academic Press, London.

[12] Mumford CL, 2013, *New heuristic and evolutionary operators for the multi-objective urban transit routing problem*, Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cardiff, pp. 939–946.

[13] Smith KI, Everson RM, Fieldsend JE, Murphy C & Misra R, 2008, *Dominance-based multiobjective simulated annealing*, IEEE Transactions on Evolutionary Computation, **12(3)**, pp. 323–342.

[14] Stellenbosch Smart Mobility Lab, 2018, *About SSML*, Stellenbosch Smart Mobility Lab, [Online], [Cited March 15th, 2018], Available from `http://www.sun.ac.za/english/faculty/eng/ssml/Pages/default.aspx`

[15] Stellenbosch University Sustainability, 2018, *Stellenbosch University Systemic Sustainability Campus shuttle service*, [Online], [Cited April 12th, 2018], Available from `http://www0.sun.ac.za/sustainability/pages/services/transport/campus-shuttle-service.php`

[16] Toth P & Vigo D, 2002, *The vehicle routing problem*, Society for Industrial and Applied Mathematics, Philadelphia (PA).

Proceedings of the 48th ORSSA Annual Conference
pp. 45–52
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3
©2019

Exact approaches towards solving generator maintenance scheduling problems

TK van Niekerk*   SE Terblanche†

Abstract

Generator Maintenance Scheduling (GMS) is concerned with finding a schedule for planned maintenance of power generation units while satisfying operating, maintenance, financial and national grid demand constraints. The practical use of two *mixed integer linear programming* (MILP) formulations are considered in this paper to solve realistic industry-sized GMS problems. The first MILP formulation is based on a popular time index formulation, which is frequently applied in the literature to solve GMS problems. The novelty of this study is the application of a second MILP formulation, which is based on a network flow representation of the well-known *resource constrained project scheduling problem.* Computational results presented in this paper show that the second MILP formulation has the potential to improve computing times when considering realistic GMS problem instances.

Key words: Generator maintenance, scheduling, mixed integer linear programming, network flows.

1 Introduction

Power generation is the conversion of chemical and mechanical energy into electrical energy. Electricity can be generated by utilising specialised process plants to convert the energy stored in coal, oil, gasoline, gas, uranium, wood, water, solar and wind sources into electricity. Coal-fired power stations contribute to more than 40% of global electricity generated [6]. Although there is a big drive for renewable energy production, developing countries like South-Africa still mainly rely on fossil fuel for power generation.

Power plant manufacturers and original equipment manufacturers recommend preventative maintenance to avoid breakdowns that may amount to significant financial expenditure compared to the actual cost of routine repairs. Therefore, planned maintenance is imperative for ensuring availability and reliability of a power generation plant while maximising production time [3].

A generator maintenance schedule is a timetable showing the allocated maintenance opportunities of each generating unit over a specific planning horizon [4]. In general, a typical

*School of Industrial Engineering, North-West University, Potchefstroom Campus, Republic of South Africa, email: thomasvn67@gmail.com

†School of Industrial Engineering, North-West University, Potchefstroom Campus, Republic of South Africa, email: fanie.terblanche@nwu.ac.za

generator maintenance schedule provides a one-year to five-year maintenance plan. These schedules may be amended based on available reserves, plant conditions, finances, maintenance crew, maintenance contracts and statutory requirements while aiming to maximise financial gain and improve availability and reliability. Moreover, an improved maintenance schedule can increase generator life which could, in turn, allow capital to be spent on building new power stations or development of renewable energy technology in order to meet the future national electricity demand requirements.

In this paper, the practical use of a time index and a resource flow problem formulation are considered to solve realistic industry-sized GMS problems. These two formulations are applied in the literature for solving general *resource constrained project scheduling problems*. The objective of this study is to empirically investigate the computational properties of these two problem formulations when applying a commercial off-the-shelf *mixed integer linear programming* (MILP) solver.

Contrary to the variety of objective functions proposed in the literature for solving GMS problems (*e.g.,* Lindner [5]), the objective function proposed in this paper prioritises the maintenance of more expensive power generation units over less expensive ones. The argument is that by removing the more expensive units from the grid for maintenance purposes, the corresponding savings in operational costs will maximise the financial gain of the power utility. More specifically, by introducing more cost savings early on during the maintenance planning horizon, the overall *net present value* (NPV) of the power utility will be maximised. This provides an opportunity for financial growth due to interest gained over time or possible investments in new capital ventures. In this study, the maximisation of the discounted *merit order operating costs* [7], which prioritises the maintenance of more expensive power generation units, is used as a proxy for the maximisation of NPV.

2 Mathematical Modelling Approach

The modelling approach followed in this paper differs from what is commonly found in the literature by formulating the GMS problem as a general *resource constrained project scheduling problem*. The maintenance activities required on a power generation plant typically involve multiple resources, *e.g.,* maintenance crews, spares, special tools *etc.,* and units may only be scheduled for maintenance if these resources are available. The GMS problem, therefore, naturally lends itself to be formulated as a resource constrained project scheduling problem.

Let $\mathcal{N} = \{1, 2, ..., |\mathcal{N}|\}$ denote the index set of all generating units. Each power station $p \in \mathcal{P}$ has a specific number of power generating units. The subset $\mathcal{N}(p) \subseteq \mathcal{N}$ denotes the set of generating units belonging to power station $p \in \mathcal{P}$. Each generating unit $i \in \mathcal{N}$ is allowed only one maintenance opportunity throughout the planning horizon $\mathcal{T}$, where $\mathcal{T} = \{1, 2, ..., |\mathcal{T}|\}$. The maintenance duration of a unit $i \in \mathcal{N}$ is d_i time periods. The value c_i represents the merit order operating cost of each unit $i \in \mathcal{N}$. For the purpose of formulating an objective function that maximises the discounted merit order operating costs, the discount rate I is introduced.

During maintenance execution each unit $i \in \mathcal{N}$ requires certain resources, *e.g.,* maintenance crews, spares, special tools *etc.* The index set of all resources is denoted by

$\mathcal{R} = \{1, 2, ..., |\mathcal{R}|\}$. Within the GMS environment, these types of resources are common among most generation units. Units can therefore only be scheduled for maintenance if the required resources are less than or equal to the available resources U_r during the maintenance phase. Let v_{ir} denote the amount of resource $r \in \mathcal{R}$ required for performing maintenance on a unit $i \in \mathcal{N}$.

Due to the use of auxiliary steam for the start-up of units within a power station and the existence of geographical transmission constraints, not all power generation units can simultaneously be scheduled for maintenance within the same power station or geographical region. For this purpose, a so-called precedence graph $H(\mathcal{N}, \mathcal{Z})$, with vertex set $\mathcal{N}$ and arc set $\mathcal{Z}$, is adopted to enforce a sequencing order when scheduling the maintenance of the units. The arc $(i, j) \in \mathcal{Z}$ stipulates that activity $i \in \mathcal{N}$ should precede activity $j \in \mathcal{N}$ for a given scheduling solution.

2.1 The Time Index GMS Model

Conceptually, the *time index GMS* (TI-GMS) model is based on the discretisation of time [2]. More specifically, the binary decision variable $x_{it} \in \{0, 1\}$ takes the value of 1 if the maintenance of unit $i \in \mathcal{N}$ is scheduled to start during time period $t \in \mathcal{T}$, and 0 otherwise. It is important to note that each unit is subjected to a fixed maintenance philosophy based on turbine running hours, statutory requirements and best practice intervals to ensure reliability of generating units. More specifically, the earliest and latest starting times S_i^b and S_i^e, dictate the specific time period in which each unit $i \in \mathcal{N}$ can be considered for maintenance. The subset $\mathcal{T}(i) \subseteq \mathcal{T}$ with $S_i^b \leq \mathcal{T}(i) \leq S_i^e$ denotes the set of allowable time periods of maintenance for unit $i \in \mathcal{N}$.

The binary decision variable $y_{it} \in \{0, 1\}$ is used to enforce a continuous maintenance duration d_i for each unit i over the planning horizon $\mathcal{T}$. The maintenance duration d_i is dependent on the type of scheduled maintenance, *e.g.,* an inspection outage or a general overhaul. The binary variable y_{it} takes the value of 1 if unit $i \in \mathcal{N}$ is in maintenance during time period $t \in \mathcal{T}$, and 0 otherwise.

The objective of the TI-GMS formulation is to

$$\text{maximise} \quad \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}} \frac{c_i}{(1 \ + \ I)^t} x_{it} \tag{1}$$

$$\text{s.t.} \quad \sum_{t\in\mathcal{T}(i)} x_{it} = 1, \qquad \forall\ i \in \mathcal{N}, \tag{2}$$

$$\sum_{t\in\mathcal{T}} y_{it} = d_i, \qquad \forall\ i \in \mathcal{N}, \tag{3}$$

$$\sum_{k=t}^{(t+di-1)} y_{ik} \geq d_i x_{it} \qquad \forall\ i \in \mathcal{N}, \forall\ t \in \mathcal{T}, \tag{4}$$

$$\sum_{i\in\mathcal{N}} v_{ir} y_{it} \leq U_r, \qquad \forall\ r \in \mathcal{R}, \forall\ t \in \mathcal{T}, \tag{5}$$

$$\sum_{t\in\mathcal{T}} t x_{jt} \geq \sum_{t\in\mathcal{T}} (t x_{it} + (d_i - 1) x_{it}), \qquad \forall\ (i, j) \in \mathcal{Z}. \tag{6}$$

The objective function (1) maximises the discounted operating costs, with the result that the more expensive units are scheduled for maintenance first while committing the cheaper units to satisfy the national electricity demand. Constraint set (2) ensures that each unit i is only allowed to be maintained once within the planning horizon $\mathcal{T}$. Constraint set (3) ensures that the maintenance duration is enforced while constraint set (4) prevents discontinuity within the maintenance schedule. Constraint set (5) is utilised to efficiently distribute resources between generating units while preventing resources from being over-committed for each time period t. Constraint set (6) ensure that the sequencing order of units in the solution is according to the precedence graph $H(\mathcal{N}, \mathcal{Z})$.

2.2 The resource flow GMS model

The idea to make use of a resource flow representation in solving resource constrained scheduling problems is due to [1]. The *resource flow GMS* (RF-GMS) formulation utilises a graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$ with the set of arcs $\mathcal{A}$ representing the flow of resources $r \in \mathcal{R}$ among the nodes $\mathcal{N}$. Each node $i \in \mathcal{N}$ in this graph corresponds to a power generation unit within the generation fleet, where $\mathcal{N} = \{0, 1, ..., N\}$. The index 0 is used to define an auxiliary source unit and the index N an auxiliary sink unit. The balance of resource flows dictate that the source and sink node resource requirements v_{0r} and v_{Nr}, respectively, should be set equal to the availability of the resources U_r for all $r \in \mathcal{R}$. The notation $(i, j) \in \mathcal{A}(i)$ represents a set of arcs where node i denotes the source and $(i, j) \in \mathcal{A}(j)$ represents a set of arcs where node j denotes the target.

The decision variable $S_i^b \leq s_i \leq S_i^e$ determines the maintenance start time of a generation unit $i \in \mathcal{N}$ and the decision variable f_{ijr} represents the flow of resources $r \in \mathcal{R}$ for each arc $(i, j) \in \mathcal{A}$. Allocation of resources v_{ir} among units is dictated by the binary scheduling decision variable z_{ij}. When flow of resources from node i to j is permitted, z_{ij} will take the value of 1 and 0 otherwise.

The resource flow formulation does not naturally lend itself to a MILP formulation that maximises the discounted merit order operating costs due to the non-linear objective function

$$\max \sum_{i \in \mathcal{N}} e^{-I \times s_i} c_i. \tag{7}$$

Therefore, the approximation approach in [8] is adopted in this paper for the purpose of formulating the RF-GMS problem. Let (k_{iv}, h_{iv}) be the data points of the piecewise linear approximation of the non-linear objective function (7), where $i \in \mathcal{N}$ and $v \in \mathcal{V} = \{0, 1, ..., V-1\}$. Two auxiliary decision variables $\lambda_{iv} \geq 0$, with $v \in \mathcal{V}$ and $l_{iv} \in \{0, 1\}$, with $v \in \mathcal{V} \setminus \{0\}$, are introduced into the resource flow formulation. The former allows the decision variable s_i to be defined as the convex combination of the data points k_{iv} and h_{iv}, respectively, while the latter is a decision variable which indicates the appropriate line segment selection with reference to the linear approximation of the objective function. The decision variable n_i represents the discounted merit order operating costs for each unit $i \in \mathcal{N}$ as determined by the piecewise linear approximation. The decision variable l_{iv} takes the value of 1 when a line segment has been selected and 0 otherwise.

The objective of the RF-GMS formulation is to

$$
\begin{aligned}
\text{maximise} \quad & \sum_{i \in \mathcal{N}} n_i & & (8)\\
\text{s.t.} \quad & s_j - s_i - (d_i + M) z_{(i,j)} \geq -M, & \forall\ (i,j) \in \mathcal{Z}, & \quad (9)\\
& \sum_{(i,j) \in \mathcal{A}(i)} f_{ijr} = v_{ir}, & \forall\ i \in \mathcal{N} \setminus \{N\}, \forall\ r \in \mathcal{R}, & \quad (10)\\
& \sum_{(i,j) \in \mathcal{A}(j)} f_{ijr} = v_{jr}, & \forall\ j \in \mathcal{N} \setminus \{0\}, \forall\ r \in \mathcal{R}, & \quad (11)\\
& f_{ijr} - \min\{v_{ir}, v_{jr}\} z_{(i,j)} \leq 0, & \forall\ (i,j) \in \mathcal{A}, \forall\ r \in \mathcal{R}, & \quad (12)\\
& s_i = \sum_{v \in \mathcal{V}} k_{iv} \lambda_{iv}, & \forall\ i \in \mathcal{N}, & \quad (13)\\
& n_i = \sum_{v \in \mathcal{V}} h_{iv} \lambda_{iv}, & \forall\ i \in \mathcal{N}, & \quad (14)\\
& \sum_{v \in \mathcal{V}} \lambda_{iv}, = 1, & \forall\ i \in \mathcal{N}, & \quad (15)\\
& \lambda_{i0} - l_{i1} \leq 0, & \forall\ i \in \mathcal{N}, & \quad (16)\\
& \lambda_{iv} - l_{iv} + l_{i(v+1)} \leq 0, & \forall\ i \in \mathcal{N}, \forall\ v \in \mathcal{V} \backslash \{0, V-1\}, & \quad (17)\\
& \lambda_{i(V-1)} - l_{i(V-1)} \leq 0, & \forall\ i \in \mathcal{N}, & \quad (18)\\
& \sum_{v \in \mathcal{V}} l_{iv}, = 1, & \forall\ i \in \mathcal{N}. & \quad (19)
\end{aligned}
$$

The objective function (8) allows the discounted operating cost to be maximised while constraint set (9) along with the linear ordering variable z_{ij} ensures that resources can only be allocated to service unit i after the preceding unit j's maintenance has been completed according to $s_i + d_i$. The parameter M in (9) is selected as the latest possible completion time of the scheduling calendar. Constraint set (10) represents the flow of resources from a node i, while constraints (11) represent the flow of resources into a node j. Constraint set (12) allows the flow of resources between units based on the solution of the linear ordering variable z_{ij}. Constraint sets (13) and (14) express s_i and n_i as convex combinations of the linear piecewise segments of the objective function. Constraints (15) enforce convexity while constraints (16)–(18) allow activation of decision variable λ_{iv} to take appropriate values based on the linear line segment selection as dictated by constraint set (19).

3 Computational Results

The data used in this paper for comparing the computational efficiency of the two proposed GMS formulations is based on historical and predicted data from the South African National power utility Eskom. The input data generated correspond well with the outage planning tool Tetris used by the National Power utility to plan maintenance opportunities based on the grid constraints. Information on a total of ninety-two generation units is included in this data set over a planning horizon of 365 days (see [9] for more detail). For the computational results below, only two resources were considered. The first resource relates to the crew requirement during maintenance of each power generation unit and the second resource relates to the maximum allowable power generation capacity that may be removed from the grid while still satisfying national demand. Implementing the latter is achieved by assigning the maximum design output of a unit $i \in \mathcal{N}$ to the resource consumption parameter v_{ir} of the said unit. The capacity U_r associated with this specific resource is defined as the excess reserve of the entire grid, which is calculated as the total generation capacity of the entire fleet of units minus the national demand (plus some safety factor).

For the purpose of comparing the scalability of the two proposed GMS formulations, problem instances comprising 5, 10, 20, 40 and 92 units, respectively, were created. Both the TI-GMS and RF-GMS model formulations were solved by means of the commercial solver CPLEX on a Dell i7-7500U laptop with 6GB of RAM.

Number of units	Time index (TI-GMS)	Resource flow (RF-GMS)
5 Units	00:02:62	00:01:10
10 Units	00:03:84	00:02:20
20 Units	00:07:29	00:03:92
40 Units	00:22:86	00:02:88
92 Units	00:44:50	00:04:37

Table 1: *Computing times (minutes) for the TI-GMS and RF-GMS problem formulations.*

The computing times recorded when solving the problem instances with 5, 10, 20, 40 and 92 units, respectively, are provided in Table 1. Each problem instance was solved to optimality for both the TI-GMS and RF-GMS problem formulations. From the results it is clear that the resource flow model formulation is significantly faster in providing an optimal generator maintenance schedule, compared to the time index formulation. This is especially true for larger problem instances. The computational time of the time index formulation is almost directly proportional to the number of units considered. This may be explained by the fact that the planning horizon is discretised for each unit in the time index formulation, leading to a significant increase in the number of binary decision variables.

Figure 1 is a graphical representation of the results in Table 1. This clearly shows the superior scalability of the resource flow formulation compared to the time index formulation.

4 Conclusion

In this paper, the computational efficiency of two mixed integer linear programming formulations was investigated when solving the generator maintenance scheduling problem. The TI-GMS is a time index formulation, which is based on the discretisation of time, and the RF-GMS is a resource flow-based formulation. The linear approximation adopted in the resource flow-based formulation allows for the computation of optimal solutions within an average error margin of 0.1%.

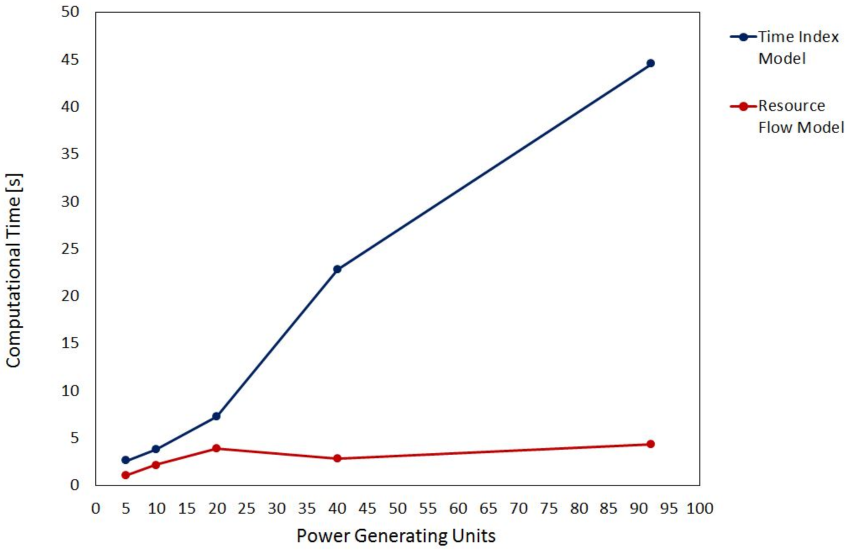


Figure 1: *Time index formulation vs. resource flow formulation of the GMS problem.*

The objective function proposed in this paper maximises the discounted operating cost of each power generation unit. By doing this, priority is given to the maintenance of more expensive units, which may be considered as being equivalent to minimising operational risk while maximising financial gain. The motivation is that by removing the more expensive units from the grid, the corresponding savings in operational costs will maximise the financial gain of the power utility. More specifically, the maximisation of the discounted operating cost is used as a proxy for the maximisation of NPV.

Computational tests were performed on data from Eskom — the South African National power utility. Initial tests demonstrated that the RF-GMS formulation achieves superior scalability, which allows the solution of a problem instance with ninety-two power generation units to be computed within five minutes.

References

[1] Artigues C, Michelon P & Reusser S, 2003, *Insertion techniques for static and dynamic resource-constrained project scheduling*, European Journal of Operational Research, **149(2)**, pp. 249–267.

[2] Christofides N, Alvarez-Valdés R & Tamarit JM, 1987, *Project scheduling with resource constraints: A branch-and-bound approach*, European Journal of Operational Research, **29(3)**, pp. 262–273.

[3] Fetanat A & Shafipour G, 2011, *Harmony search algorithm based 0-1 integer programming for generation maintenance scheduling in power systems*, Journal of Theoretical and Applied Information Technology, **24(1)**, pp. 1–10.

[4] Firuzabad MF, Aminifar F & Shahzadeh A, 2014, *Reliability-based maintenance scheduling of generating units in restructured power systems*, Turkish Journal of Electrical Engineering Computer Sciences, **22(5)**, pp. 1147–1158.

[5] Lindner BG, 2017, *Bi-objective generator maintenance scheduling for a national power utility*, PhD Dissertation, Stellenbosch University, Stellenbosch.

[6] Lindner BG & van Vuuren JH, 2014, *Maintenance scheduling for the generating units of a national power utility*, Proceedings of the 43$^{\text{rd}}$ Annual ORSSA Conference, Parys, pp. 36–44.

[7] Saraiva JT, Pereira ML, Mendes VT & Sousa JC, 2011, *A simulated annealing based approach to solve the generator maintenance scheduling problem*, Electrical Power Systems Research, **81(7)**, pp. 1283–1291.

[8] Terblanche SE, 2017, *Resource constrained project scheduling models and algorithms applied to underground mining*, PhD Dissertation, Stellenbosch University, Stellenbosch.

[9] van Niekerk TK, 2018, *Exact approaches towards solving generator maintenance scheduling problems*, MSc Thesis, North West University, Potchefstroom.

Proceedings of the 48th ORSSA Annual Conference
pp. 53–62
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3
©2019

Modelling the movement of drivers of Transportation Network Companies using an agent-based simulation model

M Rolo[*] GS Nel[†]

Abstract

Technological innovation has transformed transportation into a more personal and customer-driven experience. Traditional public transport, such as trains and buses, is limited to pre-defined routes that are fixed. Consequently, riders have to travel along one of these routes regardless of their destination. Technology-driven companies, such as Uber, Bolt and Lyft, are so-called *Transportation Network Companies* (TNCs) that provide riders with the option of selecting any pick-up and drop-off location. The nearest available driver (to the rider's pick-up location) is then assigned to the rider; thereafter the driver proceeds to pick up the rider who is then dropped off at their destination. A basic agent-based simulation model based on past pick-up data of a well-known TNC has been developed in an effort to analyse the operation of such a system. It is envisaged that the model may aid the decision making process related to various important facets of such a system, *e.g.,* choosing suitable waiting locations for drivers with the aim to minimise rider waiting times. The developed simulation model is subjected to a verification process in order to ensure it is correctly implemented with respect to the conceptual model.

Key words: Agent-based model, transportation, simulation.

1 Introduction

The *digital age* is characterised by the ability to easily collect, transfer and process information at an unprecedented scale [4]. Technological innovation has transformed the way people go about their daily lives. Activities that were typically regarded as time consuming, such as banking and commerce, have been transformed into trivial processes performed by almost any individual by means of an easily accessible platform, *i.e.,* a website or mobile application. Transportation is no exception — the ability to request a

[*] Stellenbosch Unit for Operations Research and Engineering, Department of Industrial Engineering, University of Stellenbosch, Private bag X1, Matieland, 7602, Republic of South Africa, email: 18313108@sun.ac.za

[†] Stellenbosch Unit for Operations Research and Engineering, Department of Industrial Engineering, University of Stellenbosch, Private bag X1, Matieland, 7602, Republic of South Africa, email: gsnel@sun.ac.za

personal ride service that is tailored to the needs of the riders is now at the fingertips of many. The continuous development of these services has resulted in greater control over the routes that can be travelled.

Transportation is evidently evolving into a more personal experience. So-called *Transportation Network Companies* (TNCs) are at the forefront of this phenomenon. These technology-driven companies provide web- or application-based services that enable riders (*i.e.,* individuals in need of transportation) to match and secure a ride-request with a driver [3]. Riders typically use a mobile application to input their pick-up location — usually their current location — together with their drop-off location (destination) which is then processed by the TNC. The *estimated time of arrival* (ETA), together with the estimated cost of the ride, is presented to the rider (*via* the application), which must first be accepted before the service is initiated. Upon acceptance, the nearest driver is typically assigned to the rider based on the travel distance between the driver and the rider. After the rider is picked up by the driver, the driver then proceeds to drop off the rider at the requested destination. The driver then waits to be requested again. A driver may either be requested at a waiting area, or whilst travelling towards a waiting area.

A model that accurately simulates the operation of such a system may prove useful as it facilitates further analyses, especially with respect to the implementation of potentially beneficial strategies that can be employed by TNCs. One such an example is the recommendation of improved driver waiting locations with the aim to minimise rider waiting times. To this end, a review of what is known about the operation of TNC services, as documented in the literature, is first presented in §2. This is followed in §3 by a discussion on the manner according to which the operation of such a system may be implemented in the agent-based simulation model, along with the relevant assumptions made. An introduction into the basic concepts of agent-based modelling is given in §3.1, which is followed in §3.2 by a description of the data set that forms the basis of the work presented in this paper. This is followed in §3.3 by a description of how the movement of drivers is modelled in the AnyLogic environment. In §3.4, a verification of the model against the documented process and the model assumptions is performed so as to ascertain the accuracy with which the abstraction and reconstruction is achieved. The paper then closes in §4 with some recommendations with respect to possible future work.

2 Review of the operation of TNC ride services

Simulation models of the operation of TNCs have been developed by Shaheen [15]. A simulaiton model was also developed by Lokhandwala & Cai [9] to model ride sharing (*i.e.,* where a ride is shared by riders whose routes overlap either fully or partially) using ride sharing services, such as UberPool and Lyft Line, and autonomous taxis. A popular topic of discussion in the literature, similar to TNCs, is the taxi industry. The traditional yellow taxis are the only vehicles in New York City that are allowed to pick riders up anywhere in New York City by means of street-hailing (*i.e.,* picking-up a rider from the street who has not reserved a ride) and by means of pre-arranged ride-requests. Drivers of TNCs, however, may only pick riders up by means of pre-arranged ride-requests. Simulation models have been developed by Cheng & Nguyen [2], Grau & Romeu [6] and Kim *et al.*

[7], among others, to study certain behaviours of this system. Studies have also been conducted by Lee *et al.* [10] and Zhao *et al.* [17] in which the aim is to find suitable locations (or hotspots) where taxi drivers should wait to be requested.

3 Simulating the movement of drivers

An agent-based model of drivers and riders of TNCs has been designed and developed in the AnyLogic Personal Learning Edition 8.3.2 software suite. This model simulates the movement of drivers and the interactions (*i.e.,* pick-up and drop-off) between drivers and riders. A general operation of this process is illustrated graphically in Figure 1. Once a rider requests a ride, the nearest available driver is identified and assigned to the ride-request. Some TNCs provide the option of selecting a larger vehicle or a luxury vehicle, however, an assumption of homogeneity is made — *i.e.,* all vehicles are assumed to be the same. An available driver may either be at a waiting area (waiting for a ride-request) or driving towards a waiting area after completing a ride. Waiting areas can include well-known (established) hotspots, *i.e.,* areas known for having a high pick-up concentration (based on historical data). In this model, the waiting areas are restricted to known parking areas. Once all the available drivers are identified, the driver closest to the rider, *i.e.,* the driver having the shortest driving distance to the rider, is selected and the ride-request is confirmed. If there are no available drivers, the model re-checks for available drivers until one is found. The waiting time experienced by a rider is therefore affected by the availability of drivers.

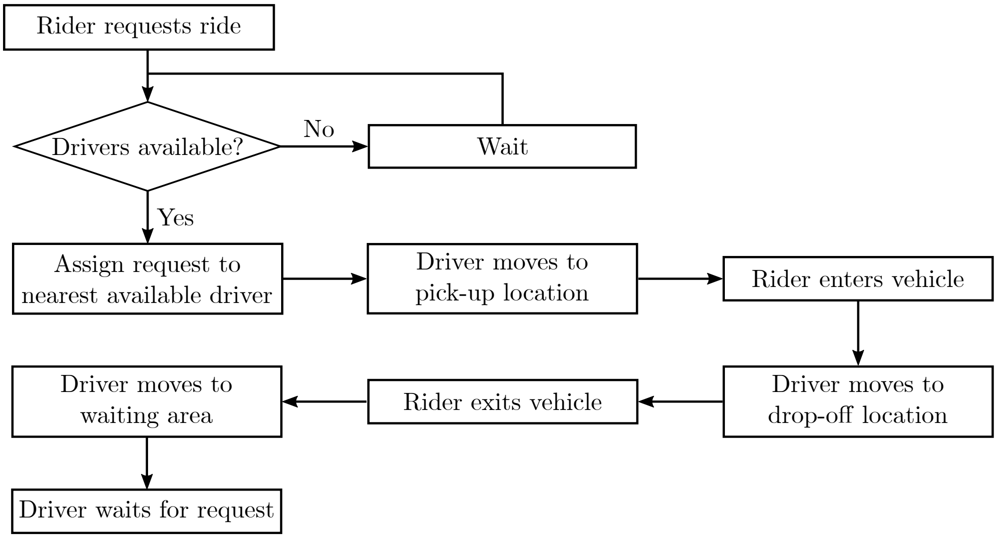

Rider requests ride
Drivers available?
No
Wait
Yes
Assign request to nearest available driver
Driver moves to pick-up location
Rider enters vehicle
Driver moves to drop-off location
Rider exits vehicle
Driver moves to waiting area
Driver waits for request

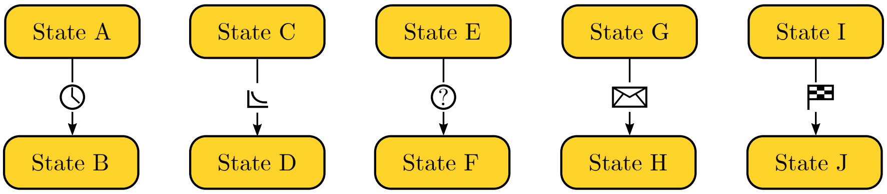
Figure 1: *A flowchart of the general operation of a TNC service.*

Once a driver is assigned to a ride-request and the driver is no longer available for other ride-requests, movement towards the rider is initiated. When the driver arrives at the pick-up location, the rider enters the vehicle after which the driver travels to the drop-off location. Upon arrival, the rider exits the vehicle and the driver moves to a parking area (*i.e.,* an open area where individuals may park their vehicles) to wait for the next ride-request. This parking area is selected at random; however, this may be changed to select the parking area closest to the driver or only to select vacant parking areas, *i.e.,* parking areas currently unoccupied by another driver. It is assumed that the driver always travels along the fastest route. AnyLogic employs the speed limits of the various road segments to determine the fastest route. It is possible to incorporate traffic information to further enhance model realism; however, this was excluded for the sake of simplicity.

3.1 Introduction of agent-based modelling

In an agent-based modelling paradigm the interactions of several objects or agents are modelled. These interactions describe (give rise to) the characteristics of the system as a whole. An object may be considered an agent if the following conditions hold true: It acts in an autonomous manner; it is a discrete entity; it interacts with other agents or its condition or state changes over time [11]. AnyLogic employs the icon to identify an agent type, *i.e.,* an object class.

An agent performs various actions which are dependent on its state. This may be described using statecharts which comprises states and transitions. A transition moves an agent between different states once it is initiated (or activated) and may be initiated by using a *timeout*, *rate*, *condition*, *message* or *arrival* trigger. A timeout trigger, illustrated graphically in Figure 2(a), indicates that once an agent has entered State A, it remains in State A for a specified time period before transitioning to State B. A rate trigger, illustrated in Figure 2(b), is similar to a timeout trigger; however, instead of specifying a constant time interval, some probability distribution (*e.g.,* exponential) can be applied. In Figure 2(c), a condition trigger is illustrated graphically. Accordingly, after an agent enters State E, it only moves to State F when a specified condition is true. A message trigger, illustrated in Figure 2(d), allows an agent to move from State G to State H when the agent receives a message. This transition may be set to trigger on any message or only on a specific message. Lastly, according to an arrival trigger, as illustrated in Figure 2(e), an agent transitions from State I to State J once it has arrived at its destination [16].



(a) Timeout (b) Rate (c) Condition (d) Message (e) Arrival

Figure 2: *Different types of transition triggers.*

An agent may have parameters associated with it which are indicated in AnyLogic by the icon. Parameters are used to represent characteristics of the agent which usually do not change during a simulation run. An agent may also have variables which are indicated in AnyLogic by the V icon. Variables may be used to store results of the simulation or to model agent characteristics that change over time.

3.2 Description of the data set

The data set forming the basis of the work presented in this paper contains the location and time of pick-ups of a well-known TNC that occurred on the 1^{st} of April 2014 in New York City [5]. The drop-off locations, however, were omitted from the original data set. To account for this, the drop-off locations were randomly sampled from another data set which captured the pick-up and drop-off locations of taxi services in New York City [12]. This ensures that the drop-off locations accurately represent potential real-world destinations. A sample of the data set is shown in Table 1.

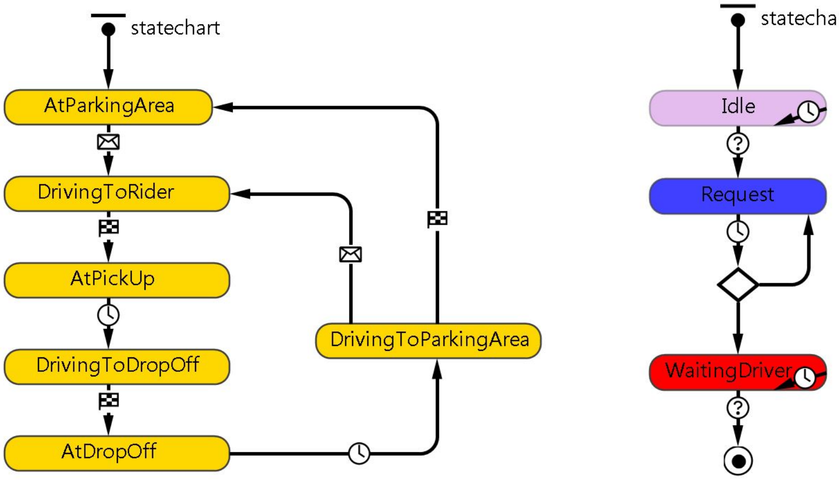
Pick-up date and time	Pick-up latitude	Pick-up longitude	Drop-off latitude	Drop-off longitude
2014/04/01 09:00:00 AM	40.7841	-73.9542	40.67358	-73.80718
2014/04/01 09:02:00 AM	40.7426	-73.9963	40.73845	-73.73162
⋮	⋮	⋮	⋮	⋮
2014/04/01 11:49:00 PM	40.7661	-73.9693	40.71436	-73.90057
2014/04/01 11:54:00 PM	40.7610	-73.9768	40.59303	-73.77763

Table 1: *A sample of the data set employed. Indicated is the pick-up and drop-off time and locations of riders on the* 1^{st} *of April 2014.*

3.3 Description of implemented process in AnyLogic environment

In the simulation model, **drivers** and **riders** agent types exist to facilitate the behaviours of the driver and rider, respectively. Another agent type called **parkingAreas** exists and is used to specify the different parking areas. This additional agent type is supplementary as it allows for complexity to be added, *e.g.,* specifying the occupancy of the parking area. As can be seen in Figure 3(a), an agent of the drivers agent type enters the simulation and assumes the **AtParkingArea** state immediately. This state indicates that the driver is currently at a parking area waiting for a ride-request. An agent of the riders type enters the simulation and immediately assumes the **Idle** state, as can be seen in Figure 3(b). An agent in this state is not active in the simulation and is therefore invisible in the animation as a ride-request has not yet been made by this agent. The parkingAreas statechart comprises only one state, the **Active** state. The driver remains in the **AtParkingArea** state until a message is received from a rider that has transitioned into the **Request** state upon which it becomes visible in the simulation. A rider transitions into this state when the time in the simulation is the same as its pick-up time. This message is only sent to the nearest available driver. A driver's availability depends on whether it is in the **AtParkingArea** state or in the **DrivingToParkingArea** state. A driver will transition into the **DrivingToParkingArea** state once the rider has been dropped-off. If an available driver cannot be found, the rider re-enters the **Request** state and continues searching for an available driver every five seconds until one is found.

When an available driver in the **AtParkingArea** state receives a message from a rider, the driver transitions into the **DrivingToRider** state, as indicated by the message transition trigger. The rider responsible for sending the message transitions to the **WaitingDriver**



(a) Statechart of driver agent.

(b) Statechart of rider agent.

Figure 3: *Statecharts of the driver agent and the rider agent.*

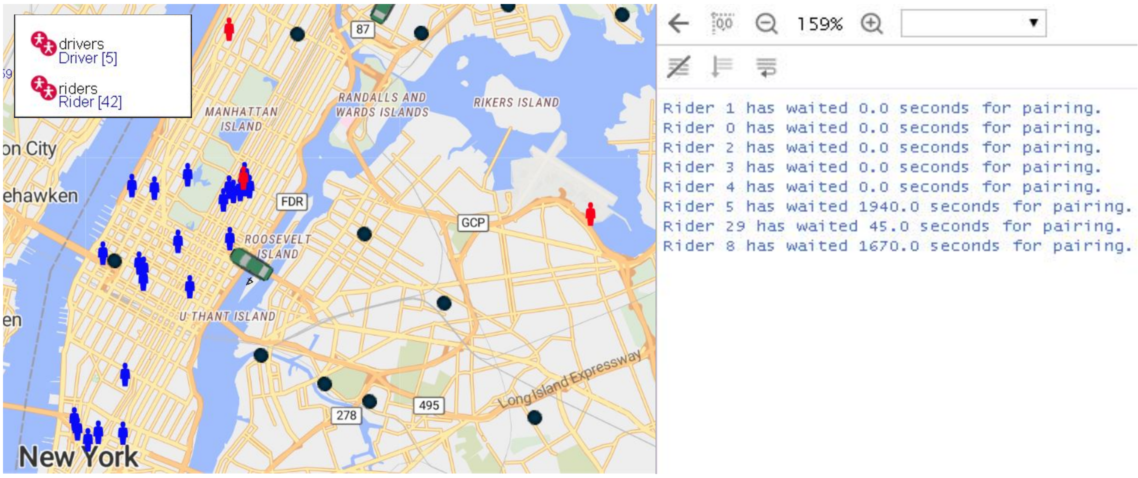
state. The driver then moves towards the location of the rider that sent the message. Upon arrival (at the rider's location) the driver transitions into the **AtPickUp** state, indicated by the arrival transition trigger. The rider then transitions into the **FinalState**, indicated in the statechart with the ◉ icon. The rider enters this state on the condition that the driver and the rider are at the same location. When a rider enters the **FinalState**, it is no longer visible in the simulation as the rider is regarded as inside the driver's vehicle. The transition between the **AtPickUp** state and the **DrivingToDropOff** state is a timeout trigger. This is used to simulate the time it takes for a rider to enter the driver's vehicle.

Upon entering the **DrivingToDropOff** state, the driver processes the drop-off location from the rider and travels towards it. Two parameters are employed in order to store the latitude and longitude coordinates of the drop-off location, namely **DropOffLat** and **DropOffLong**, respectively. Once again, an arrival transition trigger is employed in order to transition the driver into the **AtDropOff** state upon arrival at the drop-off location. A timeout trigger transitions the driver into the **DrivingToParkingArea** state to simulate the time it takes for a rider to exit the driver's vehicle. When the driver is in the **DrivingToParkingArea** state, a random agent of the parkingAreas agent type is selected and the driver moves towards it. The driver may leave this state by means of two transitions. The first transition is triggered upon receipt of a message by a rider indicating a ride-request. A driver receiving a ride-request whilst driving to a parking area is therefore simulated. When a driver receives this message it immediately transitions into the **DrivingToRider** state and the driver continues to move through the states as discussed previously. If the driver arrives at the parking area without being requested by a rider, the arrival trigger activates and the driver transitions into the **AtParkingArea** state and the driver continues to move through the states as discussed previously.

3.4 Verification

Verification is a method of testing whether a model has been implemented correctly and performs as intended [8, 14]. A model that is verified as functionally correct and logically sound can be used to answer pertinent research questions regarding the relevant domain. One method of verification is to test the model over a range of parameter values in order to determine whether the results of these changes are reflected by the model [1]. The model developed in this study is verified by changing the number of drivers whilst keeping the number of riders and parking areas constant. The positioning and time corresponding to the moment when a rider requests a ride are kept constant. The positioning of the parking areas is also fixed. The resulting effect on the *assignment waiting time* (AWT), *i.e.,* the time elapsed between when a rider requests a ride and when a driver is assigned, is then recorded and stored in a variable called Ⓥ **AssignWaitTime**. It is expected that an increase in the number of drivers will result in a decrease in the AWT — ascribed to the increased availability of drivers to be assigned.

In the first experiment that forms part of model verification, a one hour period of historical ride-requests were considered. The one-hour period comprised forty-two ride-requests. In addition, a total of seventy parkingAreas were randomly drawn from a dataset containing the parking lots in New York City [13]. A screenshot of the simulation model in AnyLogic (during runtime) is shown in Figure 4. In the top left hand corner a white box indicating the number of drivers and the number of riders in the simulation is included. Drivers and riders are visually denoted by simple illustrations of cars and silhouettes, respectively. The silhouette is blue when the rider is in the **Request** state and red when the rider is in the **WaitingDriver** state. The parkingAreas are represented visually by black dots. The console (on the right) displays information related to the length of time each rider had been waiting before being assigned to an available driver. Due to the fact that there are only five drivers in this experiment and that the simulation started with all drivers being available, the first five ride-requests were assigned to drivers immediately and therefore have an AWT of zero seconds (as shown in the console). The number of driver agents



```
Rider 1 has waited 0.0 seconds for pairing.
Rider 0 has waited 0.0 seconds for pairing.
Rider 2 has waited 0.0 seconds for pairing.
Rider 3 has waited 0.0 seconds for pairing.
Rider 4 has waited 0.0 seconds for pairing.
Rider 5 has waited 1940.0 seconds for pairing.
Rider 29 has waited 45.0 seconds for pairing.
Rider 8 has waited 1670.0 seconds for pairing.
```

Figure 4: *Screenshot of the simulation model during runtime.*

employed in the simulation model and the corresponding resulting average AWT is shown in Table 2. By increasing the number of drivers, the average AWT is decreased. As stated previously, this is an expected outcome which is both logical and intuitive. The average AWT equates to zero once a critical number of drivers are present, ascribed to the fact that there are enough drivers available to be immediately assigned to riders upon ride-requests.

Number of driver agents	Average AWT (seconds)
5	7420.00
10	2691.55
15	1201.55
20	448.81
25	126.31
30	0.00
35	0.00

Table 2: *The number of driver agents used in the simulation model together with the corresponding average AWT of rider agents.*

The second method of verification is to test the impact of different waiting strategies on the *total waiting time* (TWT), *i.e.,* the time elapsed between when a rider requests a ride and when he/she is picked up by a driver. The first waiting strategy is to assign drivers to parking areas with the highest number of pick-up requests received in the past hour within a 10km travel distance. According to the second waiting strategy, drivers are assigned to the parking areas with zero pick-up requests received in the past hour within a 10km travel distance. For both strategies, the TWT is recorded and stored in a variable called Ⓥ **TotalWaitTime**. The drivers in Strategy 1 should be able to access the riders sooner than the drivers in Strategy 2 and the riders should therefore have a lower average TWT. For both scenarios, there were a total of twenty drivers available. The results of this simulation are shown in Table 3. As expected, the riders in Strategy 1 have a markedly lower average TWT than those in Strategy 2.

Strategy number	Average TWT (seconds)
1	1237.04
2	1906.21

Table 3: *The number of driver agents used in the simulation model together with the corresponding average TWT experienced by the rider agents.*

4 Conclusion and future work

The manner in which the operation of drivers and riders of TNCs has been modelled within an agent-based simulation paradigm was described in this paper. The modelling approach simulated the interactions of three agent types, namely, drivers, riders and parking areas. This model was then verified by observing and quantifying the impact of a change in the number of drivers on the average AWT. In addition, different driver waiting strategies were evaluated. It was found that when drivers wait in areas with a high number of surrounding ride-requests, the TWT is less than in the strategy with drivers who wait in areas having no surrounding ride-requests. Based on the results of the two verification tests performed, it is evident that the simulation model developed represents, to an extent, the operation of such a system.

This simulation model forms part of an on-going project at Stellenbosch University aimed at identifying suitable waiting areas for drivers in order to minimise total rider waiting time. These waiting areas will be determined using an unsupervised machine learning technique called *clustering*. The clusters are formed based on historical rider pick-up locations (also in New York City). The parking areas used in the model (as mentioned in this paper) will be replaced by the waiting areas corresponding to the clustering technique. The average total rider waiting times of the two approaches will be compared. The first approach will represent the waiting areas found using clustering, whereas the second approach will represent the waiting areas selected based on nearby hotspots and parking areas. Along with this, a decision support system will be developed in order to assist the drivers in selecting a waiting area.

References

[1] Carson JS, 2002, *Model verification and validation*, Proceedings of the 2002 Winter Simulation Conference, Piscaraway (NJ) pp. 52–58.

[2] Cheng SF & Nguyen TD, 2011, *TaxiSim: A multiagent simulation platform for evaluating taxi fleet operations*, Proceedings of the 2011 IEEE/WIC/ACM International conferences on Web Intelligence and Intellegent Agent Technology, Singapore, pp. 14–21.

[3] Clewlow RR & Mishra GS, 2017, *Disruptive transportation: The adoption, utilization, and impacts of ride-hailing in the United States*, Institution of Transportation Studies at University of California, Davis (CA).

[4] Dictionary.com, 2018, *Information Age*, [Online], [Cited April 4^{th}, 2019], Available from `http://www.dictionary.com/browse/information-age`

[5] Kaggle, 2016, *Uber Pickups in New York City*, [Online], [Cited July 9^{th}, 2019], Available from `https://www.kaggle.com/fivethirtyeight/uber-pickups-in-new-york-city`

[6] Grau JMS & Romeu MAE, 2015, *Agent based modelling for simulating taxi services*, Procedia Computer Science, **52**, pp. 902-907.

[7] Kim H, Yang I, & Choi Keechoo, 2011, *An agent-based simulaiton model for analysing the impact of asymmetric passenger demand on taxi service*, KSCE Journal of Civil Engineering, **15(1)**, pp. 187–195.

[8] Kleijnen JPC, 1995, *Verification and validation of simulation models*, European Journal of Operational Research, **82(1)**, pp. 145–162.

[9] Lokhandwala M, & Cai H, 2018, *Dynamic ride sharing using traditional taxis and shared autonomous taxis: A case study of NYC.*, Transportation Research Part C: Emerging Technologies, **97**, pp. 45–60.

[10] Lee J, Shin I & Park GL, 2018, *Analysis of the passenger pick-up pattern for taxi location recommendation.*, Proceedings of the 4th International Conference on Networked Computing and Advanced Information Management, Jeju City, pp. 1999–204.

[11] Macal CM & North MJ, 2010, *Tutorial on agent-based modeling and simulation*, Journal of simulation, **4(3)**, pp. 151–162.

[12] New York City Taxi and Limousine Commision, 2019, *TLC Trip Record Data*, [Online], [Cited July 9th, 2019], Available from `https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page`

[13] NYC Open Data, 2018, *Parking Lot*, [Online], [Cited July 16th, 2019], Available from `https://data.cityofnewyork.us/City-Government/Parking-Lot/h7zy-iq3d`

[14] Sargent RG, 2008, *Validation and verification of simulation models*, Proceedings of the 2008 Winter Simulation Conference, Syracuse (NY), pp. 39–48.

[15] Shaheen JA, 2019, *Simulating the ridesharing economy: The Individual Agent Metro-Washington Area Ridesharing Model (IAMWARM)* , pp. 143–168 in Carmichael T, Collins A & Hadžikadić M (Eds.), *Complex adaptive systems*, Springer, Cham.

[16] XJ Technologies Company, 2018, *AnyLogic 8 user's guide*, [Online], [Cited August 26th, 2019], Available from `http://www.xjtek.com/`

[17] Zhao PX, Qin K, Zhou Q, Liu K & Chen YX, 2015, *Detecting hotspots from taxi trajectory data using spacial cluster analysis*, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, **2(4)**, pp. 131–135.

Proceedings of the 48th ORSSA Annual Conference
pp. 63–71
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3
©2019

An optimisation model for supporting mobile rural healthcare services

I Guess[*] SE Terblanche[†]

Abstract

The use of day clinics is for a large proportion of the South African population the only way currently to gain access to public healthcare. Some of the day clinics in the country have to provide healthcare services to as many as 32 000 inhabitants per district, with an average of only four nurses on duty per day. Apart from the lack of resources within the day clinics, for some patients, a trip to the nearest day clinic may take up to several hours. In this paper, the viability of adopting mobile healthcare is explored as a means of bringing healthcare services closer to the patient. Such a service will be complimentary to the day clinics with the primary objective to provide better quality healthcare to the rural population. A mathematical programming formulation is proposed to deal with the operational management of a mobile healthcare service, and a fictitious problem instance is used to demonstrate the functionality of the formulation.

Key words: Rural healthcare, mobile clinics, scheduling, mathematical programming.

1 Introduction

Within rural communities, healthcare is not always easily accessible. According to Neely & Ponshunmugam [6], the main reasons are resource scarcity, transportation networks not designed around healthcare services, and families that are split between rural and urban areas. Patients sometimes live too far from day clinics or government hospitals, and they struggle to find or afford transportation [4]. In the past decade, various countries have investigated and tested the feasibility of mobile clinics. In India, for instance, the Ila Trust was founded in 1994 to provide mobile healthcare to rural communities [1]. Mobile clinics go to predefined locations and operate according to a fixed weekly schedule. The treatment of general ailments and chronic diseases, such as AIDS and diabetes, is provided by a staff complement of doctors, nurses, pharmacists and drivers. Due to the fixed mode of operation, resource allocation and logistical management are simplified.

[*] School of Industrial Engineering, North-West University, Republic of South Africa, email: guessinge1@gmail.com

[†] School of Industrial Engineering, North-West University, Republic of South Africa, email: fanie.terblanche@nwu.ac.za

In this paper, alternative operational models to manage mobile clinics are considered. More specifically, instead of only routing mobile clinics to the same predefined locations according to a fixed weekly schedule, the possibility of scheduling mobile services according to patient demands is explored. With the extensive coverage of mobile networks in South Africa, as well as the increasing affordability of smartphone technologies, the use of a mobile application for gathering patient demands could be a viable option in providing better patient-oriented healthcare services. If patients could indicate their preferences towards certain treatment locations as well as appointment times, it may be possible to determine a feasible schedule and route which may accommodate as many patients as possible. The use of mobile phones in poor and remote areas of South Africa provides unique opportunities to improve healthcare access [8].

In the following section, different models are discussed that support the operational management of mobile clinics. In §3, a novel mathematical programming formulation is proposed to deal with a semi-flexible operational model when managing a mobile clinic service. The proposed model is a combination of the well-known *resource constrained project scheduling problem* and the *facility location problem.* The proposed model is applied to a fictitious problem instance in §4 for the purpose of demonstrating the behaviour of the model. The paper is concluded in §5.

2 Operational models for mobile clinic services

In this study, three different operational models are considered. In the following subsections, a short description of each conceptual model is provided, and possible advantages and disadvantages are discussed.

2.1 A fixed schedule operational model

Several countries are busy implementing mobile clinics within their cities as well as rural areas [2, 5]. Their operational models are typically based on a fixed schedule and routing plan for a week. It is, therefore, the patient's responsibility to ensure that they arrive at the predetermined time and location to receive medical treatment from the mobile clinic. The approach typically followed to set up the service schedule is based on historical patient data, and it uses a cost and distance analysis to determine the optimal route. This model does not use any patient inputs to determine the schedule and routing of mobile clinics.

The Yale Clinical & Community Research department implemented an initiative called the Community Health Care Van, which makes use of a fixed schedule routing model. Patients have access to the schedule of the Healthcare Van via their website [5]. Furthermore, in 2007, the Global AIDS Interfaith Alliance obtained funding from the Elizabeth Taylor HIV/AIDS Foundation to deploy two mobile clinics in Malawi. Their operational model is based on a fixed scheduling and routing approach [2].

Some of the advantages of using a fixed schedule operational model are, for instance, simplified operational planning and a predictable schedule that allows patients to know when and where a mobile clinic will be to provide medical services. The disadvantage, however, is that the approach does not take into account the availability of patients according to their own circumstances and, furthermore, it has no way of prioritising patient cases. It may happen, for instance, that mobile clinics are stationed where no one needs medical assistance, thus wasting time and resources.

2.2 A user-driven operational model

A user-driven operational model provides more flexibility to patients in terms of the treatment locations and the time of availability of healthcare services. That is, the final schedule and routing of a mobile clinic are based entirely on patient demand information. The treatment locations may vary from day to day, depending on the ability of a patient to be present at a particular place during a specific time of the day. The time spent at each treatment location may also vary, and multiple stop locations within a day could also be a possibility. The viability of implementing a user-driven operational model relies on the assumption that the healthcare service provider has access to the demand information provided by patients.

The AitaHealth initiative in South Africa provides a platform for collecting patient information within rural communities [3]. The current use of the AitaHealth mobile phone application is to provide community-oriented healthcare services. The mobile phone application is used by community health workers who are responsible for visiting patients on a frequent basis. The responsibilities of the community health workers may include the delivery of medication and to update patient's information on the AitaHealth database. The success of the AitaHealth initiative has been celebrated by registering one million patients on their database by 2018. This is an indication that a user-driven operational model is plausible, provided that a mobile phone application such as AitaHealth is available to patients. Additional functionality may be added to the application for patients to enter their preferences towards mobile treatment locations and appointment times. Although this type of operational model may be perceived as placing more of a logistical burden on the healthcare service provider, the use of mathematical modelling approaches and information technology could support the cost-efficient implementation of a user-driven model.

The major advantage of a fully user-driven operational model is that there is complete control over the prioritising of patient cases. More specifically, mobile clinics will only be stationed where patients need medical assistance, thus always using time and resources effectively. Patients can request their appointment time most suitable to their schedule. The disadvantage of this operational model is that it will be data-intensive and require solving complex optimisation problems. Furthermore, high initial capital layout may be expected to implement the supporting information technology.

2.3 A semi-flexible operational model

The two operational models introduced above are extremes to the spectrum of possible operational models. In proposing a semi-flexible operational model, an attempt is made to compromise between a low cost fixed schedule and a more expensive fully user-driven operational model. A semi-flexible model will still allow patients to specify their preferences towards appointment times, but with a fixed list of treatment locations. The time and duration spend by a mobile clinic at a treatment location will be based on the patient's ability to be at a specific location during a preferred time slot. Patients will be allowed to indicate their preference for multiple treatment locations during different time periods. For example, it may be possible for a patient to be at a specific treatment location during the morning on the way to work, or at another treatment location in the afternoon while returning from work. By applying a tailored optimisation model, a schedule may be calculated that maximises the overall patient preference to treatment locations and appointment times.

The anticipated advantage of a semi-flexible operational model is that a list of predetermined treatment locations is used for providing medical services. The time and duration of delivering the services at each location will be based, however, on patient demand information. By applying an optimisation model, optimal use of resources may be achieved while still taking patient demand information into account. Compared to the fully user-driven operational model, the disadvantage of this approach is that not all patients may be accommodated during a first round of scheduling. Patients will have to be informed when their preferred treatment locations and appointment times are infeasible in order to provide them with the opportunity to make another appointment.

3 An optimisation model for a semi-flexible service

The optimisation model proposed in this paper to support a mobile healthcare service is based on a semi-flexible operational model. The basic assumption for this model is that a list of predetermined treatment locations is available to the patients. A patient will have the opportunity to indicate his or her preference towards making an appointment at one or more of these locations — in advance — by using a mobile phone application. The routing of the mobile clinics between the different locations and the time spent at each location will be based on patient preferences as well as the availability of resources.

The formulation of the proposed optimisation model is based on a combination of a *facility location problem* and a *resource constrained project scheduling problem*. For the latter, the time-indexed formulation of Pritsker *et al.* [7] is adopted. Within the context of the resource constraint scheduling problem, the project activities correspond to the time spent by a mobile clinic at a predefined treatment location. More than one activity is, however, associated with a treatment location since a mobile clinic may visit it more than once during a week. Consider an example with three possible treatment locations, namely, A, B and C. The order in which these locations may be visited is determined as part of the solution. Figure 1 illustrates how, over a two-day schedule, more than one activity is associated with each location. For instance, **activity 1** is associated with **location A** during **day 1**, and **activity 5** is associated with **location A** during **day 2**. The arrows in Figure 1 implies a certain sequencing order. For example, **activities 1–3** during **day 1** may be scheduled in any order, as long as they complete before **activity 4** (the clinic). This implies that the mobile unit has to return to the clinic at the end of the day.

Let $\mathcal{I} = \{0, 1, \ldots, |\mathcal{I}|\}$ be the index set of activities which are associated with locations where healthcare services will be provided. The breakdown of the activities per scheduling day, as illustrated in Figure 1, is facilitated by a non-cyclic precedence graph $H(\mathcal{I}, \mathcal{Z})$, with

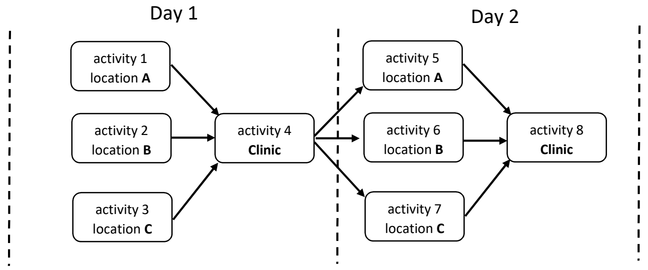


Figure 1: *Associating scheduling activities with mobile treatment locations A, B and C over a two-day schedule.*

vertex set $\mathcal{I}$ and arc set $\mathcal{A}$. The requirement of stopping at location i prior to stopping at location j is enforced by including the arc (i, j) in the arc set $\mathcal{A}$.

When a mobile clinic is stationary at one of the predetermined treatment locations, resources are consumed while providing the required healthcare services. These resources may include, for instance, the medical staff, medication, medical testing kits and other medical equipment. The set of resources are denoted by $\mathcal{R}$. Let v_{ir} be a numerical value for the quantity of resource $r \in \mathcal{R}$ consumed per hour at a location $i \in \mathcal{I}$, and let h_{rij} be the expected delay when transferring a resource $r \in \mathcal{R}$ from treatment location $i \in \mathcal{I}$ to $j \in \mathcal{I}$. The amount of resources available per hour for a resource $r \in \mathcal{R}$ is given by U_r.

Let $\mathcal{T} = \{1, 2, \ldots, |\mathcal{T}|\}$ be the set of time periods. The duration of each time period $t \in \mathcal{T}$ is considered to be, for example, one hour. The binary decision variable $x_{it} \in \{0, 1\}$ is used to indicate whether a mobile clinic is scheduled to stop at a location $i \in \mathcal{I}$ during time period $t \in \mathcal{T}$. The total duration (in hours) required of a mobile clinic to stop at a predetermined location $i \in \mathcal{I}$ to allow for patient appointments, is specified by d_i. The capacity c_i of a mobile clinic is defined as the maximum number of patients treatable during a time period $t \in \mathcal{T}$. The decision variable $z_{ij} \in \{0, 1\}$ is introduced to facilitate the sequencing order of the treatment locations. More specifically, if $z_{ij} = 1$, then the treatment location j is allowed to be visited after completion of treatment service at location i.

The set of patients is denoted by the index set $\mathcal{P} = \{1, 2, \ldots, |\mathcal{P}|\}$. Within the context of the facility location problem, each location where a mobile clinic has to stop to provide healthcare services is considered to be a facility. Therefore, the binary decision variable $y_{ip} \in \{0, 1\}$ is used to determine whether a mobile clinic at location $i \in \mathcal{I}$ will be providing a healthcare service to patient $p \in \mathcal{P}$. Each patient $p \in \mathcal{P}$ has a preference level l_{ip} towards receiving medical care at a location $i \in \mathcal{I}$. Greater values of l_{ip} will indicate higher preferences towards a specific location. In addition to a preference towards specific locations, each patient $p \in \mathcal{P}$ also has to provide a preferred timeslot $[E_{ip}, L_{ip}]$ for an appointment, with E_{ip} and L_{ip} the earliest and the latest available time for an appointment at a location $i \in \mathcal{I}$.

The objective of the proposed optimisation model is to

$$\text{maximise} \quad \sum_{i\in\mathcal{I}}\sum_{p\in\mathcal{P}} l_{ip}y_{ip}, \tag{1}$$

subject to the constraints

$$\begin{aligned}
\sum_{t\in\mathcal{T}} x_{it} &= 1, & i \in \mathcal{I}, && (2)\\
\sum_{t\in\mathcal{T}} tx_{it} - \sum_{t\in\mathcal{T}} tx_{jt} &\leq -di, & (i,j) \in \mathcal{A}, && (3)\\
z_{ij} + z_{ji} &= 1, & i \in \mathcal{I}, j \in \mathcal{I}, i \neq j, && (4)\\
\sum_{t\in\mathcal{T}} tx_{it} - \sum_{t\in\mathcal{T}} tx_{jt} + (M + d_i + h_{rij})z_{ij} &\leq M, & r \in \mathcal{R}, i \in \mathcal{I}, j \in \mathcal{I}, && (5)\\
\sum_{i\in\mathcal{I}} \sum_{k=t-d_i+1}^{t} v_{ir}x_{ik} &\leq U_r, & r \in \mathcal{R},\ t \in \mathcal{T}, && (6)\\
\sum_{i\in\mathcal{I}} y_{ip} &\leq 1, & p \in \mathcal{P}, && (7)\\
\sum_{p\in\mathcal{P}} y_{ip} &\leq c_i, & i \in \mathcal{I}, && (8)\\
\sum_{t\in\mathcal{T}} tx_{it} + (M - E_{ip})y_{ip} &\leq M, & i \in \mathcal{I},\ p \in \mathcal{P}, && (9)\\
-\sum_{t\in\mathcal{T}} tx_{it} + L_{ip}y_{ip} - d_i &\leq 0, & i \in \mathcal{I},\ p \in \mathcal{P}. && (10)
\end{aligned}$$

The objective function (1) maximises the overall patient preferences. Constraints (2)–(6) are the usual constraints found in the formulation of the standard resource constrained scheduling problem. Constraint set (2) ensures that healthcare services are scheduled at each location. Constraint set (3) applies the routing policy of the mobile clinics based on the precedence graph $H(\mathcal{I}, \mathcal{A})$. Together, constraint sets (4) and (5) enforce the expected delay in transferring a resource $r \in \mathcal{R}$ from a treatment location $i \in \mathcal{I}$ to $j \in \mathcal{I}$. The resource availability for each time period $t \in \mathcal{T}$ is modelled by constraint set (6).

The remaining constraints in the problem formulation are responsible for selecting the optimal location that a patient is assigned to for receiving medical treatment. Constraint set (7) is required to model the assignment of a patient to at most one location. The maximum number of patients treatable at a specific location associated with an activity $i \in \mathcal{I}$, is formulated according to constraint set (8). A higher value of the parameter c_i in constraint set (8) implies that more patients may be treated during a time period $t \in \mathcal{T}$. Constraint sets (9) and (10) allow a patient to be treated at a specific location and appointment time, provided the associated activity is scheduled to overlap the preferred time slot of the patient.

	Max 2 patients per hour			Max 4 patients per hour			Max 6 patients per hour		
	Location	Start	End	Location	Star	End	Location	Start	End
Day 1	C	08:00	10:00	C	08:00	10:00	B	08:30	10:30
	A	11:00	13:00	A	11:30	13:30	A	11:30	13:30
	B	14:00	16:00	B	14:30	16:30	C	14:30	16:30
Day 2	A	08:30	10:30	A	08:30	10:30	A	08:30	10:30
	B	13:00	15:00	B	11:30	13:30	B	11:30	13:30
	C	16:00	18:00	C	18:00	20:00	C	16:30	18:30

Table 1: *Results for a fictitious problem instance having three treatment locations and thirty-two patients.*

4 Model behaviour and computational results

A fictitious data set was created to demonstrate the application of the proposed optimisation model for determining a semi-flexible operational service schedule. Although the proposed problem formulation above may accommodate several resources, the only resource included in the fictitious problem instance is the availability of a single mobile clinic. In a real-life application, the proposed model may be applied to find an optimal service schedule that includes more than one mobile clinic as well as multiple resources.

For illustrative purposes, only three treatment locations and a total of thirty-two patients were included in the data set. The time spent at each treatment location by the mobile clinic (given by the parameter d_i), is fixed at two hours. The travel times between the three locations (captured as the transfer delay parameter h_{rij}) are assumed to be an hour between locations A and B, half an hour between B and C, and an hour between locations A and C. It is assumed that each patient may have certain preferences towards specific treatment locations as well as appointment times. Therefore, for each treatment location associated with an activity $i \in \mathcal{I}$ and patient $p \in \mathcal{P}$, the parameters l_{ip}, E_{ip} and L_{ip} were generated randomly.

The results obtained by applying the proposed optimisation model to the fictitious problem instance is provided in Table 1. The problem was solved by specifying a maximum number of patients per treatment location of two, four and six, respectively. By doing this, three sets of results were obtained. For each set, the order in which the different treatment locations has to be visited are indicated by the associated start and end times. For instance, for a maximum of two patients per hour, the mobile clinic has to be at **location C** from 08:00 to 10:00 during the first day of the schedule, then at **location A** from 11:00 to 13:00 and, finally, at **location B** from 14:00 to 16:00, before returning back to the clinic. The following day, the mobile clinic has to be at **location A** from 08:30 to 10:30, then at **location B** from 13:00 to 15:00 and, finally, at **location C** from 14:00 to 16:00.

Although the objective of the problem is to maximise the overall preference of the patients, some patients may not get their first choice or may not be accommodated at all. For instance, in the case where a maximum of only two patients is treated at a specific location, not all of the thirty-two patients could be accommodated during the two-day schedule. More specifically, the proposed treatment schedule indicates that a total of twenty-three patients will not be accommodated.

For the case where a maximum of four patients is treated per hour, a total of seventeen patients will not be accommodated by the suggested treatment schedule. When a maximum of six patients is allowed per treatment location, eleven patients will not be accommodated by the treatment schedule.

From Table 1, it is observed that idle times are present in the proposed schedule. For example, in the case where a maximum of two patients is treated per hour, the allocated treatment times at **locations A** and **B** during **day 2** are from 08:30 to 10:30 and from 13:00 to 15:00, respectively. Since the travel time from **location A** to **B** is only one hour, there is an idle period of an hour and a half. This is due to the objective function that attempts to maximise the overall patient preferences, without taking into account the possible waste in resources.

5 Conclusion

In this paper, a mathematical programming formulation is proposed to deal with the operational management of a mobile healthcare service. The problem formulation is based on a semi-flexible operational model, which allows patients to specify their preferences towards both the location and the time of appointment for receiving medical treatment. The objective of the optimisation model is to maximise overall patient preference while taking into account resource availability. Results for a fictitious problem instance were presented for the purpose of demonstrating the model behaviour when considering different levels of treatment capacities.

This study is still work in progress and extensions to the model proposed in this paper may include, *e.g.,* a more flexible model in which the time spent at a treatment location is determined as part of the solution, and not taken as a fixed input parameter. Furthermore, a more realistic data set may shed light on the appropriateness of the model assumptions, as well as the computational requirements of the solution approach.

References

[1] ILA Trust, 2019, *ILA Trust background*, [Online], [Cited January 29th, 2019], Available from `http://www.ilatrust.org/background.php`

[2] Lindgren TG, Deutsch K, Schell E, Bvumbwe A, Hart KB, Laviwa J & Rankin SH, 2011, *Using mobile clinics to deliver HIV testing and other basic health services in rural Malawi*, [Online Serial], Rural and Remote Health, **11(2)**, [Cited January 29th, 2019], Available from `www.rrh.org.au/journal/article/1682`

[3] Mezzanine, 2019, *AitaHealth*, [Online], [Cited January 29th, 2019], Available from `https://www.mezzanineware.com/aitahealth/`

[4] Mokomane Z, Mokhele T, Mathews C & Makoae M, 2017, *Availability and accessibility of public health services for adolescents and young people in South Africa*, Children and Youth Services Review, **74**, pp. 125–132.

[5] Morano J, Zelenev A, Walton M, Bruce R & Altice F, 2014, *Latent tuberculosis infection screening in foreign-born populations: A successful mobile clinic outreach model*, American Journal of Public Health, **104(8)**, pp. 1508–1515.

[6] Neely AH & Ponshunmugam A, 2019, *A qualitative approach to examining health care access in rural South Africa*, Social Science & Medicine, **230**, pp. 214–221.

[7] Pritsker A, Alan B, Waiters LJ & Wolfe PM, 1969, *Multiproject scheduling with limited resources: A zero-one programming approach*, Management Science, **16(1)**, pp. 93–108.

[8] Anstey Watkins JOT, Goudge J, Gómez-Olivé FX & Griffiths F, 2018, *Mobile phone use among patients and health workers to enhance primary healthcare: A qualitative study in rural South Africa*, Social Science & Medicine, **198**, pp. 139–147.

Proceedings of the 48th ORSSA Annual Conference
pp. 72–79
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3
©2019

Resource constrained project scheduling approaches to fibre network deployment

Q van Riet[*] MJ Grobler[*] SE Terblanche[†]

Abstract

Optical fibre networks are currently considered the most suitable option for providing high-speed bandwidth for both access and long haul data networks. Despite the decrease in optical fibre costs, deploying fibre networks is still considered expensive. Existing project management tools do not take complex objective functions into account and are unable to cater for telecommunication-specific side constraints. By formulating the fibre network deployment planning problem as a general *resource constrained project scheduling problem* (RCPSP), various solution approaches may be followed to generate solutions. Computational results, which are based on benchmark instances from the literature, demonstrate the practical application of constraint programming and mixed integer linear programming in solving the fibre network deployment planning problem.

Key words: Fibre network deployment, resource constrained project scheduling, mixed integer linear programming, constraint programming.

1 Introduction

Fibre optic deployment typically involves two separate activities — the first step involves the design of the fibre network, followed by the physical deployment. Considerable work has been done in the last decade to provide the telecommunications industry with automated fibre network design tools [11]. A typical objective of fibre network design models is to provide the least cost network design, based on demand projections and topology information. With an “optimal” network design at hand, the next step is to put a project management plan in place for the physical deployment.

Fibre uptake by businesses as well as home users is rapidly increasing due to the ability of fibre to provide high-speed bandwidth for both access and long haul data networks. The challenge faced by operators, however, is that the payback period for the operator’s investments is too long. The critical issue for operators is to reduce fibre deployment cost through improved resource allocation and project scheduling.

[*] School of Electrical, Electronic and Computer Engineering, North-West University, Republic of South Africa, email: quinton@quintonvr.com, leenta.grobler@nwu.ac.za

[†] School of Industrial Engineering, North-West University, Republic of South Africa, email: fanie.terblanche@nwu.ac.za

If fibre deployment companies can improve their deployment schedules based on *net present value* (NPV), it would lead to financial benefit to the network operator as well as improve end-users satisfaction. In a publication by Ding [3], reference is made to a study by the *Iternational Telecomunnications Union* (ITU) that states "if broadband penetration increases by ten percentage points, GDP rises by 1.3 percent, employment rises by two to three percent, productivity increases five to ten percent, and innovation rockets fifteenfold. At the same time, greenhouse gas emissions fall by five percent."

The objective of this paper is to formulate the fibre network deployment planning problem as a general *resource constrained project scheduling problem* (RCPSP). Solving an RCPSP involves the scheduling of project activities over time such that the total resources consumed by the activities do not exceed the resource availability. An additional requirement is that activities must be scheduled according to predefined sequencing rules, which are typically captured by a so-called precedence graph. Various model formulations and algorithmic approaches for solving the RCPSP exist in the literature, which may affect the quality of the solutions obtained as well as computational efficiency.

2 Technical background

Although different architectures may be considered for the deployment of an optical fibre network, a *passive optical network* (PON) is considered the most suitable since it has no active devices requiring electric power. PON architectures are popular for the deployment of *fibre-to-the-home* (FTTH). Figure 1 is an illustration of a typical PON configuration. A PON consists of a *central office* (CO), which is connected to several splitters. Each splitter requires a separate fibre to be connected to the CO. The splitters then split the signal coming from the CO to allow multiple fibres to be connected to *optical network units* (ONUs), which are the demand points in the fibre network that represent the end-users. In order to save on trenching costs, operators may place fibres in the same conduit — a technique known as fibre duct sharing.

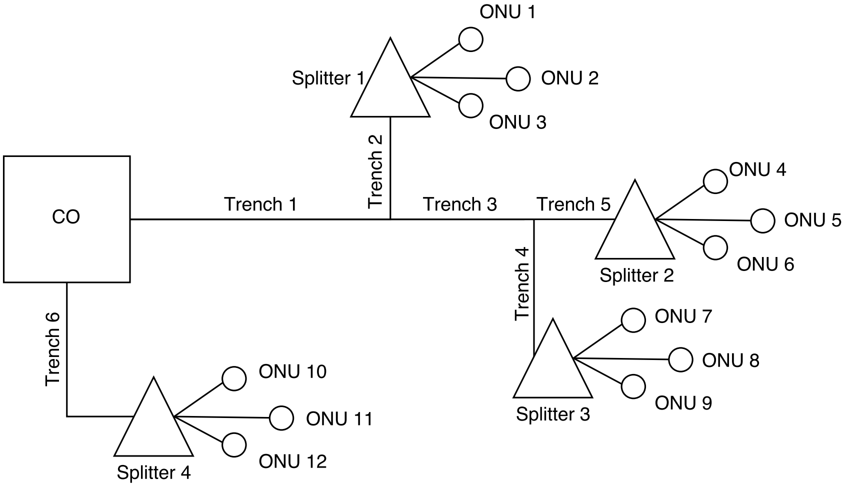


Figure 1: *Elements of a simple fibre network.*

Telecommunication companies typically formulate deployment plans by applying techno-economic analysis. Techno-economic analysis evaluates the technical requirements of the project (*e.g.,* amount of fibre, splitters, optimal network design, *etc.*) as well as the economic viability thereof (*e.g.,* cost and profitability).

Azodolmolky & Tomkos [2] developed a techno-economic model to assist network planners and service providers in selecting a deployment strategy and network design. Their model provides high-level insight into Ethernet *fibre-to-the-business* (FTTB) deployment through *capital expenditure* (CAPEX) and *operating expenditure* (OPEX). The results are based on a case study for an Ethernet FTTB deployment in Athens, Greece.

Jerman-Blazic [4] compared and evaluated suitable methods for delivering broadband services at municipal and backbone level. The model is based on a network value analysis that also involves CAPEX and OPEX calculations. The financial assessment for technology deployment is reflected through techno-economic evaluations such as NPV. The results are based on a case study for the Telekom Slovenia company and deals with upgrading the current network backbone while considering future development and trends of broadband services in the country.

Further work by Kampouridis *et al.* [5] provided a framework that encapsulates existing techno-economic models by using a *genetic algorithm* (GA) as a decision support tool when it comes to deciding which areas fibre networks should be deployed to first. Although this model searches for deployment areas that return the highest profit, it does not attempt to further increase profit by scheduling the order in which activities need to be executed when examining a specific deployment area.

In fibre network deployment scheduling, the *project activities* involve the construction of a central office; the excavation of trenches; and the installation of conduits, fibre, splitters, and ONUs. Other time-consuming activities include the splicing and testing of fibre cables. The *renewable resources* are the number of excavation tools; man-hours of the workforce; and the availability of fibre, splitters, and ONUs.

The cash flow associated with each activity may either be an expense (negative cash flow) or an income (positive cash flow). Within the context of fibre deployment, positive cash flows are produced by ONUs once the connection to the CO has been completed. When areas that contain high-value customers who have the desire to spend money on fibre, is supplied first, the uptake will be high and service providers can start producing revenue at the early stage of deployment. Negative cash flows are the costs associated with constructing the CO, trenching and installing the fibres, the splitters and the ONU equipment.

By using the above information on a fibre network deployment project, an RCPSP approach may be applied to obtain a deployment schedule that adheres to resource and sequencing constraints while maximising NPV.

3 Resource constrained project scheduling

In general, a project comprises activities indexed by the set $\mathcal{N} = \{1, 2, ..., |\mathcal{N}|\}$. The project is completed when all activities have been processed. Each activity $i \in \mathcal{N}$ has an expected processing time d_i. There is a set $\mathcal{R}$ of renewable resources that may be used during the processing of the activities. The resources are called renewable because their full capacity is available in every time period $t \in \mathcal{T} = \{1, 2, \ldots, |\mathcal{T}|\}$. While being processed, activity $i \in \mathcal{N}$ requires some quantity v_{ir} of resource $r \in \mathcal{R}$. Resource r has a limited instantaneous capacity of U_r during the processing of activities. A so-called precedence constraint determines the order in which activities may be processed. For this purpose, a non-cyclic graph $G(\mathcal{N}, \mathcal{Z})$ is defined, where $\mathcal{N}$ is the set of vertices and $\mathcal{Z}$ the set of arcs. The arc $(i, j) \in \mathcal{Z}$ states the precedence requirement that activity $j \in \mathcal{N}$ is preceded by activity $i \in \mathcal{N}$.

Let $s_i \geq 0$ denote the starting time of an activity $i \in \mathcal{N}$. Note that the starting time s_i does not necessarily have to coincide with the start times of the scheduling periods, and the duration d_i may not necessarily be divisible by the specified period duration. Therefore, to facilitate the conceptual formulation of the RCPSP, the function $\phi(i, t, s_i)$ is used to calculate the proportion of resources being consumed by activity $i \in \mathcal{N}$ during time period $t \in \mathcal{T}$, if the activity is scheduled to start at time s_i. The implementation details of the function $\phi(i, t, s_i)$ will depend on the solution approach adopted. Therefore, the model provided below is only a conceptual model, and the solution approaches described in the next section will dictate the explicit implementation of the function $\phi(i, t, s_i)$.

If c_i represents the cash flow associated with an activity $i \in \mathcal{N}$, and α is the NPV discount rate, the objective function of the RCPSP is to

$$\text{maximise} \sum_{i \in \mathcal{N}} e^{-\alpha s_i} c_i, \tag{1}$$

subject to

$$\begin{aligned} s_i - s_j &\leq -d_i, & (i, j) \in \mathcal{Z}, \quad (2) \\ \sum_{i \in \mathcal{N}} \phi(i, t, s_i) v_{ir} &\leq U_r, & r \in \mathcal{R},\ t \in \mathcal{T}. \quad (3) \end{aligned}$$

The objective function (1) maximises NPV based on the continuous compound application of the discount rate α. Constraint set (2) enforces the precedence requirement according to the graph $G(\mathcal{N}, \mathcal{Z})$, and constraint set (3) limits the resource consumption per time period.

4 Solution approaches

Several algorithmic approaches may be followed to generate solutions for the conceptual RCPSP formulation provided in the previous section. In this paper, the computational efficiency when employing both a *constraint programming* (CP) and a *mixed integer linear programming* (MILP) approach, is investigated. Although the implementation of a CP model is typically technology-dependent, there are three main components to formulating the RCPSP as a CP [8]:

- Interval variables — decision variables in CP that represent an interval of time relating to an activity in a schedule. In a CP model, the variables s_i in the conceptual model (1)–(3) above are defined as interval variables.

- Precedence constraints — constraints in CP expressed in terms of interval variables which ensure the relative position of the activities in the scheduling solution. Constraint set (2) in the conceptual model above are implemented in CP as precedence constraints.
- Cumulative expression constraints — constraints in CP that limits the accumulation of resource usage over time. Constraint set (3) in the conceptual model above are implemented in CP as Cumulative expression constraints.

In this paper, the commercial solver CP Optimiser, a product by IBM, was employed to solve the fibre deployment planning problem as an RCPSP.

Several formulations of the RCPSP as a MILP are provided in the literature. The most popular formulations include a time index formulation [9], an event-based formulation [7] and a resource flow formulation [1]. A time-indexed formulation of the RCPSP was adopted by Van Riet, Terblanche & Grobler [12] to solve the resource allocation and scheduling of fibre deployment projects. In this paper, the resource flow formulation by Terblanche [10] is adopted due to the duration characteristics of some of the fibre deployment project activities. More specifically, trenching activities may take up to several weeks to complete, depending on civil and infrastructure restrictions. Empirical results provided by Terblanche [10] demonstrate that the resource flow formulation of the RCPSP is computationally more efficient when solving problem instances that are characterised by activities with extended completion times. The MILP solver, IBM CPLEX, was employed in this study to solve the resource flow formulation of the fibre deployment planning problem.

5 Computational study on benchmark instances

In order to validate the application of an RCPSP model to solve the fibre deployment planning problem, the Dassierand suburb of Potchefstroom in South Africa is considered as an example of a fibre deployment project. Figure 2 shows the fibre network configuration for the suburb, comprising 300 ONUs that are connected to a central office through ten different splitters. A total of 632 activities are associated with the corresponding deployment plan.

The resources that were available for the execution of the deployment plan include man-hours of the workforce, the availability of fibre, splitters, and ONUs. The project activities with a negative cash flow entail the construction of a central office; the excavation of trenches; and the installation of conduits, fibre, splitters, and ONUs. Positive cash flows are considered to be the revenue that ONUs produce once a single path from the CO through to one of the ONUs has been completed.

The Dassierand deployment planning problem was solved on an Intel Core i5-3470 processor with four cores operating at 3.2 GHz and 8GB RAM. As a first attempt to obtain a solution, the resource flow formulation of the RCPSP was solved with the commercial MILP solver, CPLEX. Due to the computational complexity of the problem, optimality was not reached after ten hours of computing time, and only a feasible solution could be computed. However, by employing CP to solve the problem, an improvement of 60% in objective function value was achieved.

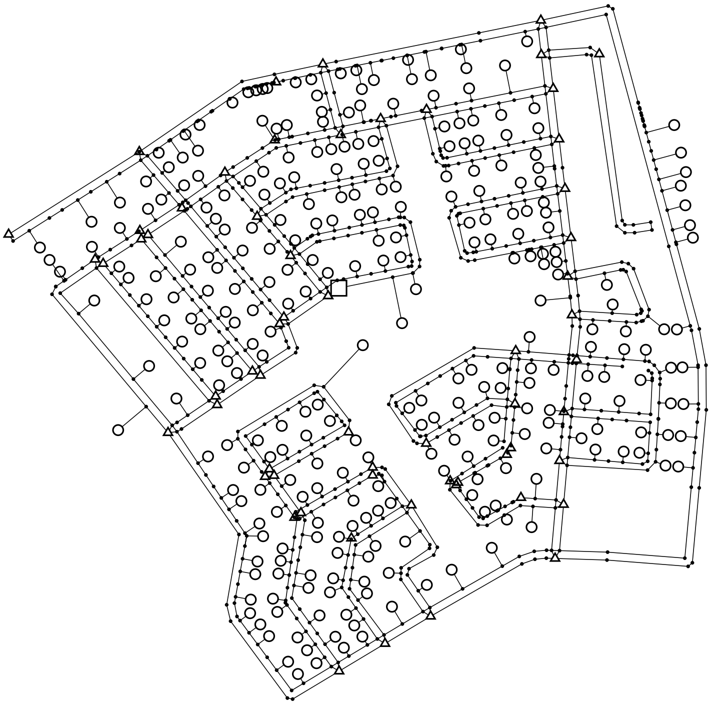


Figure 2: *PON topology input for the Dassierand dataset.*

In order to assess the computational efficiency of applying CP in general, a comparative study between MILP and CP was conducted by considering benchmark RCPSP problem instances from the literature. For this purpose, the *project scheduling problem library* (PSPLIB) [6] was considered. The PSPLIB repository contains three sets of data, namely, j30, j60 and j90. The data sets contain RCPSP instances comprising thirty, sixty and ninety activities, respectively. There are 480 instances in each set, which all differ in complexity. Each RCPSP instance includes a set of four resources.

The PSPLIB problem instances were modified with longer activity durations in order to be more reflective of the type of activities to be found in a fibre deployment planning problem. The duration d_i of all activities were adjusted by a factor of ten, *i.e.*, $d_i = 10d_i$. The adjusted datasets are referred to below as the j30(10x), j60(10x) and j90(10x).

The time limit imposed on each computational run was based on the number of activities of the problem instance being solved. That is, a total of ten seconds of computing time was allocated to each activity in the problem instance. For example, the j30(10x) dataset, where each of the 480 problem instances included thirty activities, the total computing time allowed for each of the j30 instances was 300 seconds.

As part of the comparative study on the computational efficiency between a MILP and a CP approach, a hybrid approach combining the two approaches was also considered. More specifically, for each of the problem instances solved, CP was used to generate feasible solutions, which in turn were then applied within the MILP approach as initial starting solutions. The hybrid approach is set within the exact framework of MILP and, therefore, provides a quality guarantee on solutions computed.

Dataset	Number of activities	Approach	Avg. NPV	Avg. solution time(s)	Feasible solution instances (%)
j30(10x)	30	MILP	540.93	43.24	100.00
		CP	535.07	0.12	100.00
		Hybrid	541.01	19.87	100.00
j60(10x)	60	MILP	573.18	258.26	95.00
		CP	546.01	0.62	100.00
		Hybrid	580.75	183.45	100.00
j90(10x)	90	MILP	497.88	600.89	51.46
		CP	556.39	1.42	100.00
		Hybrid	593.11	890.22	100.00

Table 1: *The maximisation of NPV on the extended activity duration datasets — resource flow formulation.*

Table 1 provides information on the average solution times and the percentage of instances for which a feasible solution could be computed within the allocated time limit, when considering the application of MILP, CP and the Hybrid approach. As a first observation, it is clear that the average computing time when applying only MILP is significantly higher than CP, but CP on its own does not perform that well in maximising NPV, except for the j90(10x) case. The Hybrid approach is successful in reducing computing time when compared to MILP, and improving the NPV when compared to CP, for both the j30(10x) and j60(10x) problem instances. Although the Hybrid approach does improve NPV for the j90(10x) data set, it is not successful in reducing computing times when compared to the MILP.

6 Conclusion

In this paper, the fibre deployment planning problem is formulated as a general *resource constrained project scheduling problem* (RCPSP). A conceptual model formulation was presented, which was implemented as both a constraint programming model and a mixed integer linear programming model. Validation of the two solution approaches was performed by considering a real-world data set. Computational results were reported to demonstrate the use of constraint programming as a warm-start heuristic for solving the fibre deployment planning problem within an exact mixed integer linear programming framework. Solving the fibre deployment planning problem within an exact framework has the benefit that quality guarantees can be obtained for the computed solutions.

References

[1] Artigues C, Michelon P & Reusser S, 2003, *Insertion techniques for static and dynamic resource-constrained project scheduling*, European Journal of Operational Research, **149(2)**, pp. 249–267.

[2] Azodolmolky S & Tomkos I, 2008, *Techno-economic study of a modeled active Ethernet FTTB deployment*, Proceedings of the 6th International Symposium on Communication Systems, Networks and Digital Signal Processing, Graz, pp. 496–499.

[3] Ding R, 2017, *Fiber networks: Faster payback means better connected*, Huawei Publications, **28**, pp. 22–27.

[4] Jerman-Blažič B, 2007, *Comparative study and techno-economic analysis of broadband backbone upgrading: A case study*, Informatica (Ljubljana), **31(3)**, pp. 279–284.

[5] Kampouridis M, Glover T, Shaghaghi AR & Tsang E, 2012, *Using a genetic algorithm as a decision support tool for the deployment of fiber optic networks*, Proceedings of the 2012 IEEE Congress on Evolutionary Computation, Brisbane, pp. 2484–2491.

[6] Kolisch R & Sprecher A, 1997, *PSPLIB — A project scheduling problem library: OR software — ORSEP Operations Research Software Exchange Program*, European Journal of Operational Research, **96(1)**, pp. 205–216.

[7] Koné O, Artigues C, Lopez P & Mongeau M, 2013, *Comparison of mixed integer linear programming models for the resource-constrained project scheduling problem with consumption and production of resources*, Flexible Services and Manufacturing Journal, **25(1-2)**, pp. 25–47.

[8] Laborie P, 2009, *IBM ILOG CP optimizer for detailed scheduling illustrated on three problems*, pp. 148–162 in van Hoeve WJ & Hooker JN (Eds.), *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Springer, Heidelberg.

[9] Pritsker AAB and Waiters LJ & Wolfe PM, 1969, *Multiproject scheduling with limited resources: A zero-one programming approach*, Management Science, **16(1)**, pp. 93–108.

[10] Terblanche SE, 2017, *Resource constrained project scheduling models and algorithms applied to underground mining*, PhD Dissertation, Stellenbosch University, Stellenbosch.

[11] Van Loggerenberg SP, Ferreira M, Grobler MJ & Terblanche SE, 2015, *PON-OPT: A passive optical network design optimisation tool*, Proceedings of the Southern African Telecommunication Networks and Applications Conference (SATNAC) 2015, Hermanus, pp. 25–30.

[12] Van Riet Q, Terblanche SE & Grobler MJ, 2018, *Resource allocation and scheduling of fibre network deployment*, Proceedings of the Southern African Telecommunication Networks and Applications Conference (SATNAC) 2018, Hermanus, pp. 418–423.

Proceedings of the 48th ORSSA Annual Conference
pp. 80–87
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3
©2019

The role of spatial spillover effects in relation to regional poverty at the EU

A Furková[*]

Abstract

This paper explores the role of spatial spillovers in the context of poverty and social exclusion across 141 *Nomenclature of Units for Territorial Statistics* (NUTS) 2 *European Union* (EU) regions for the 2016–2017 period. An attempt was made to model the behaviour of poverty and social exclusion at the EU regional level on the basis of two versions of spatial econometric models. Based on the quantifications and statistical verifications of the direct, indirect and total impacts of chosen indicators, it was found that the spatial spillovers of economic, social and demographic indicators among the EU regions do matter, *i.e.,* there is a link not only between these indicators and poverty and social exclusion within the region, but that there are spillovers to neighbouring regions.

Key words: Poverty and social exclusion, spatial spillovers, spatial autocorrelation, spatial econometric model, NUTS 2 EU regions.

1 Introduction

Poverty reduction is one of the most challenging issues for economic development. The Europe 2020 strategy [3] promotes social inclusion, in particular through the reduction of poverty, by aiming for twenty million less people to be at risk of poverty and social exclusion. Fighting poverty is a part of inclusive growth declared by the *European Union* (EU) strategy 2020. Inclusive growth captures a high-employment economy delivering economic, social and territorial cohesion. It is also essential that the benefits of economic growth spread to all parts of the EU, including its outermost regions, thus strengthening territorial cohesion. Also, for the next long-term EU strategy 2021–2027, the topic of poverty is a part of a strategic goal — "a more Social Europe" — presented by the European Commission [4].

In 2017, there were 112.8 million people in the EU-28 who lived in households at risk of poverty or social exclusion, equating to 22.4% of the entire population. Indicator *At risk of poverty or social exclusion* includes people who were in at least one of the following

[*] Department of Operations Research and Econometrics, University of Economics in Bratislava, Dolnozemská cesta 1, 852 35 Bratislava, Slovakia, email: furkova.andrea@gmail.com. This work was supported by the Grant Agency of Slovak Republic – VEGA grant no. 1/0248/17 *Analysis of regional disparities in the EU based on spatial econometric approaches.*

situations: at risk of poverty after social transfers (income poverty), severely materially deprived, or living in households with very low work intensity [6]. We can notice some positive trends in the fight against poverty within the EU. If we compare years 2016 and 2017 (see Figure 1), the number of people at risk of poverty or social exclusion in 2017 had decreased by 5.1 million, equivalent to a 1.1 percentage point decrease in the share of the total population in the EU-28 (23.5%–22.4%). As such, the share of the EU-28 population at risk of poverty or social exclusion fell to a level that had not been recorded since data became available in 2010. Unfavourable information is that approximately a third of the population was still at risk of poverty or social exclusion in three EU member states (Bulgaria, Romania and Greece).

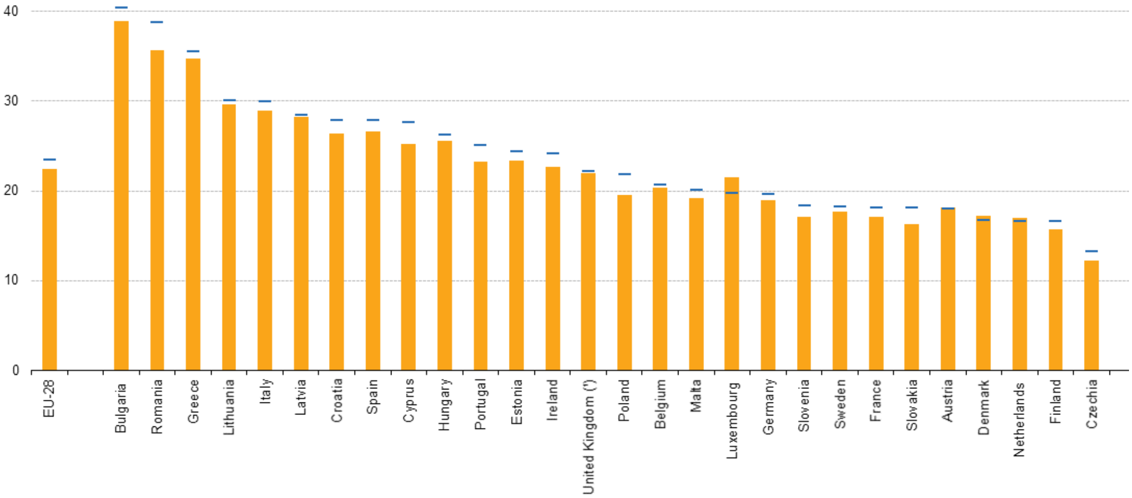


Figure 1: *At risk of poverty or social exclusion rate (share of total population) 2016 and 2017 [6].*

It has become clear that economic factors explain a large amount of variation in poverty (see, *e.g.,* Levernier *et al.* [10] who studied poverty across US counties). However, economic factors alone are not sufficient to explain nor reduce poverty. Other social, political, demographic and spatial factors should be taken into account. Generally, raising educational attainment levels is considered as one of the means of moving people out of poverty. On the other hand, Rupasingha & Goetz [12] point to that investments in human capital do not occur automatically if other complementary factors are not also in place. Certainly geographic variation and concentration of poverty rates remain a major aspect of the poverty problem. Goals formulated in the strategic documents of the EU and national social policies distinctly indicate the need of analysing poverty at regional and local levels. Regional differences and marginalization of some EU regions is one of the main areas of interest of the EU integration policies [11].

In this paper we incorporate economic, social, demographic, and spatial factors into one model that explains variation in poverty rates in the EU at the regional level. *At risk of poverty or social exclusion rate in 2017* was chosen as a dependant variable of the spatial econometric model. The aim will be to verify the role of potential *spatial spillover* *effects* (SSEs) in relation to regional poverty at the EU. Our main hypothesis is that the spatial spillovers of chosen economic, social and demographic indicators among the EU regions do matter, *i.e.,* there is a link not only between these indicators and poverty within the region, but that there are spillovers to neighbouring regions. The selected spatial econometric model will be used to quantify the direct, indirect and total impacts of chosen indicators. The rest of the paper is structured as follows: Section 2 provides brief theoretical backgrounds of the study. Data and empirical results are presented and interpreted in §3. The paper closes with concluding remarks in §4.

2 Methodology

The main methodological framework upon which our empirical analysis is based, will be very briefly presented in this section. Geographic variation and concentration of poverty rates seem to be a major aspect of the poverty problem. Generally, in situations where geographic location of regions matter, spatial autocorrelation is present. In order to avoid the misspecification problems due to spatial dependencies, tools of spatial statistics and spatial econometrics should be applied. A generalized version of the spatial econometric model is called the *General Nesting Spatial* (GNS) model. This model includes all types of spatial interaction effects, and the GNS model for cross-sectional data in matrix form can be written as follows:

$$\begin{aligned} &\mathbf{y} = \rho\mathbf{W}\mathbf{y} + \mathbf{X}\boldsymbol{\beta} + \mathbf{W}\mathbf{X}\boldsymbol{\gamma} + \mathbf{u}, \\ &\mathbf{u} = \lambda\mathbf{W}\mathbf{u} + \mathbf{v}, \qquad\qquad \mathbf{v} \sim \mathcal{N}(\mathbf{0},\, \sigma_v^2\mathbf{I}_N), \end{aligned} \tag{1}$$

where $\mathbf{y}$ represents the vector of the observed dependent variable for all N observations (locations); $\mathbf{X}$ denotes an $N \times k$ matrix of exogenous explanatory variables (k denotes the number of explanatory variables); $\boldsymbol{\beta}$ is a $k \times 1$ vector of unknown parameters to be estimated; u is an $N \times 1$ error vector which follows spatial autoregressive process $\mathbf{u} = \lambda\mathbf{W}\mathbf{u} + \mathbf{v}$; $\mathbf{v} \sim \mathcal{N}(\mathbf{0},\, \sigma_v^2\mathbf{I}_N)$ is an $N \times 1$ vector of random errors; σ_v^2 is the random error variance; $\mathbf{I}_N$ is an N-dimensional unit matrix; and $\mathbf{W}$ is an N-dimensional spatial weighting matrix (the issues related to the spatial weighting matrix — see, *e.g.,* Anselin & Rey [1] or Chocholatá [2]). Model (1) includes all types of interaction effects, namely endogenous interaction effects among the dependent variable ($\mathbf{W}\mathbf{y}$), the exogenous interaction effects among the independent variables ($\mathbf{W}\mathbf{X}$), and the interaction effects among the disturbance term of the different units ($\mathbf{W}\mathbf{u}$). Hence, the $k \times 1$ vector $\boldsymbol{\gamma}$, and parameters ρ and λ represent spatial autoregressive parameters; their statistical significance, value and mathematical character indicate the direction and the strength of spatial dependence [7]. Estimation of GNS models and other spatial autoregressive models require special estimation methods. A review of the models and estimation methods is presented, *e.g.,* by Anselin & Rey [1].

Since spatial regression models contain spatial lags of explanatory variables and/or dependent variables, interpretation of the parameters is not as straightforward as in linear regression models. The expected value of the dependent variable in the i^{th} location is no longer influenced only by exogenous location characteristics, but also by the exogenous characteristics of all other locations through a spatial multiplier $(\mathbf{I}_N - \rho\mathbf{W})^{-1} = \mathbf{I}_N + \rho\mathbf{W} + \rho^2\mathbf{W^2} + \rho^3\mathbf{W^3} + \ldots$. LeSage & Pace [9] suggested summary impact measures of the average total, direct and indirect impacts based on the $\mathbf{S_r}(\mathbf{W})$ [9]. Consider the *Spatial Autoregressive* (SAR) model[1], (Spatial Autoregressive) model which can be formulated as follows:

$$\mathbf{y} = \rho\mathbf{W}\mathbf{y} + \mathbf{l}_N\alpha + \mathbf{X}\boldsymbol{\beta} + \mathbf{u} \tag{2}$$

where $\mathbf{l}_N$ represents an $N \times 1$ vector of ones associated with the constant term α and all remaining components are as defined before. An overview of the average total, direct and indirect impacts for this model is shown in Table 1.

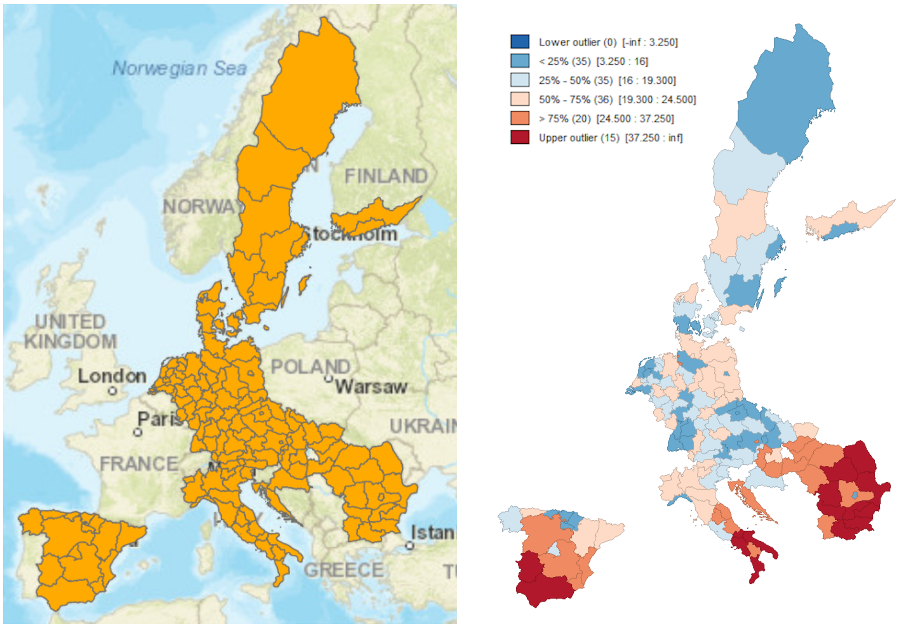
SAR model: $\mathbf{S_r}(\mathbf{W}) = (\mathbf{I}_N - \hat{\rho}\mathbf{W})^{-1}(\mathbf{I}_N\hat{\beta}_r)$	
Average total impact *(ATI)*	$ATI = N^{-1}\mathbf{l}_N^T\mathbf{S_r}(\mathbf{W})\mathbf{l}_N$
Average direct impact *(ADI)*	$ADI = N^{-1}tr(\mathbf{S_r}(\mathbf{W}))$
Average indirect impact *(AII)*	$AII = ATI - ADI$

Table 1: *Overview of the ATI, ADI and AII impact measures for the SAR model.*

3 Data and empirical results

The data set used in the empirical part of this paper is obtained from the regional Eurostat statistics database [5] and it covers 141 NUTS 2 EU regions from thirteen countries surveyed over the 2016–2017 period. The selection of the model variables and time period was influenced by significant lack of the data (especially poverty rates) for regions in the NUTS 2 structure. Another data set reduction had to be done due to isolated observations. In order to verify the research hypothesis (defined in the introduction of the paper), two spatial econometric models are used. The response variable is *Poverty* — people at risk of poverty or social exclusion (% of total population) in 2017. The chosen explanatory variables are *GDP* per inhabitant (at current market prices in Euro) in 2016, variable *Employment* — employment rates (in % of population aged 15–64) in 2016, variable *Education* — educational attainment level — less than primary, primary and lower secondary education (in % of population aged 25–64) in 2016, and variable *Density* — population density (inhabitants per square kilometre). The themeless map presented in Figure 2(a) and the box map presented in Figure 2(b) give an overview of regions included in the analysis. Interesting information can be found in the box map where one can notice no outliers in the first quartile, but up to fifteen outliers in the fourth quartile. The highest poverty rates are evident for the regions of Bulgaria, Romania and southern regions of Italy and Spain. In addition, upper outliers are not localized irregularly, but these regions are considerably clustered. A statistically significant value of global Moran's *I* statistic (0.6579) (*e.g.,* Getis [8]) for *Poverty* showed the existence of a strong positive spatial autocorrelation process.

[1] The SAR model is presented because this model is applied in an empirical part of the paper.



(a) (b)

Figure 2: *Themeless NUTS 2 EU regions map, 2(a), and box map (four categories) for People at risk of poverty or social exclusion rate in 2017, 2(b). Source: author's elaboration in ArcGISPro and GeoDa.*

Indicated significant spatial patterns will be taken into account in the following econometric analysis. In line with the "from general to specific" strategy, we begin with *Ordinary Least Square* (OLS) model estimation — OLS model, and next we proceed with spatial specification estimations — *Spatial Autoregressive* (SAR) model and *Spatial Error Model* (SEM). These spatial econometrics models are derived from the GNS model defined in (1) by imposing restrictions on one or more of its parameters. The estimations of the SAR model and the SEM model were based upon the specification (3) and (4), respectively. SAR specification:

$$\mathbf{y} = \rho\mathbf{W}\mathbf{y} + \mathbf{l}_N\alpha + \mathbf{X}\boldsymbol{\delta} + \mathbf{u}, \tag{3}$$

SEM specification:

$$\begin{aligned} &\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}, \\ &\mathbf{u} = \lambda\mathbf{W}\mathbf{u} + \mathbf{v}, \qquad \mathbf{v} \sim \mathcal{N}(\mathbf{0},\, \sigma_v^2\mathbf{I}_N), \end{aligned} \tag{4}$$

where $\mathbf{W}$ is a spatial weight matrix of the queen case contiguity form [9] and the remaining model terms are as defined before. The estimations were performed by the *Spatial Maximum Likelihood* (SML) estimator. The estimation results are given in Table 2.

Although OLS estimation provides statistically significant estimation of all parameters, statistical significance of the Moran's I applied on the OLS residuals, as well as the

Lagrange Multiplier test specification statistics and their robust versions (see Table 2) confirmed the presence of spatial dependencies. The Lagrange Multiplier test specification statistics suggested SEM model specification (4); however, we also decided to perform SAR specification estimation. The choice of SAR specifications has been supported by our assumption of the existence of global spillover effects in relation to modelling of regional poverty and social exclusion. Unlike the SEM model, the SAR model allows spillover effects at the global level (due to the presence of spatial lag of the dependent variable in the model) which means that changes in the i^{th} region activate a series of responses in others, potentially in all regions. SML estimations of models (3) and (4) produced statistically significant estimates of all parameters and almost of them had expected signs. Adequacy of spatial lag explicit incorporation into both the models was confirmed by the Likelihood Ratio test. Since the SAR model is a model with global spillover effects, one needs to be careful about verifying the statistical significance of the parameters and their interpretation. Statistical inference of the individual impacts associated with the changes in the explanatory variables should be based on the summary measures of impact. Tables 3–4 summarise the cumulative impacts (see Table 1) of all explanatory variables calculated on the basis of the SAR estimates. Testing the statistical significance of these cumulative impacts was based on a simulation approach [9].

	OLS model	SAR model	SEM model
Estimation	OLS	SML	SML
α	9.4836***	6.7351***	9.3717***
β_1(ln *GDP*)	-0.2252***	-0.1709***	-0.2177***
β_2(ln *Employment*)	-1.1211***	-0.8226***	-1.1375**
β_3(ln *Education*)	0.1239***	0.1000**	0.1146***
β_4(ln *Density*)	0.0360*	0.0412**	0.0625***
ρ	–	0.3267***	–
λ	–	–	0.5093***
R-squared	0.5669	–	–
Log likelihood	–	4.7432	10.3197
Moran's I (residuals)	5.8400***	–	–
Lagrange Multiplier (SAR)	15.3932***	–	–
Robust Lagrange Multiplier (SAR)	0.0559	–	–
Lagrange Multiplier (SEM)	24.6976***	–	–
Robust Lagrange Multiplier (SEM)	9.0400***	–	–
Likelihood Ratio test	–	15.2450***	26.3981***

Table 2: *Estimation results — OLS, SAR and SEM models. Symbols ***, **, and * indicate the rejection of the null hypotheses at 1%, 5%, and 10% level of significance, respectively.*

All cumulative impacts are statistically significant. Our assumption about spatial spillovers among regions can also be perceived as verified because of the statistical significance of all indirect impacts as well as the differences between parameter estimates and average indirect impacts (see Table 4). Next, take a closer look at the impacts associated with, *e.g.,* the variable *Education* — educational attainment level — less than primary, primary and lower secondary education. It was assumed that the higher the proportion of people with low levels of education, the higher the proportion of poor residents. This assumption was confirmed by positive signs of all impacts of educational attainment level (see Table 3). The average direct impact does not match the estimate of the parameter β_3

(ln*Education*). The difference between the average direct impact (0.1032) and the value of the parameter estimate (0.1000), is 0.0032, which is the amount of feedback that arises from the effects passing through the neighbouring regions, and is reversed by the region itself. Average total impact can be interpreted as elasticity, since the variables were expressed in logarithmic form. For example, based on average total educational attainment level impact, we can conclude that a 1% increase in the proportion of people with low levels of education will cause on average 0.1486% increase in the proportion of people at risk of poverty or social exclusion, while approximately 69% of this impact is attributed to direct impact, and 31% to indirect impact (see Table 4).

	Direct impact	Indirect impact	Total impact
ln*GDP*	-0.1763***	-0.0775***	-0.2538***
ln*Employment*	-0.8490***	-0.3728**	-1.2218***
ln*Education*	0.1032**	0.0453*	0.1486**
ln*Density*	0.0424**	0.0186*	0.0610**

Table 3: *Cumulative impacts of GDP, Employment, Education and Density.*

	ln*GDP*	ln*Employment*	ln*Education*	ln*Density*
Parameter estimate	-0.1709	-0.8226	0.1000	0.0412
Average direct impact *(ADI)*	-0.1763	-0.8490	0.1032	0.0424
Difference *ADI* and parameter estimate	-0.0054	-0.0264	0.0032	0.0012
Average indirect impact *(*AII)	-0.0775	-0.3728	0.0453	0.0186
Average total impact *(*ATI)	-0.2538	-1.2218	0.1486	0.0610
AII/ATI	0.3056	0.3051	0.3048	0.3049
ADI/ATI	0.6948	0.6949	0.6949	0.6951

Table 4: *Summary of direct, indirect and total impacts.*

4 Conclusion

The paper was aimed at the verification of the role of spatial spillovers in relation to regional poverty and social exclusion at the EU. In order to verify the hypothesis that there is a link not only between the chosen indicators and poverty within the region but that there are spillovers to neighbouring regions, two specifications of spatial econometric models were used. Based on the SAR model and calculated summary impacts, it can be concluded that neighbouring regions significantly participate in the share of total impact of each variable included in the model. Therefore, geographical location of the EU regions plays a significant role in explaining regional poverty and social exclusion. This conclusion was also confirmed by the results of model SEM. However, this model does not allow spatial spillovers at global level; it provides information only about direct effects in the form of parameter estimates. These effects are slightly different from direct effects calculated based on the SAR model, but both models indicated strong spatial interactions and significance of the chosen economic, social and demographic indicators. Constant monitoring of poverty at a regional level is needed in order to adequately allocate EU funds aimed at combating poverty and social exclusion and assess the effectiveness of their spending.

References

[1] ANSELIN L & REY SJ, 1993, *Modern spatial econometrics in practice: A guide to GeoDa, GeoDaSpace and PySAL*, GeoDa Press LLC, Chicago.

[2] CHOCHOLATÁ M, 2018, *Spatial analysis of the tertiary educational attainment in European Union*, Proceedings of the 15th International Conference, Efficiency and Responsibility in Education, Praha, pp. 117–124.

[3] EUROPEAN COMMISSION, 2010, *Europe 2020: A European strategy for smart, sustainable and inclusive growth*, [Online], [Cited May 5th, 2019], Available from `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2010:2020:FIN:EN:PDF`

[4] EUROPEAN COMMISSION, 2019, *New Cohesion Policy*, [Online], [Cited April 5th, 2019], Available from`https://ec.europa.eu/regional_policy/en/2021_2027/`

[5] EUROSTAT, 2019a, *Regional Eurostat statistics database*, [Online], [Cited April 5th, 2019], Available from `https://ec.europa.eu/eurostat/web/regions/data/database`

[6] EUROSTAT, 2019b, *Number of people at risk of poverty or social exclusion*, [Online], [Cited April 5th, 2019], Available from `https://ec.europa.eu/eurostat/statistics-explained/index.php/People_at_risk_of_poverty_or_social_exclusion`

[7] FURKOVÁ A, 2018, *Do R&D spillovers matter in the EU Regional Convergence Process? A spatial econometric approach*, Proceedings of the 34th International Scientific Conference on Economic and Social Development, XVIII International Social Congress, Moscow, pp. 24–33.

[8] GETIS A, 2010, *Spatial autocorrelation*, pp. 255–278 in FISCHER MM & GETIS A (EDS.), *Handbook of applied spatial analysis. Software tools, methods and applications*, Springer, Heidelberg.

[9] LESAGE J & PACE K, 2009, *Introduction to spatial econometrics*, Chapman and Hall/CRC, Boca Raton.

[10] LEVERNIER J, PATRIDGE MD & RICKMAN DS, 2000, *The causes of regional variations in US poverty: a cross-county analysis*, Journal of Regional Science, **40**, pp. 473–497.

[11] PANEK T & ZWIERZCHOWSKI J, 2014, *Comparative analysis of poverty in the EU member states and regions*, Warsaw school of Economics Press, Warsaw.

[12] RUPASINGHA A & GOETZ SJ, 2007, *Social and political forces as determinants of poverty: A spatial analysis*, The Journal of Socio-Economics, **36**, pp. 650–671.

Proceedings of the 48th ORSSA Annual Conference
pp. 88–95
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3
©2019

The simultaneous optimisation of price and loan-to-value

M Smuts[*] SE Terblanche[†]

Abstract

In a competitive financial industry, one of the challenges faced with secured retail lending products is to determine the optimal prices (*i.e.,* interest rates) that maximise both the loan take-up probability and expected revenue to the lender. We discuss a response model which relates take-up probabilities with price and loan-to-value (loan amount expressed as a percentage of the value of the underlying asset), taking into account a customer's willingness to pay for a loan. With the objective to maximise the expected *net present interest income* (NPII), a non-linear and piece-wise linear approximation approach was followed to simultaneously determine the optimal price and *loan-to-value* (LTV) for a potential customer while still adhering to the risk distribution constraints on the portfolio. By following a piece-wise linear approximation approach, logical decision making capability is introduced into the model, allowing for the exclusion of customers from the portfolio based on a trade-off between risk and profitability.

Key words: Simultaneous optimisation, price, loan-to-value, piece-wise linear approximation.

1 Introduction

Traditionally, cost based pricing was used to determine the prices of secured retail lending products. For these products, the costs included a risk premium based on the risk category of the customer. However, in recent years, pricing methodologies moved away from cost-based pricing towards demand-based pricing [3]. In demand-based pricing, the demand of a potential customer is mathematically captured by a price elasticity model (response model or logistic regression model) where the demand is expressed as a function of price. In secured retail lending products, the demand is referred to as the probability that the potential customer will take up a loan.

With demand-based pricing, the probability of take-up can be related to the change in price using a response model. In this paper, a response model is proposed that relates take-up probabilities with not only price, but also loan-to-value (loan amount expressed

[*] School of Mathematical and Statistical Sciences, North-West University, Private Bag X6001, Potchefstroom, 2520, Republic of South Africa, email: smuts.marius@nwu.ac.za

[†] School of Industrial Engineering , North-West University, Private Bag X6001, Potchefstroom, 2520, Republic of South Africa, email: fanie.terblanche@nwu.ac.za

as a percentage of the value of the underlying asset). This allows the lender to determine optimal levels of price and *loan-to-value* (LTV) through the application of an explicit optimisation model. The optimisation model considered in this paper is based on the work of Phillips [3] and Terblanche & De la Rey [4]. To the best of our knowledge, price and LTV optimisation had not yet been considered previously in secured retail lending products.

A piece-wise linear approximation approach is followed to simultaneously find the optimal price and LTV. In addition, the linear problem formulation allows for the introduction of logical decision variables, hence, allowing for the exclusion of customers from the portfolio based on a trade-off between risk and profitability.

2 A non-linear approach to price and LTV optimisation

Suppose a secured retail credit portfolio consists of customers $\mathcal{C} = \{1, 2, ..., C\}$. Consider Table 1 below, in which some of the parameters used in the credit retail price optimisation problem for each customer $c \in \mathcal{C}$ are shown.

Symbol	Description
a_c	loan amount approved
n_c	loan term
p_c	probability of default
r_c	repurchase rate
v_c	underlying asset value
l_c	LTV

Table 1: *Parameters used in the credit retail price optimisation problem.*

Let δ denote the loss given default and assume that $\delta = 1$. An approximation of the *net present interest income* (NPII) for a price x_c, $c \in \mathcal{C}$, is given by [3]

$$I(x_c) := I(x_c|n_c, a_c, r_c, p_c) = n_c a_c \left(\frac{x_c}{12} - \frac{r_c}{12}\right) - a_c p_c \delta. \tag{1}$$

An approximation of the probability of take-up (obtained from fitting a single logistic regression model to the data) for a price x_c, $c \in \mathcal{C}$, is given by the following response function [4].

$$R(x_c) := R(x_c|n_c, a_c, l_c, r_c, p_c) = 1/(1 + e^{(-(\beta_0+\beta_1 a_c+\beta_2 n_c+\beta_3 p_c+\beta_4 r_c+\beta_5 x_c+\beta_6 l_c))}), \tag{2}$$

with the regression coefficients $\beta_0, \beta_1, ..., \beta_6$ estimated using maximum likelihood. Figure 1 illustrates how take-up of a credit product is expected to decrease with an increase in price.

An approximation of the expected NPII is then given by the product of (1) and (2) and the unconstrained price optimisation problem is to

$$\begin{aligned} \text{maximise } & \textstyle\sum_{c\in\mathcal{C}} R(x_c)I(x_c) \\ \text{s.t. } & x_c \geq 0, \end{aligned}$$

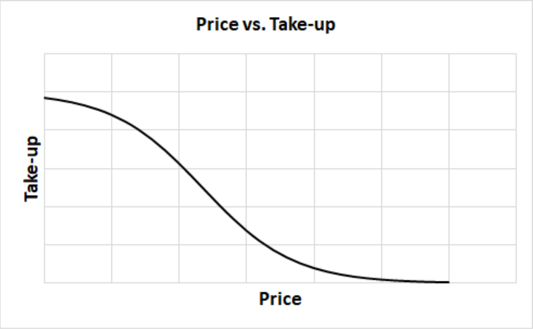


Figure 1: *Relationship between price and take-up probability.*

where x_c is the only decision variable (unknown variable) to be determined. A unique optimal solution for this price optimisation problem is guaranteed provided that $I(x_c)$ is a linear approximation, $R(x_c)$ has the increasing failure rate property [3] and the constraints are from a convex feasible region [1].

The objective of this study is, however, to maximise the expected NPII by finding the right balance between price and LTV according to the price elasticity model. For this purpose, the decision variable y_c, with $c \in \mathcal{C}$, is introduced to express LTV in terms of the loan amount (a) and the value of the underlying asset (v), that is

$$y_c = \frac{a_c}{v_c}. \tag{3}$$

The effect of LTV on take-up probability is illustrated by the graph in Figure 2. The take-up of a credit product is expected to increase with an increase in LTV, especially for first-time property owners who do not necessarily have enough funds available for the required deposit amount.

Writing equation (3) in terms of a_c and substituting the expression for a_c into equations (1) and (2), approximations of the NPII and the probability of take-up for a price x_c and LTV y_c are

$$I(x_c, y_c) := I(x_c, y_c | n_c, v_c, r_c, p_c) = n_c v_c y_c \left(\frac{x_c}{12} - \frac{r_c}{12}\right) - v_c y_c p_c \delta \tag{4}$$

and

$$R(x_{c,} y_c) := R(x_{c,} y_c | n_c, v_c, r_c, p_c) = 1/(1 + e^{(-(\beta_0 + \beta_1 v_c y_c + \beta_2 n_c + \beta_3 p_c + \beta_4 r_c + \beta_5 x_c + \beta_6 y_c))}), \tag{5}$$

respectively.

The expected NPII for a customer $c \in \mathcal{C}$ is then given by the product of (4) and (5). That is

$$f(x_c, y_c) := R(x_{c,} y_c) I(x_c, y_c).$$

Hence, the price and LTV optimisation problem, without any constraints, is to

$$\text{maximise } \textstyle\sum_{c \in \mathcal{C}} f(x_c, y_c)$$
$$\text{s.t. } x_c, y_c \geq 0.$$

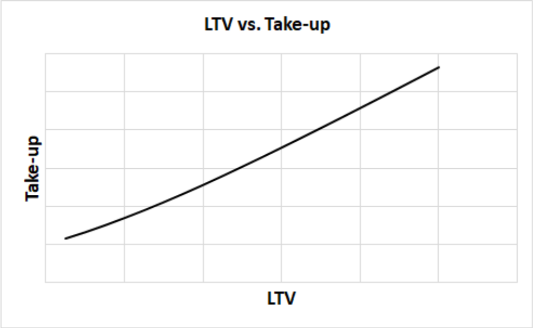


Figure 2: *Relationship between LTV and take-up probability.*

A risky portfolio can be regulated by incorporating constraints on the portfolio's risk distribution [3]. Let $\mathcal{C}(g)$ be the set of customers having a risk grading $g \in G = \{Low, Medium, High\}$. Furthermore, let U_g denote the upper bound for the proportion of customers having a risk grading of $g \in \mathcal{G}$, *i.e.,* for the unconstrained optimisation problem $U_g = \{1, 1, 1\}$. Let t_c be an auxiliary variable denoting the take-up probability of a customer $c \in \mathcal{C}$. The objective of the non-linear price and LTV optimisation model, which incorporates risk distribution constraints, is to

$$\begin{aligned} \text{maximise } & \textstyle\sum_{c\in\mathcal{C}} f(x_c, y_c) = R(x_{c,}y_c)I(x_c, y_c) & \\ \text{s.t. } & t_c = R(x_{c,}y_c), & \forall\, c \in \mathcal{C}, \\ & \textstyle\sum_{c\in\mathcal{C}(g)} t_c \leq U_g \sum_{c\in\mathcal{C}} t_c, & \forall\, g \in \mathcal{G}, \\ & x_c, y_c \geq 0, & \forall\, c \in \mathcal{C}. \end{aligned}$$

3 A linear approach to price and LTV optimisation

By using a linearisation approach to solve the price and LTV optimisation problem, proven optimal solutions are obtained and the option of introducing logical decision-making variables is made possible. More specifically, by introducing binary decision variables into the price and LTV problem formulation, it is possible to model the exclusion of certain customers from the credit portfolio, based on the required level of risk and the objective of maximising profitability.

The linear approximation approach entails dividing the price range into equally spaced intervals $\mathcal{I} = \{1, 2, ..., I\}$ for each $c \in \mathcal{C}$. Furthermore, let x_{ci} denote the price at the end point of the interval $i \in \mathcal{I}_0 = \mathcal{I} \cup 0 = \{0, 1, ..., I\}$, where $i = 0$ denotes the starting point of interval 1. Similarly, divide the LTV range into equally spaced intervals $\mathcal{J} = \{1, 2, ..., J\}$ for each $c \in \mathcal{C}$. Also, let y_{cj} denote the LTV at the end point of the interval $j \in \mathcal{J}_0 = \mathcal{J} \cup 0 = \{0, 1, ..., J\}$, where $j = 0$ denotes the starting point of interval 1. The expected NPII at the grid point (x_{ci}, y_{cj}) is given by

$$f_{cij} := f(x_{ci}, y_{cj}) = R(x_{ci,}y_{cj})I(x_{ci}, y_{cj}) = R_{cij}I_{cij}.$$

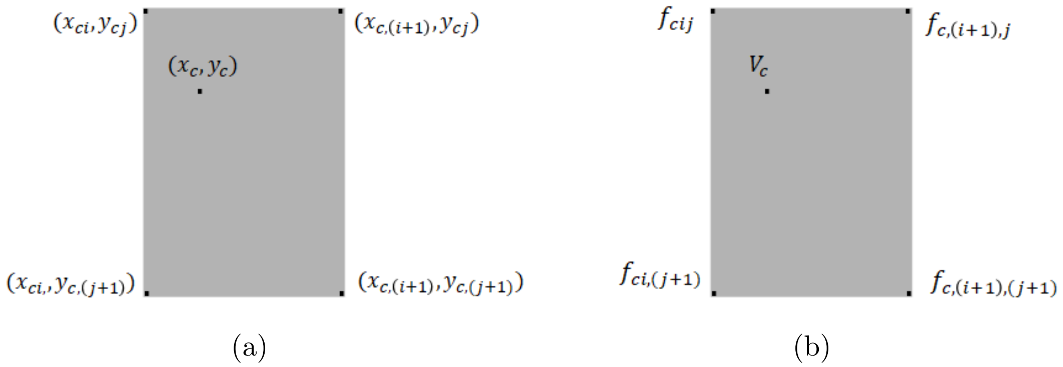
To interpolate between grid points, the price x_c and LTV y_c are expressed as convex combinations of the grid points (x_{ci}, y_{cj}), for all $i \in \mathcal{I}_0$ and $j \in \mathcal{J}_0$. For this purpose, the decision variable $\lambda_{cij} \in [0, 1]$ is introduced such that

$$x_c = \textstyle\sum_{i\in\mathcal{I}_0} \sum_{j\in\mathcal{J}_0} x_{ci}\lambda_{cij}, \tag{6}$$

$$y_c = \textstyle\sum_{i\in\mathcal{I}_0} \sum_{j\in\mathcal{J}_0} y_{cj}\lambda_{cij}, \tag{7}$$

$$V_c = \textstyle\sum_{i\in\mathcal{I}_0} \sum_{j\in\mathcal{J}_0} f_{cj}\lambda_{cij}, \tag{8}$$

with V_c the corresponding expected NPII. Note that, for an optimal solution to the linearisation problem, not all of the λ_{cij} variables may take on a value. More specifically, only the λ_{cij} variables corresponding to the vertices of the rectangle that contains the point (x_c, y_c) may be allowed to take on a value [2]. The illustration in Figure 3(a) shows the grid points that form the vertices of the rectangle that contains the point (x_c, y_c). The corresponding function values that form a rectangle containing the expected NPII value, V_c, are shown in Figure 3(b).



(a) (b)

Figure 3: *(a) Convex combination of grid points; (b) Function approximation.*

The activation of the appropriate rectangles is facilitated by the binary decision variable $s_{cij} \in \{0, 1\}$, $i \in \mathcal{I}$ and $j \in \mathcal{J}$. For the illustration in Figure 3, by letting $s_{c,(i+1),(j+1)} = 1$, the optimal price, LTV and expected NPII, are expressed as the following convex combinations:

$$x_c = x_{ci}\lambda_{cij} + x_{ci}\lambda_{ci,(j+1)} + x_{c,(i+1)}\lambda_{c,(i+1),j} + x_{c,(i+1)}\lambda_{c,(i+1),(j+1)}, \tag{9}$$

$$y_c = y_{cj}\lambda_{cij} + y_{c,(j+1)}\lambda_{ci,(j+1)} + y_{c,j}\lambda_{c,(i+1),j} + y_{c,(j+1)}\lambda_{c,(i+1),(j+1)} \tag{10}$$

and

$$V_c = f_{cij}\lambda_{cij} + f_{ci,(j+1)}\lambda_{ci,(j+1)} + f_{c,(i+1),j}\lambda_{c,(i+1),j} + f_{c,(i+1),(j+1)}\lambda_{c,(i+1),(j+1)}, \tag{11}$$

respectively. The final selection of customers to be included in the credit portfolio, is facilitated by introducing the binary decision variable $z_c \in \{0, 1\}$, for each customer $c \in \mathcal{C}$. If $z_c = 1$, the customer will be made a loan offer at an interest rate of x_c and an LTV of

y_c. The objective of the linear price and LTV optimisation problem with risk distribution constraints is to

$$
\begin{aligned}
\text{maximise } & \textstyle\sum_{c\in\mathcal{C}} V_c && && (12)\\
\text{s.t. } & V_c \leq \textstyle\sum_{i\in\mathcal{I}_0}\sum_{j\in\mathcal{J}_0}\lambda_{cij}f_{cij} + (1-z_c)M, && \forall\, c\in\mathcal{C}, && (13)\\
& V_c \geq \textstyle\sum_{i\in\mathcal{I}_0}\sum_{j\in\mathcal{J}_0}\lambda_{cij}f_{cij} - (1-z_c)M, && \forall\, c\in\mathcal{C}, && (14)\\
& V_c \leq\ Mz_c, && \forall\, c\in\mathcal{C}, && (15)\\
& V_c \geq\ -Mz_c, && \forall\, c\in\mathcal{C}, && (16)\\
& x_c = \textstyle\sum_{i\in\mathcal{I}_0}\sum_{j\in\mathcal{J}_0} x_{ci}\lambda_{cij}, && \forall\, c\in\mathcal{C}, && (17)\\
& y_c = \textstyle\sum_{i\in\mathcal{I}_0}\sum_{j\in\mathcal{J}_0} y_{cj}\lambda_{cij}, && \forall\, c\in\mathcal{C}, && (18)\\
& t_c \leq \textstyle\sum_{i\in\mathcal{I}_0}\sum_{j\in\mathcal{J}_0}\lambda_{cij}R_{cij} + (1-z_c)M, && \forall\, c\in\mathcal{C}, && (19)\\
& t_c \geq \textstyle\sum_{i\in\mathcal{I}_0}\sum_{j\in\mathcal{J}_0}\lambda_{cij}R_{cij} - (1-z_c)M, && \forall\, c\in\mathcal{C}, && (20)\\
& t_c \leq\ Mz_c, && \forall\, c\in\mathcal{C}, && (21)\\
& t_c \geq\ -Mz_c, && \forall\, c\in\mathcal{C}, && (22)\\
& \textstyle\sum_{j\in\mathcal{J}_0}\lambda_{c0j} \leq \sum_{j\in\mathcal{J}} s_{c1j}, && \forall\, c\in\mathcal{C}, && (23)\\
& \textstyle\sum_{j\in\mathcal{J}_0}\lambda_{cij} \leq \sum_{j\in\mathcal{J}}(s_{cij} + s_{c,(i+1),j}), && \forall\, c\in\mathcal{C}, \forall i\in\mathcal{I}/\{I\}, && (24)\\
& \textstyle\sum_{j\in\mathcal{J}_0}\lambda_{cIj} \leq \sum_{j\in\mathcal{J}} s_{cIj}, && \forall\, c\in\mathcal{C}, && (25)\\
& \textstyle\sum_{i\in\mathcal{I}_0}\lambda_{ci0} \leq \sum_{i\in\mathcal{I}} s_{ci1}, && \forall\, c\in\mathcal{C}, && (26)\\
& \textstyle\sum_{i\in\mathcal{I}_0}\lambda_{cij} \leq \sum_{i\in\mathcal{I}}(s_{cij} + s_{ci,(j+1)}), && \forall\, c\in\mathcal{C}, \forall j\in\mathcal{J}/\{J\}, && (27)\\
& \textstyle\sum_{i\in\mathcal{I}_0}\lambda_{ciJ} \leq \sum_{i\in\mathcal{I}} s_{ciJ}, && \forall\, c\in\mathcal{C}, && (28)\\
& \textstyle\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} s_{cij} = 1, && \forall\, c\in\mathcal{C}, && (29)\\
& \textstyle\sum_{i\in\mathcal{I}_0}\sum_{j\in\mathcal{J}_0}\lambda_{cij} = 1, && \forall\, c\in\mathcal{C}, && (30)\\
& \textstyle\sum_{c\in\mathcal{C}(g)} t_c \leq U_g \sum_{c\in\mathcal{C}} t_c, && \forall\, g\in\mathcal{G}. && (31)
\end{aligned}
$$

In the optimisation model above, constraints (13)–(16) incorporate the logical decision making capability by allowing the exclusion of a customer from the portfolio through the use of the binary decision variable z_c. When $z_c = 1$, constraints (13) and (14) bind V_c to a convex combination of points, similar to equation (11) and therefore including the customer in the portfolio. However, when $z_c = 0$, constraints (15) and (16) bind V_c to 0 and therefore excluding the customer from the portfolio. In order to improve numerical stability, the constant M is assigned the value $\max(f_{cij})$. Constraints (17) and (18) are included to obtain the values of the auxiliary variables x_c and y_c, similar to that of equation (9) and (10). Constraints (19)–(22) ensure that the auxiliary variable for the take-up, t_c, either takes on the value of 0 when the customer is excluded from the portfolio, or the value of a convex combination of points when included in the portfolio. Constraints (23)–(30) ensure that only a convex combination of weights associated with the single rectangle is activated per customer $c \in \mathcal{C}$ and that these weights add up to 1. Constraint (31) is included to regulate the risk in the portfolio by setting upper bounds on the risk distribution.

4 Model behavior and computational results

In order to validate the proposed price and LTV optimisation model, an empirical study was performed on a real-world data set from a financial institution. Due to the sensitive nature of the data, no details on the data or the institution itself could be made available. The results reported below are also normalised in order to disguise price and LTV characteristics.

The results displayed in the tables below are summarised per risk grading $g \in G = \{Low, Medium, High\}$. For the average optimal price per risk grading, the values are expressed as a percentage of the risk grading(s) with highest average optimal price. The following tables summarise the model behavior for the unconstrained ($U_g = \{1, 1, 1\}$) price and LTV optimisation problem for both of the optimisation models implemented.

Risk grading	Take-up proportion	Average price	Average LTV
Low	0.28	1.00	1.00
Medium	0.32	0.98	1.00
High	0.40	0.94	1.00

Table 2: *Computational results (unconstrained): The non-linear approach.*

Risk grading	Take-up proportion	Average price	Average LTV
Low	0.27	1.00	1.00
Medium	0.33	0.97	1.00
High	0.40	0.93	1.00

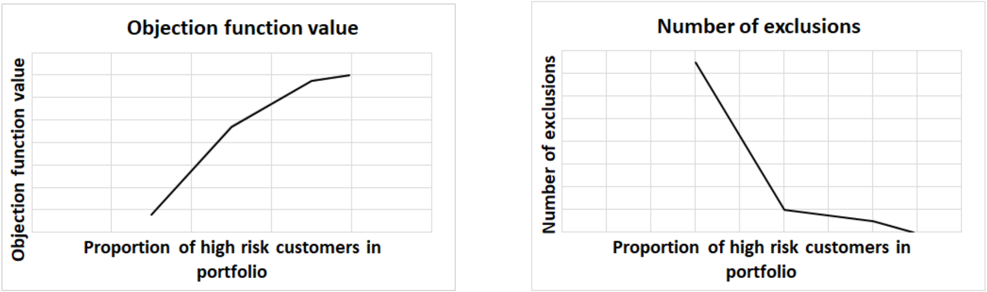
Table 3: *Computational results (unconstrained): The piece-wise linear approach.*

From the results in Tables 2 and 3, it can be seen that the average optimal price and LTV per risk grading are similar over the portfolio and also similar for the two different approaches followed. These results serve as a validation of the linearisation approach.

In order to illustrate the behavior of the price and LTV optimisation model when high-risk customers are excluded from the final portfolio, the objective function value and the corresponding number of customers excluded are reported below for a range of high-risk grading constraint values. That is, the proportion of high-risk customers in the portfolio were restricted to 0.05, 0.15, 0.25 and 0.35. The graphs below illustrate the effect of the these constraints with respect to objective function values and the number of high-risk customers excluded.

From Figure 4 (a) and (b) it can be seen that as the proportion of high-risk customers in the portfolio decreases, the objective function value decreases and the number of customers excluded increases.

The results in Table 4 demonstrate what the impact is on the price and LTV when the constraints $U_g = \{1, 0.3, 0.2\}$ are imposed on the risk distribution.



(a) (b)

Figure 4: *(a) Impact of logical decision making capability on objective function value; (b) Impact of logical decision making capability on number of exclusions.*

Risk Category	Take-up proportion	Average price	Average LTV
Low	0.5	0.63	1.00
Medium	0.3	0.88	1.00
High	0.2	1.00	0.92

Table 4: *Computational results (constrained): The piece-wise linear approach.*

From the results in Table 4, it is clear that loans to be offered to high-risk customers will have a higher average price and a lower average LTV compared to the lower risk customers. The expectation is that higher risk customers will be discouraged from taking up a loan if the price is too high and the LTV is too low. Conversely, if lower risk customers are offered lower rates and higher LTV values, it may result in an improved take-up by low-risk customers.

5 Conclusion

Two different approaches were used in the simultaneous optimisation of price and *loan-to-value* (LTV) when maximising the expected *net present interest income* (NPII) to the lender. Although both approaches yielded similar results for the unconstrained case, the piece-wise linear approximation approach always provides proven optimal solutions and, in addition, it allows for the inclusion of binary decision variables which facilitate logical decision-making on a portfolio level.

References

[1] Boyd S & Vandenberghe L, 2004, *Convex optimization*, Cambridge university press, Cambridge.

[2] Misener R & Floudas CA, 2010, *Piecewise-linear approximations of multidimensional functions*, Journal of optimization theory and applications, **145(1)**, pp. 120–147.

[3] Phillips R, 2013, *Optimizing prices for consumer credit*, Journal of Revenue and Pricing Management, **12(4)**, pp. 360–377.

[4] Terblanche SE & De la Rey T, 2014, *Credit price optimisation within retail banking*, ORiON, **30(2)**, pp. 85–102.

Proceedings of the 48th ORSSA Annual Conference
pp. 96–105
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3
©2019

A single-objective *versus* a bi-objective optimisation approach towards the training of artificial neural networks

GS Nel[*] JH van Vuuren[†]

Abstract

Artificial neural networks (ANNs) have proven to be adept at performing various tasks within the field of *machine learning* (ML) — delivering state of the art performance in respect of numerous classification and regression problems. Arguably, the most salient aspect of ANNs pertains to the process of *training* the network weights. The employment of gradient-based optimisation algorithms, and more specifically first-order techniques such as gradient descent, is regarded as the standard approach towards finding good network weights. The efficacy of a gradient-based approach is, however, impeded by its tendency to converge to poor local optima. Gradient-free optimisation algorithms, on the other hand, provide an alternative approach that better mitigates the perils of traversing highly non-linear search spaces — a common predicament associated with ANN training. The realm of *evolutionary optimisation* comprises many powerful and robust gradient-free approaches and the domain of *multi-objective optimisation* (MOO) provides further versatility over a standard *single-objective optimisation* (SOO) approach. The problem considered in this paper is to investigate the potential performance benefits of an MOO approach over an SOO approach in the context of training ANNs using the celebrated *genetic algorithm* — arguably the best-known evolutionary algorithm. The *multi-objective evolutionary algorithm* that forms the basis of the work presented in this paper is the popular *non-dominated sorting genetic algorithm II* (NSGA-II). The results indicate that the adoption of an MOO approach results in a noteworthy improvement in performance over an SOO approach with respect to the quality of ANNs obtained.

Key words: Artificial neural networks, training algorithms, evolutionary optimisation.

1 Introduction

In 1959, Arthur Samuel defined *machine learning* (ML) as the "field of study that gives computers the ability to learn without being explicitly programmed." Since its inception, ML has developed into a field of research that is leading the scientific endeavour of

[*] Stellenbosch Unit for Operations Research and Engineering, Department of Industrial Engineering, University of Stellenbosch, Private Bag X1, Matieland, 7602, Republic of South Africa, email: gsnel@sun.ac.za

[†] **(Fellow of the Operations Research Society of South Africa)**, Stellenbosch Unit for Operations Research and Engineering, Department of Industrial Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602, Republic of South Africa, email: vuuren@sun.ac.za

achieving *artificial intelligence* (AI) [8]. According to Russel & Norvig [8], a machine that exhibits intelligence can communicate, memorise, make informed decisions, and adapt dynamically — as corroborated by the famous *Turing Test* designed by mathematician Alan Turing. These desirable traits serve as sufficient justification why ML is still an active research field experiencing marked levels of innovation.

Just as ML is at the forefront of AI, *artificial neural networks* (ANNs) are at the forefront of ML. ANNs are computational models inspired by the neurological design of *biological neural networks* (BNNs) [6]. The emulation of the information processing capability associated with BNNs enable ANNs to inherit a key ability — to learn from experience. Within the given context, "experience" is represented by data, within which complex patterns are embedded that can be "learnt" by ANNs. Consequently, challenging tasks such as *natural language processing* and *computer vision* have become far less arduous to perform by computers as a result of the processing capabilities afforded to ANNs [2].

The complex patterns found within data sets are captured by the adjustable parameters of ANNs — *i.e.,* the *network weights.* The process of learning these weights transpire during the so-called *training* stage. Given a data set, the weights of an ANN are algorithmically adjusted until the network is an appropriate functional representation of the data set. A training algorithm governs this process. The standard approach employs first-order optimisation algorithms (in conjunction with *backpropagation*), but this approach is inhibited by its tendency to converge to poor local optima [1]. Derivative-free approaches offer an (admittedly computationally expensive) alternative that mitigates the pitfalls associated with highly non-linear search spaces — a property synonymous with ANN training.

The prominent field of *evolutionary optimisation* comprises powerful solution methodologies that can be employed in the context of training ANNs [10]. Some of the most powerful *evolutionary algorithms* (EAs) include the *genetic algorithm* (GA) [7], *differential evolution* [9], and *particle swarm optimisation* [5]. Furthermore, the domain of *multi-objective optimisation* (MOO) has proven useful in delivering high-quality solutions to complex problems in many fields of scientific study, such as engineering, economics, and logistics [10]. To the best of the authors' knowledge, an investigation into the performance benefits related to the employment of the celebrated GA towards training ANNs in an MOO context *versus* in an *single-objective optimisation* (SOO) context has not yet been conducted. More specifically, a bi-objective optimisation approach towards the training of ANNs, in which the main objective function represents the network performance measure and the secondary objective function represents a regularisation technique to guide the search process, warrants investigation. A powerful and robust MOO version of the GA forms the basis of the work presented in this paper — the *non-dominated sorting genetic algorithm II* (NSGA-II) [3].

This paper is organised as follows. Section 2 contains a detailed description of *feed-forward neural networks* (FNNs) — a sufficiently general representation of ANNs — and its most salient parts. In §3, the working of the GA and that of its MOO counterpart, the NSGA-II, are described in full. Results pertaining to the performance comparison between the SOO and MOO approaches are presented in §4. The conclusions of the paper are finally presented in §5.

2 Feed-forward neural networks

Feed-forward neural networks (FNNs) are one of the most prominent types of ANNs and are therefore selected to form the basis of discourse in this paper. Other prominent types of ANNs include *recurrent neural networks* and *convolutional neural networks*, but FNNs are sufficiently representative of the general operation of ANNs — the process of training remains, by-and-large, similar amongst the different ANN types [6]. FNNs are characterised by the fact that data flow strictly in a forward direction, *i.e.,* no feedback connections are present. This distinguishing trait is illustrated graphically in Figure 1, in which a *multi-layer feed-forward neural network* (MFNN) comprising a single hidden layer is presented. As can be seen, this MFNN comprises an input layer and an output layer, with a single hidden layer connecting these two layers. According to Bishop [2], for most supervised learning problems a single hidden layer has the same processing capability as two hidden layers. This MFNN is assumed as the basis of the discussion in the remainder of the section, which aims to delineate the inner workings of an FNN as well as define the accompanying notation.

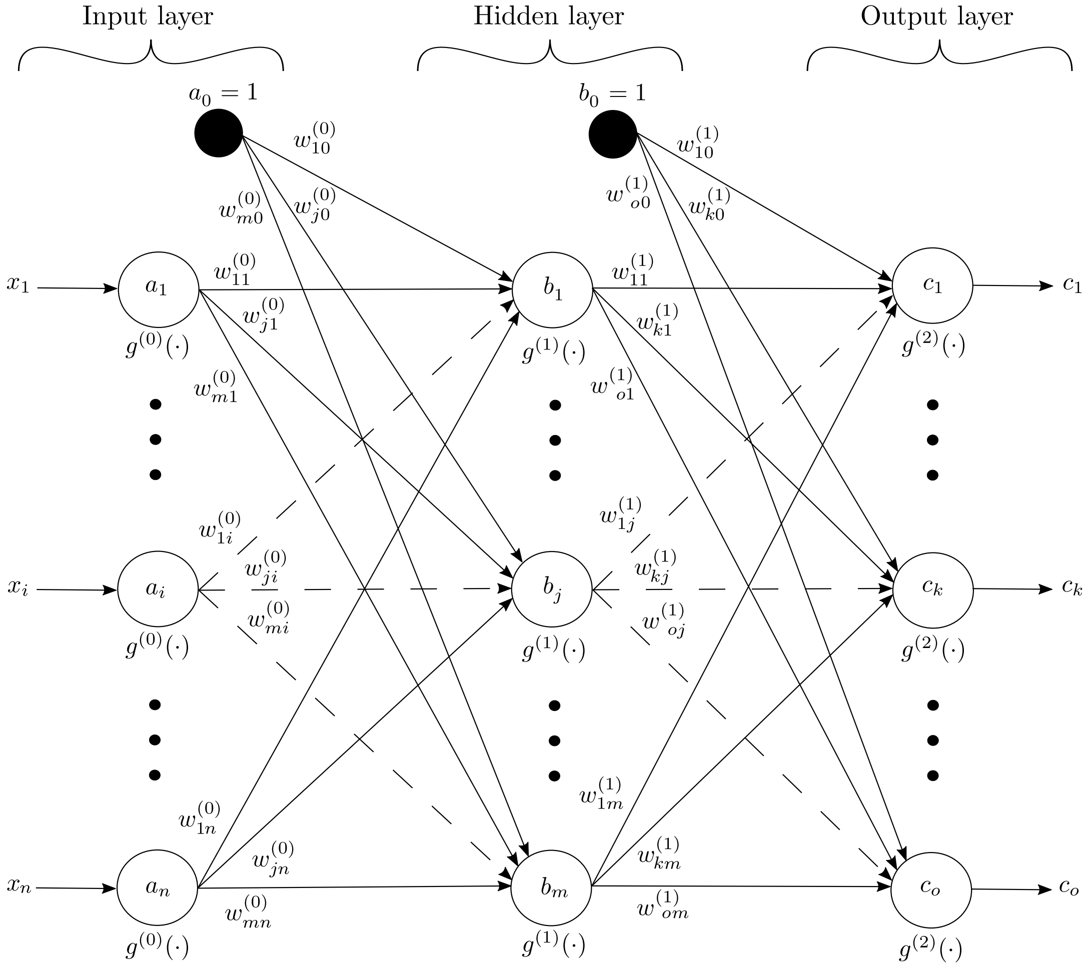


Figure 1: *An MFNN comprising one hidden layer.*

2.1 Analytical expression of an MFNN

The MFNN illustrated in Figure 1 comprises n input neurons, o output neurons, and m hidden neurons. Furthermore, an input vector (or training example) $\boldsymbol{x} = [x_1 \cdots x_i \cdots x_n]$ is presented to the network. The main components of an MFNN are as follows:

- Input layer activations, denoted by $\boldsymbol{a} = [a_1 \cdots a_i \cdots a_n]^T$ (excluding bias) or $\tilde{\boldsymbol{a}} = [a_0\ a_1 \cdots a_i \cdots a_n]^T$ (including bias) where $a_0 = 1$;
- hidden layer activations, denoted by $\boldsymbol{b} = [b_1 \cdots b_j \cdots b_m]^T$ (excluding bias) or $\tilde{\boldsymbol{b}} = [b_0\ b_1 \cdots b_j \cdots b_m]^T$ (including bias) where $b_0 = 1$;
- output layer activations, denoted by $\boldsymbol{c} = [c_1 \cdots c_k \cdots c_o]^T$;
- weights corresponding to the connection between input neuron $i \in \{0, \ldots, n\}$ and hidden neuron $j \in \{1, \ldots, m\}$, denoted by $w_{ji}^{(0)}$, and contained within a weight-matrix $\boldsymbol{W}^{(0)}$;
- weights corresponding to the connection between hidden neuron $j \in \{0, \ldots, m\}$ and output neuron $k \in \{1, \ldots, o\}$, denoted by $w_{kj}^{(1)}$, and contained within a weight-matrix $\boldsymbol{W}^{(1)}$ with the entire network's weights being contained within an ordered list $\boldsymbol{w} = \{\boldsymbol{W}^{(0)}, \boldsymbol{W}^{(1)}\}$;
- input layer activation functions, denoted by $g^{(0)}(\cdot)$;
- hidden layer activation functions, denoted by $g^{(1)}(\cdot)$;
- output layer activation functions, denoted by $g^{(2)}(\cdot)$.

A mathematical function that expresses the working of an MFNN analytically is now derived[1]. The input layer employs an identity (activation) function. The activation of each input neuron therefore equates to the input itself, *i.e.,* $a_i = x_i$, which can be ascribed to the fact that $x_i = g_i^{(0)}(x_i)$ for $i \in \{1, \ldots, n\}$. The net input to hidden neuron j, denoted by $\eta_j^{(1)}$ for $j \in \{1, \ldots, m\}$, can be calculated by using these input layer activations. Correspondingly, the net input is given by

$$\eta_j^{(1)} = \sum_{i=1}^{n} w_{ji}^{(0)} a_i + w_{j0}^{(0)} = \sum_{i=0}^{n} w_{ji}^{(0)} a_i, \tag{1}$$

which is used to calculate the hidden neuron activations, denoted by b_j. By employing the corresponding activation function $g^{(1)}(\cdot)$, the respective hidden neuron activations can be expressed as

$$b_j = g^{(1)}\left(\eta_j^{(1)}\right). \tag{2}$$

An alternative derivation relates to the expression of the hidden layer activations $\boldsymbol{b}$ as the product of the weights between the input layer and the hidden layer, denoted by $\boldsymbol{W}^{(0)}$, and the input layer activations $\tilde{\boldsymbol{a}}$. This mathematical operation can be expressed succinctly as

$$\boldsymbol{b} = \boldsymbol{g}^{(1)}\left(\boldsymbol{W}^{(0)}\tilde{\boldsymbol{a}}\right),$$

where $\boldsymbol{g}^{(1)}$ represents the vector of activation functions (of length m). Finally, the calculation of the net input to output neuron k, denoted by $\eta_k^{(2)}$ for $k \in \{1, \ldots, o\}$, is similar,

[1] The derivation is based on the work of Bishop [2].

yielding

$$\eta_k^{(2)} = \sum_{j=1}^{m} w_{kj}^{(1)} b_j + w_{k0}^{(1)} = \sum_{j=0}^{m} w_{kj}^{(1)} b_j. \tag{3}$$

The corresponding activation function $g^{(2)}(\cdot)$ is employed in order to calculate the activation of an output neuron, denoted by c_k. This calculation yields

$$c_k = g^{(2)}\left(\eta_k^{(2)}\right). \tag{4}$$

The mathematical function that analytically represents an MFNN is obtained by combining (1)–(4), which consequently yields

$$c_k = g^{(2)}\left(\sum_{j=0}^{m} w_{kj}^{(1)} g^{(1)}\left(\sum_{i=0}^{n} w_{ji}^{(0)} a_i\right)\right), \tag{5}$$

or, expressed more succinctly,

$$\boldsymbol{c} = \boldsymbol{g}^{(2)}\left(\boldsymbol{W}^{(1)}\tilde{\boldsymbol{b}}\right). \tag{6}$$

Given the derivation above, an MFNN may be expressed as a non-linear function of the adjustable network weights. The degree of this function's non-linearity can further be "enhanced" by employing non-linear activation functions in the hidden layer, such as the s-shaped *sigmoid* function, thus affording greater computational capability to the network — *i.e.,* more complex patterns and abstractions within the data set can be *learnt*. Network learning represents the process of algorithmically adjusting the network weights — a matter elucidated shortly.

2.2 Mathematical representation of the process of supervised learning

The field of ML comprises five main learning paradigms, namely *supervised learning*, *unsupervised learning*, *semi-supervised learning*, *reinforcement learning*, and *self-supervised learning*. Supervised learning forms the basis of the work presented in this paper. In essence, *labelled* data are presented to the network, which comprise *input features* and *target variables* — akin to independent and dependent variables, respectively — after which the aim is to approximate the underlying functional representation of the input-to-output mapping [6]. Supervised learning problems can be categorised into *classification* problems and *regression* problems. The work presented in this paper, however, focusses specifically on classification problems — a decision ascribed to its greater popularity. Classification problems essentially deal with the assignment of a vector of input features to one of a finite number of discrete categories. A mathematical representation of supervised learning now follows.

The MFNN (illustrated in Figure 1) once again forms the basis of the elucidation. Recall that the ordered list $\boldsymbol{w}$ comprises the weight-matrices $\boldsymbol{W}^{(0)}$ and $\boldsymbol{W}^{(1)}$. The network is usually first presented with a set of inputs, comprising Q input vectors, which is denoted here by $\mathcal{X} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^q, \ldots, \boldsymbol{x}^Q\}$. The multivariate function f that represents the MFNN is given by

$$\boldsymbol{c}^q = f(\boldsymbol{x}^q; \boldsymbol{w}), \qquad q \in \{1, \ldots, Q\}, \tag{7}$$

where $\boldsymbol{c}^q$ represents the network output with respect to input vector $\boldsymbol{x}^q$. The analytical representation in (7) is a more general form of that in (6). In a typical training process, an input vector $\boldsymbol{x}^q$ together with its corresponding target vector $\boldsymbol{y}^q$ are known *a priori*, and the weights in $\boldsymbol{w}$ are methodically adjusted so as to approximate an appropriate functional mapping from the input $\boldsymbol{x}^q$ to the desired output $\boldsymbol{y}^q$. The set of target vectors is denoted by $\mathcal{Y} = \left\{\boldsymbol{y}^1, \ldots, \boldsymbol{y}^q, \ldots, \boldsymbol{y}^Q\right\}$. The network output $\boldsymbol{c}$ may therefore be regarded as a prediction $\hat{\boldsymbol{y}}^q$ of $\boldsymbol{y}^q$, *i.e.,* $\hat{\boldsymbol{y}}^q \approx \boldsymbol{c}^q$. The corresponding functional mapping is thus

$$f : \mathcal{X} \mapsto \mathcal{Y}.$$

The training algorithm is tasked with adjusting the network weights in $\boldsymbol{w}$ in order to learn the functional relationship between the input and output. The network is iteratively presented with input-output pairs, *i.e.,* different training examples $q \in \{1, \ldots, Q\}$, until the network's adjustable parameters sufficiently capture the relationship.

Consequently, the training of FNNs may be regarded as an optimisation problem over a (typically) high-dimensional parameter space. The objective function is represented by the network's *performance measure*, often referred to as the *network error* or *network accuracy* (to be minimised or maximised, respectively). The function representing the performance measure has the form

$$Z(\boldsymbol{y}^q, f(\boldsymbol{x}^q; \boldsymbol{w})). \tag{8}$$

The objective function corresponding to most FNN training problems is typically highly non-linear and non-convex. Consequently, an abundance of local minima and maxima (of varying depths or heights) are present [1, 2]. As a result, gradient-based optimisation approaches tend to converge to poor local optima. Fortunately, evolutionary optimisation offers powerful search strategies that better mitigate the pitfalls of traversing highly non-linear search spaces.

2.3 Regularisation

A key challenge faced by ANN researchers and practitioners pertains to *overfitting* — the network performs well with respect to the training set (*i.e.,* good memorisation), but it performs poorly with respect to the independent testing set (*i.e.,* poor generalisation). According to Goodfellow *et al.* [6], *regularisation* is "any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error." This section contains a brief description of one of the most prominent regularisation strategies found in the literature, namely L^2 regularisation. This technique is employed as the secondary regularising objective function, which guides the search process by mitigating overfitting.

L^2 parameter regularisation, often referred to as *ridge regression* or *Tikhonov regularisation*, aims to *drive* the weights closer to the origin[2]. According to this strategy, the regularisation term

$$\frac{\lambda}{2Q} \sum_{w} w^2 \tag{9}$$

[2] Zero is chosen as the default value to regularise towards as it is unclear, beforehand, whether the *best* value to regularise towards is positive or negative. According to Goodfellow *et al.* [6], zero is by far the most common regularisation *target*.

is added to the network error and equates to the squared sum of all weight values scaled by $\lambda/2Q$, with Q being the size of the data set and $\lambda > 0$ the *regularisation parameter*[3]. The regularisation term aims to bias the training algorithm towards preferring and learning small weights. The fundamental intuition is that, collectively, *all* the neurons within the network have a greater capacity to learn the *true* underlying relationships embedded within the data. Networks comprising few, large weights are therefore avoided.

2.4 Model formulation

The purpose of this section is to formulate an appropriate mathematical model of the problem at hand — *i.e.,* training FNN weights in order to facilitate its effective solution of various classification problems. Suppose a mini-batch of random training examples is evaluated so as to estimate the network's performance. This subset, denoted by $(\hat{\mathcal{X}}, \hat{\mathcal{Y}}) \subset (\mathcal{X}, \mathcal{Y})$, is given by $\{(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{y}}^1), \ldots, (\hat{\boldsymbol{x}}^p, \hat{\boldsymbol{y}}^p), \ldots, (\hat{\boldsymbol{x}}^P, \hat{\boldsymbol{y}}^P)\}$, where $\hat{\boldsymbol{x}}^p = [\hat{x}_1^p \cdots \hat{x}_i^p \cdots \hat{x}_n^p]$ and $\hat{\boldsymbol{y}}^p = [\hat{y}_1^p \cdots \hat{y}_k^p \cdots \hat{y}_o^p]$. Whenever a candidate solution is evaluated, a random mini-batch of examples is used to evaluate the network's performance. Given the aforementioned derivations, the main objective function is to

$$\left.\begin{array}{lll} \text{minimise} & h_1\left(\hat{\mathcal{X}}, \hat{\mathcal{Y}}; \boldsymbol{w}\right) = \dfrac{1}{P}\displaystyle\sum_{p=1}^{P}\sum_{k=1}^{o}(\hat{y}_k^p - c_k^p)^2, & \\ \text{subject to} & w_{ji}^{(0)} \in \mathbb{R}, & j \in \{1, \ldots, m\},\ i \in \{0, \ldots, n\}, \\ & w_{kj}^{(1)} \in \mathbb{R}, & k \in \{1, \ldots, o\},\ j \in \{0, \ldots, m\}, \end{array}\right\} \tag{10}$$

in which the network performance measure is represented by the *mean squared error* (MSE) in respect of a random mini-batch. The MSE provides a relatively (computationally) inexpensive measure of how good the network prediction c_k^p is in respect of the corresponding target value y_k^p for $k \in \{1, \ldots, o\}$. The secondary regularising objective function h_2 is simply given by the expression in (9). The incorporation of this objective function aims to guide the search process by preferring networks that are less inclined to overfit.

3 The NSGA-II

One of the best-founded and celebrated EAs is the GA, which was originally proposed by Holland [7]. Charles Darwin's biological theory of evolution by natural selection serves as the main source of inspiration for this powerful *metaheuristic*[4]. In essence, this theory states the following: Evolution within a species is represented by the survival — by means of competition (*i.e.,* selecting those that adapt best) — of the individuals deemed most "fit". The flow diagram in Figure 2 illustrates the fundamental working of a generic GA, which may be applied in the contexts of both SOO and MOO. One of the most prominent MOO versions is the NSGA-II, proposed by Deb *et al.* [3]. Some of the main improvements over its predecessor relate to the reduction in computational complexity (attributable to its non-dominated sorting algorithm) and the incorporation of its crowded comparison operator which allows for the exploration of more diverse solutions. After its inception

[3] Goodfellow *et al.* [6] report that a value of $\lambda = 0.1$ generally results in satisfactory performance.

[4] A search procedure for finding — within a reasonable computation time — high-quality solutions to computationally hard optimisation problems.

in 2002, the NSGA-II quickly became an influential EA, as stated by Talbi [10]. To the best of the authors' knowledge, the application of the NSGA-II to the bi-objective model described in §2.4 has not yet been attempted in the literature. The performance improvements that can potentially be gained by following the proposed MOO approach, instead of a standard SOO approach, warrant further investigation. The SOO approach simply represents the case in which there is only the one objective function — as expressed in (10) — to which a standard GA is applied.

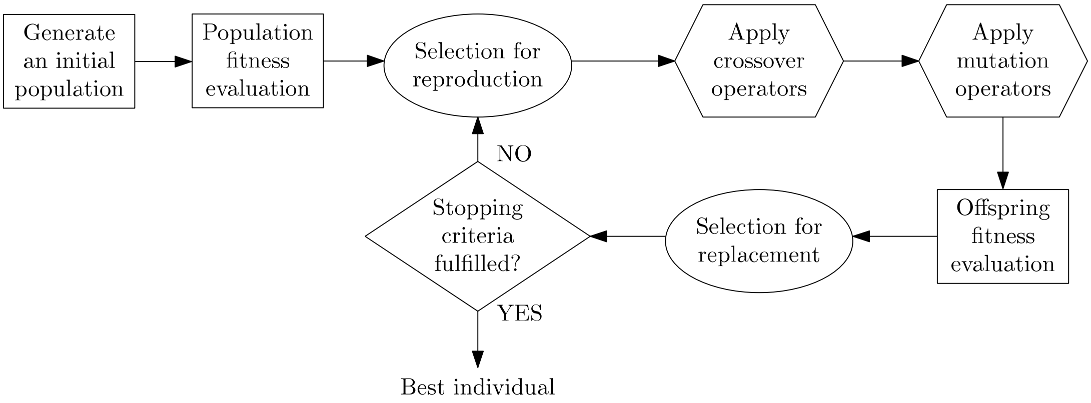

Generate an initial population
Population fitness evaluation
Selection for reproduction
Apply crossover operators
Apply mutation operators
Offspring fitness evaluation
Selection for replacement
Stopping criteria fulfilled?
NO
YES
Best individual

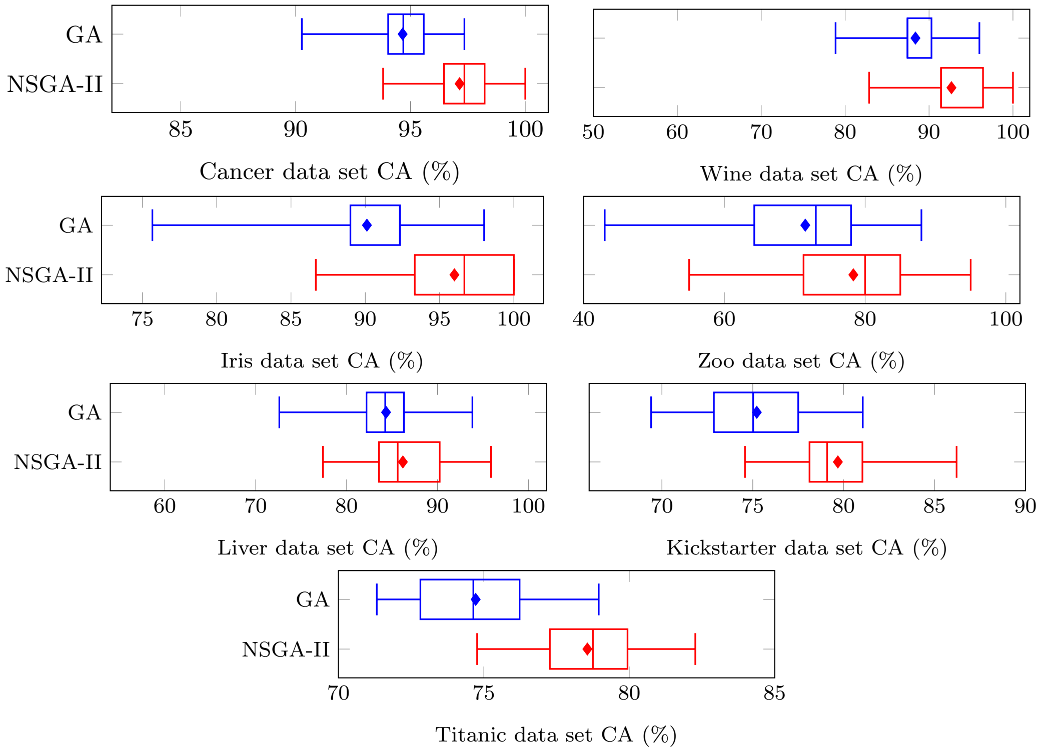
Figure 2: *The working of a generic GA [7].*

Based on findings in the literature [10], the following parameter values were selected for the GA and the NSGA-II (*i.e.,* the SOO version and MOO version, respectively): A crossover probability of 0.9, a mutation probability of 0.05, a population size of 100, and a maximum number of generations of 100. The stochastic nature of the solution methodology (ascribed to the random initialisation of solutions and the probabilistic nature of the evolutionary operators) necessitates that multiple experiments (or optimisation runs) be carried out, whilst employing a fixed set of different random number generator seeds, along with a fixed set of random initial solutions for each data set. These *matched samples* are then subjected to statistical inference procedures to determine, at a 5% level of statistical significance, whether there is a significant difference between any two (or more) samples under investigation. Both algorithms are developed and implemented in the Python 3.7 programming language.

4 Results

In order to ascertain the extent to which the proposed MOO approach delivers improved performance over an SOO appoach, a test suite comprising diverse data sets is selected. The diversity criteria are based on the size of the data set Q, the nature of the input data, the number of independent variables n, and the number of dependent variables o. The seven data sets (obtained from Dheeru & Karra Taniskidou [4]) that form part of the test suite are *Breast Cancer Wisconsin (Diagnostic)*, *Wine*, *Iris*, *Zoo*, *Liver spectroscopy*, *Kickstarter projects*, and *Titanic*. Each data set induces a different optimisation problem instance. As part of the experimental design, the number of hidden neurons is set to $m = \max\{n, o\}$ as per the commonly adopted convention in the literature [2, 6]. The activation functions employed in the hidden layer are sigmoid functions, whereas the output layer employs an identity function in conjunction with a *softmax* layer, so as to produce a vector of probabilities. At the start of each optimisation run, the network weights are randomly sampled from a continuous uniform distribution on the interval $(-1, 1)$.

The box plots in Figure 3 provide a graphical summary of the generalisation performance — *i.e., classification accuracy* (CA) — achieved by the GA and the NSGA-II in respect of each data set. Each sample comprises thirty optimisation runs. The Friedman test is an omnibus test performed in respect of each data set in order to determine whether there are statistically significant differences between the GA and the NSGA-II. A p-value of less than 0.05 denotes a difference at a 5% level of significance. Based on the findings of the Friedman test, statistically significant differences exist for each data set, except for the Liver data set, as can be seen in Table 1. In addition, the greatest performance improvement was found for the Zoo data set, although the other data sets for which statistically significant differences are present also exhibit marked performance improvements.



Cancer data set CA (%)

Wine data set CA (%)

Iris data set CA (%)

Zoo data set CA (%)

Liver data set CA (%)

Kickstarter data set CA (%)

Titanic data set CA (%)

Figure 3: *Box plots illustrating the CA achieved by the GA and the NSGA-II.*

5 Conclusion

The algorithmic performance improvements corresponding to the adoption of an MOO approach over an SOO approach in the context of training FNN weights were investigated in this paper. The celebrated GA and its powerful MOO version, *i.e.,* the NSGA-II, formed the basis of the work presented in this paper. To this end, a formal derivation of an analytical expression of FNNs was first presented, and this was followed by the mathematical representation of supervised learning. A key regularisation technique, called L^2 regularisation, was subsequently discussed, which formed a key part of the bi-objective mathematical model formulated thereafter. The GA and the NSGA-II were subsequently

Data set	Friedman test *p*-value	Performance improvement
Cancer	0	2.48%
Wine	0.0350	4.29%
Iris	0.0010	5.89%
Zoo	0.0280	6.83%
Liver	0.1234	-
Kickstarter	0.0157	4.46%
Titanic	0.0023	3.84%

Table 1: *Friedman test p-values in respect of each data set. A table entry less than 0.05 (indicated in red) denotes a difference at a 5% level of significance. The sample mean CA improvement of the NSGA-II over the GA is also indicated.*

evaluated in respect of a test suite comprising seven diverse data sets. Statistically significant performance improvements were reported for all but one data set. The results indicated that an MOO approach which incorporates a secondary regularising objective function results in marked performance improvements over a standard SOO approach.

References

[1] Alba E & Marti R, 2009, *Metaheuristic procedures for training neural networks*, Springer Science & Business Media, Berlin.

[2] Bishop CM, 2006, *Pattern recognition and machine learning*, Springer, New York (NY).

[3] Deb K, Pratap A, Agarwal S & Meyarivan TAMT, 2002, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation, **6(2)**, pp. 182–197.

[4] Dheeru D & Karra Taniskidou E, 2010. *UCI machine learning repository*, [Online], [Cited May 2019], Available from `http://archive:ics:uci:edu/ml`

[5] Eberhart R & Kennedy J, 1995, *A new optimizer using particle swarm theory*, Proceedings of the 6th International Symposium on Micro Machine and Human Science, Nagoya, pp. 39–43.

[6] Goodfellow I, Bengio Y & Courville A, 2009, *Deep learning*, MIT Press, Cambridge (MA).

[7] Holland JH, 1975, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, University of Michigan Press, Ann Arbor (MI).

[8] Russell SJ & Norvig P, 2009, *Artificial intelligence: A modern approach*, 3rd Edition, Pearson Education, London.

[9] Storn R & Price K, 1997, *Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces*, Journal of Global Optimization, **11(4)**, pp. 341–359.

[10] Talbi EG, 2009, *Metaheuristics: From design to implementation*, John Wiley & Sons, New York (NY).

[11] Vrugt JA & Robinson BA, 2007, *Improved evolutionary optimization from genetically adaptive multimethod search*, National Academy of Sciences, **104(3)**, pp. 708–711.

[12] Wolpert DH & Macready WG, 1993, *No free lunch theorems for optimization*, IEEE Transactions on Evolutionary Computation, **1(1)**, pp. 67–82.

Proceedings of the 48th ORSSA Annual Conference
pp. 106–113
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3
©2019

The Travelling Thief Problem for advancing combinatorial optimisation

KM Malan*

Abstract

When benchmarking the performance of combinatorial optimisation algorithms or teaching students about combinatorial optimisation, it is common practice to consider self-contained classical problems such as the *Travelling Salesman Problem* or the *Quadratic Assignment Problem.* Although these classical problems occur in many real-world problems, they are seldom found in isolation. Instead, real-world problems are more commonly composed of different sub-problems that interact with each other in non-trivial ways. The *Travelling Thief Problem* (TPP) was recently introduced with the aim of providing a benchmark problem that more closely matches some of the complexity evident in real-world problems. This paper provides a survey of the research emanating from the introduction of this problem. It is argued that the introduction of this problem is advancing the field of combinatorial optimisation in positive ways that is directing research to be more focused on addressing some of the challenges that arise when solving real-world problems.

Key words: Combinatorial optimisation, travelling thief problem, multiple interdependent components.

1 Introduction

The *Travelling Thief Problem* (TTP) was introduced in 2013 by Bonyadi *et al.* [3] to address the need for more realistic problems for benchmarking metaheuristics. The TTP is a combination of two classical combinatorial problems: the *Travelling Salesman Problem* (TSP) and the *Knapsack Problem* (KP). Both the TSP and the KP are NP-hard problems, making it infeasible to find the optimal solution for large instances. Heuristic and metaheuristic approaches have therefore been widely used for finding approximate solutions to these individual problems in a reasonable time.

Real world problems frequently consist of sub-problems that are interdependent. This implies that finding the best solutions of the sub-problems in isolation is not helpful, because the overall quality depends on the interaction between the sub-problems [3]. The TTP captures this complexity by combining two sub-problems that depend on each other for constructing feasible solutions of high quality. This is confirmed by Mei *et al.* [17]

*Department of Decision Sciences, University of South Africa, PO Box 392, UNISA, 0003, Republic of South Africa, email: malankm@unisa.ac.za

when they show that the computational complexity of the search required for TTP is much higher than the complexity required for the individual TSP and KP sub-components of the problem.

The parameters of the TTP can be used to control the level of interdependence between the two sub-problems. Instances of the TTP can be designed to have simple dependencies that can be decomposed and solved more easily, but instances can also be designed to have tighter dependencies so that they are not easily decomposable [5]. There are different ways of modelling the problem and choosing the most appropriate model depends on the aims of the researcher.

This paper provides an overview of research in the TTP: proposed models of the problem (§2), problem instances and characteristics (§3), and algorithmic solutions (§4). Finally, §5 discusses the ways in which the introduction of the TTP is advancing research in discrete optimisation and argues for the value in the TTP as a case study for tuition in combinatorial optimisation.

2 The travelling thief problem

The original formulation of the TTP is as follows [3]. A thief carrying a knapsack of maximum capacity, W, visits each of n cities exactly once, picking at most m items at cities to fill his knapsack. The thief starts and ends at city 1, but may only pick up items on the first visit to city 1. Each item I_k has a value p_k and a weight w_k and is available at a subset of cities: $A_k \subseteq \{1, \dots, n\}$, but the thief is only able to pick one of each item. For example, $A_5 = \{1, 10, 30\}$ indicates that item 5 is available at cities 1, 10 and 30. If the thief decides to pick item 5, he can only do so at one of these three cities. A matrix $D = d_{ij}$ defines the distances d_{ij} between cities i and j. A solution to the TTP is in the form of two vectors:

- a tour $\overline{x} = (x_1, \dots, x_n)$, where x_i is the index of a city, and
- a picking plan $\overline{z} = (z_1, \dots, z_m), z_k \in \{0 \cup A_k\}$, indicating from which city item I_k should be picked (0 implies the item is not picked at all).

The tour element, $\overline{x}$, is the same as a solution in the traditional TSP. However, the picking plan, $\overline{z}$, differs from the traditional KP in being an m-length vector of city index values, in contrast to the m-length binary vector of KP.

Four versions of the TTP model are described below.

Model 1: TTP$_1$

TTP$_1$ [3] has a single objective to maximise the benefit which equals the value of the items in the knapsack minus a knapsack rental cost of R per time unit. The thief travels at a current velocity, v_c, which is related to the current weight of the knapsack, W_c, and is defined as

$$v_c = v_{max} - W_c \left(\frac{v_{max} - v_{min}}{W} \right), \tag{1}$$

where v_{max} and v_{min} are the maximum and minimum velocities of the thief. When the knapsack is empty ($W_c = 0$), the thief travels at the maximum velocity and when the knapsack is full ($W_c = W$), the thief travels at the minimum velocity.

The time that a thief takes to travel between two cities is the distance between the two cities divided by the current velocity of the thief, and the total time for the thief to travel the full tour $\overline{x}$ with picking plan $\overline{z}$ is defined as

$$t(\overline{x}, \overline{z}) = \sum_{i=1}^{n-1}(t_{x_i,x_{i+1}}) + t_{x_n,x_1}, \tag{2}$$

where

$$t_{x_i,x_j} = \frac{d_{ij}}{v_c}.$$

Given the above, the single-objective function for TTP$_1$ is to maximise

$$f(\overline{x}, \overline{z}) = g(\overline{z}) - R \times t(\overline{x}, \overline{z}), \tag{3}$$

where $g(\overline{z})$ is the total value of the picked items, subject to the knapsack capacity, W.

Notice the interdependence between the sub-problems: the selected tour affects the time travelled via the distances (as in the usual TSP), but the picking plan also affects the time travelled via the velocity of the thief. One picking plan may result in a higher value than another, but could also result in higher rental by slowing down the thief.

Model 2: TTP$_2$

TTP$_2$ [3] is formulated as a bi-objective problem: maximisation of the value and minimisation of travel time. The thief travels at a velocity as defined for TTP$_1$, however, the value of picked items reduce over time while they are being carried in the knapsack. Each item is multiplied by a factor smaller than 1, equal to $\delta^{\lceil \frac{T_k}{C} \rceil}$, where δ is a proportion of value per time unit of travel, T_k is the total time that item k is carried in the knapsack from the time it is picked up to the end of the tour, and C is a constant.

The objective for TTP$_2$ is to simultaneously maximise the value in the knapsack, $g(\overline{x}, \overline{z})$, and minimise time, $t(\overline{x}, \overline{z})$. Notice that in this formulation, both the value and the time depend on the tour and the picking schedule, because the chosen tour affects the time each item is in the knapsack and the weight of the items affects the speed of the thief.

Model 3: BO-TTP

Wu *et al.* [28] propose a second bi-objective formulation that involves maximising the combined value function as in equation 3 while simultaneously minimising the accumulated weight. This has the effect of treating the constraint as a second objective.

Model 4: MTTP

Chand and Wagner [6] propose a formulation called the *Multiple TTP* (MTTP) with multiple thieves travelling to cities with the aim of maximising the collective profit of the group. Added to the original formulation, the MTTP includes p thieves and each thief can visit a set of n_p cities ($n_p \leq n$). Unlike the original TTP, visiting all cities is not required. All thieves start and end at city one and can visit the same city, but each item can only be picked up by one thief. This modified formulation of the TTP is more complex as it involves multiple tours and packing plans.

3 Travelling thief problem instances and characteristics

When the TTP was originally introduced, Bonyadi *et al.* [3] proposed a procedure for generating problem instances and later provided a set of forty-five TTP instances [4]. At the same time, Polyakovskiy *et al.* [21] established a large benchmark suite of 9 720 different TTP instances to cover a wide range of features and different levels of correlation between the sub-components. This suite has been the basis for a number of TTP competitions. The first competition at the 2014 Congress on Evolutionary Computation focused on the single-objective formulation of the problem, TTP_1, while the most recent competition at the 2019 Genetic and Evolutionary Computation Conference is focused on the bi-objective formulation, TTP_2.

A few studies have focused on understanding the complexity and nature of the TTP problem. Wu *et al.* [26] investigated the effect of the knapsack renting rate on problem difficulty and showed how this parameter could be manipulated to create hard problem instances for evolutionary algorithms. Polyakovskiy and Neumann [22] showed that a simpler version of the TTP with a fixed route is NP-hard for both the constrained and unconstrained cases. Wagner *et al.* [25] investigated the link between basic features of the TTP and algorithm performance and found that the most important features for distinguishing algorithm performance were the knapsack capacity and the renting rate.

El Yafrani *et al.* [10] performed a fitness landscape analysis based on a full enumeration of the search space of small TTP instances. They used local optimal networks [20] to model the global structure of search space using the Cartesian product between the neighbourhoods of the two sub-problems as the basis for the definition of neighbourhood. Their analysis showed that there is a direct correlation between lower knapsack capacity and the problem hardness.

4 Algorithmic approaches to solving the TTP

Most of the techniques reported in the literature to solving the TTP are approximate heuristic or metaheuristic approaches. One study by Wu *et al.* [27] proposed three exact techniques to solving the TTP, using dynamic programming, branch and bound search, and constraint programming. These exact techniques are useful for evaluating the success of approximate techniques, but are only applicable to small TTP instances.

The simplest heuristic approach to solving the TTP is to tackle the two sub-problems sequentially, by first solving the TSP component of the problem — using an approach such as the *Chained Lin-Kernighan* (CLK) [1] — and then using a heuristic to choose a picking plan for the given TSP solution. A number of studies have used this approach, differing only in the heuristics for the KP sub-problem using the route found by CLK [4, 12, 13, 21]. Mei *et al.* [17] extended this two-stage approach to use a population of tours generated by CLK and later proposed the evolution of the gain and picking heuristic functions using genetic programming [18]. Wagner [24] deviated from the practice of solving the TSP component using CLK by using ant colony optimisation for finding tours, followed by heuristics for the KP sub-component and further local search on the tours. He found that this approach was able to find better quality solutions by considering longer tours, but only for instances with up to 250 cities and 2 000 items.

Faulkner *et al.* [12] performed a comparative analysis of eleven different heuristic approaches on seventy-two instances of the TTP and concluded that there is no single best heuristic approach to solving the TTP. Given the fact that there is no single best heuristic approach to solving TTP (known as *performance complementarity* [14]), Martins *et al.* [16] and El Yafrani *et al.* [11] proposed hyper-heuristic approaches to solving the TTP that select the best combination of heuristics for a particular problem instance.

An alternative to the two-stage sequential heuristic approach includes some mechanism for influence between the two sub-problems. Bonyadi *et al.* [4] proposed CoSolver, that solves the two sub-problems in parallel, but includes revision of solutions through “negotiation” between the two solvers. El Yafrani and Ahiod [7] proposed a modification to CoSolver utilising local search metaheuristics (a hill climber and simulated annealing) and Nieto-Fuentes *et al.* [19] proposed a further improvement to CoSolver by using guided local search to escape local optima in the tour optimisation stage. A different approach to combining the sub-problems was proposed by El Yafrani & Ahiod [9]. They used the usual two-stage approach of first finding a CLK tour followed by a heuristic-based picking plan, but then implemented a local search using a joined neighbourhood formed by the Cartesian product of the neighbourhoods of the two sub-problems. Although their approach gave good results for small instances, the high computational complexity resulting from the Cartesian product neighbourhood is a limitation of the approach.

All of the above approaches involved solving the first, single-objective model, TTP_1. In contrast, El Yafrani *et al.* [8] investigated solving the bi-objective model, TTP_2, using the NSGA-II framework to search for a Pareto set of solutions. Blank *et al.* [2] proposed three approaches to the bi-objective TTP_2: two deterministic approaches and the third using NSGA-II. They showed that the NSGA-II approach provided the best performance of the algorithms. Wu *et al.* [28] proposed a solution to the second bi-objective formulation, BO-TTP, using an evolutionary approach to constructing tours and a dynamic programming approach to solving the picking plan.

In the case of the third model, the MTTP, only the authors that introduced the formulation, Chand & Wagner [6], proposed a number of heuristics for solving the MTTP.

5 Advancing the field of combinatorial optimisation

Considering the algorithmic solutions to the TTP proposed in the literature, all proposed solutions use a combination of different algorithmic approaches. Most solutions use established heuristics that have been in use for decades in combination with newly designed heuristics, local search and/or established metaheuristics. The performance complementarity [14] of different algorithmic approaches to the TTP is clear in the success of algorithm selection in solving the problem, as investigated by Wagner *et al.* [25]. In the context of a complex problem with interdependent subcomponents, introducing a "new" metaheuristic inspired by some arbitrary phenomenon in nature [23] would be counterproductive. The challenge is rather in finding efficient and effective ways of combining established techniques.

Another challenge for researchers that has become evident through the TTP is the need for new ways of understanding search spaces and neighbourhood of these interdependent subspaces. Fitness landscape analysis is commonly used in the evolutionary computation community to understand complex problems and guide the choice of appropriate algorithms [15]. The notion of neighbourhood is central to the concept of fitness landscapes. Preliminary work in the landscape analysis of TTP [10] defined neighbourhood based on the Cartesian product between the neighbourhoods of the two sub-problems. Alternative notions of neighbourhood are needed for landscape analysis of the TTP to be computationally feasible.

The multi-faceted nature of the TTP makes it an ideal case study for educational modules in combinatorial optimisation. The TSP and KP sub-components can be introduced individually at first, covering permutation-based and binary search spaces. Small instances can be used to introduce exact solvers, moving to heuristic solvers and metaheuristics for larger instances. Using a problem like TTP will convey the essential message to students that there is value in different approaches and it is in the combination of approaches that we are able to develop good solutions to complex problems.

6 Conclusion

There has been a flurry of research activity in the evolutionary computation community dedicated to the newly introduced *Travelling Thief Problem* (TTP). It is argued that this benchmark problem has had a positive effect on research in combinatorial optimisation, resulting in researchers focusing on the combination of different approaches from classical heuristic methods to newer evolutionary and swarm-based approaches. The problem has also highlighted a new direction for research in fitness landscape analysis to accommodate these complex interdependent sub-spaces. Furthermore, the problem provides an excellent case study for tuition in combinatorial optimisation.

References

[1] Applegate D, Cook W & Rohe A, 2003, *Chained Lin-Kernighan for large traveling salesman problems*, INFORMS Journal on Computing, **15(1)**, pp. 82–92.

[2] Blank J, Deb K & Mostaghim S, 2017, *Solving the bi-objective traveling thief problem with multi-objective evolutionary algorithms*, Proceedings of the 9th International Conference on Evolutionary Multi-Criterion Optimization, Münster, pp. 46–60.

[3] Bonyadi MR, Michalewicz Z & Barone L, 2013, *The travelling thief problem: the first step in the transition from theoretical problems to realistic problems*, Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancún, pp. 1037–1044.

[4] Bonyadi MR, Michalewicz Z, Przybylek MR & Wierzbicki A, 2014, *Socially inspired algorithms for the travelling thief problem*, Proceedings of the 2014 Genetic and Evolutionary Computation Conference, Vancouver, pp. 421–428.

[5] Bonyadi MR, Michalewicz Z, Wagner M & Neumann F, 2019, *Evolutionary computation for multicomponent problems: Opportunities and future directions*, pp. 13–30 in Datta S & Davim J (Eds.), *Optimization in Industry: Present practices and future scopes*, Springer, Cham.

[6] Chand S & Wagner M, 2016, *Fast heuristics for the multiple traveling thieves problem*, Proceedings of the 2016 Genetic and Evolutionary Computation Conference, Denver (CO), pp. 293–300.

[7] El Yafrani M & Ahiod B, 2016, *Population-based vs. single-solution heuristics for the travelling thief problem*, Proceedings of the 2016 Genetic and Evolutionary Computation Conference, Denver (CO), pp. 317–324.

[8] El Yafrani M, Chand S, Neumann A, Ahiod B & Wagner M, 2017, *Multi-objectiveness in the single-objective traveling thief problem*, Proceedings of the 2017 Genetic and Evolutionary Computation Conference Companion, Berlin, pp. 107–108.

[9] El Yafrani M & Ahiod B, 2017, *A local search based approach for solving the travelling thief problem: The pros and cons*, Applied Soft Computing, **52**, pp. 795–804.

[10] El Yafrani M, Martins MSR, Krari ME, Wagner M, Delgado MRBS, Ahiod B, & Lüders R, 2018, *A fitness landscape analysis of the travelling thief problem*, Proceedings of the 2018 Genetic and Evolutionary Computation Conference, Kyoto, pp. 277–284.

[11] El Yafrani M, Martins MSR, Wagner M, Ahiod B, Delgado MRBS & Lüders R, 2018, *A hyperheuristic approach based on low-level heuristics for the travelling thief problem*, Genetic Programming and Evolvable Machines, **19(1-2)**, pp. 121–150.

[12] Faulkner M, Polyakovskiy S, Schultz T & Wagner M, 2015, *Approximate approaches to the traveling thief problem*, Proceedings of the 2015 Genetic and Evolutionary Computation Conference, Madrid, pp. 385–392.

[13] Gupta BC & Prakash VP, 2015, *Greedy heuristics for the travelling thief problem*, Proceedings of 39th National Systems Conference, Noida, pp. 1–5.

[14] Kerschke P, Hoos HH, Neumann F & Trautmann H, 2019, *Automated algorithm selection: Survey and perspectives*, Evolutionary Computation Journal, **27(1)**, pp. 3–45.

[15] Malan KM & Engelbrecht AP, 2013, *A survey of techniques for characterising fitness landscapes and some possible ways forward*, Information Sciences, **241**, pp. 148–163.

[16] Martins MSR, El Yafrani M, Delgado MRBS, Wagner M, Ahiod B & Lüders R, 2017, *HSEDA: a heuristic selection approach based on estimation of distribution algorithm for the travelling thief problem*, Proceedings of the 2017 Genetic and Evolutionary Computation Conference, Berlin, pp. 361–368.

[17] Mei Y, Li X & Yao X, 2014, *Improving efficiency of heuristics for the large scale traveling thief problem*, Proceedings of SEAL 2014: Simulated Evolution And Learning, Dunedin, pp. 631–643.

[18] Mei Y, Li X, Salim F & Yao X, 2015, *Heuristic evolution with genetic programming for traveling thief problem*, Proceedings of the 2015 IEEE Congress on Evolutionary Computation, Sendai, pp. 2753–2760.

[19] Nieto-Fuentes R, Segura C & Valdez SI, 2018, *A guided local search approach for the travelling thief problem*, Proceedings of the 2018 IEEE Congress on Evolutionary Computation, Rio de Janeiro, pp. 1–8.

[20] Ochoa G, Verel S, Daolio F & Tomassini M, 2014, *Local optima networks: A new model of combinatorial fitness landscapes*, pp. 233–262 in Richter H & Engelbrecht A (Eds.), *Recent advances in the theory and application of fitness landscapes*, Springer, Heidelberg.

[21] Polyakovskiy S, Bonyadi MR, Wagner M, Michalewicz Z & Neumann F, 2014, *A comprehensive benchmark set and heuristics for the traveling thief problem*, Proceedings of the 2014 Genetic and Evolutionary Computation Conference, Vancouver, pp. 477–484.

[22] Polyakovskiy S & Neumann F, 2017, *The packing while traveling problem*, European Journal of Operational Research, **258(2)**, pp. 424–439.

[23] Sörensen K, 2015, *Metaheuristics – the metaphor exposed*, International Transactions in Operational Research, **22**, pp. 3–18.

[24] Wagner M, 2016, *Stealing items more efficiently with ants: A swarm intelligence approach to the travelling thief problem*, Proceedings of the 10th International Conference on Swarm Intelligence, Brussels, pp. 273–281.

[25] Wagner M, Lindauer M, Misir M, Nallaperuma S & Hutter F, 2018, *A case study of algorithm selection for the traveling thief problem*, Journal of Heuristics, **24(3)**, pp. 295–320.

[26] Wu J, Polyakovskiy S & Neumann F, 2016, *On the impact of the renting rate for the unconstrained nonlinear knapsack problem*, Proceedings of the 2016 Genetic and Evolutionary Computation Conference, Denver (CO), pp. 413–419.

[27] Wu J, Wagner M, Polyakovskiy S & Neumann F, 2017, *Exact approaches for the travelling thief problem*, Proceedings of Simulated Evolution and Learning, Shenzen, pp. 110–121.

[28] Wu J, Polyakovskiy S, Wagner M & Neumann F, 2018, *Evolutionary computation plus dynamic programming for the bi-objective travelling thief problem*, Proceedings of the 2018 Genetic and Evolutionary Computation Conference, Kyoto, pp. 777–784.

Proceedings of the 48th ORSSA Annual Conference
pp. 114–123
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3
©2019

The use of attrition analyses to assess the stability of retail indeterminate maturity deposits

WD Schutte*

Abstract

South African banks are required to report their liquidity risk position to the *South African Reserve Bank* (SARB) on a monthly basis. Amongst others, banks should report information relating to the stability and volatility of balances deposited by clients at the bank. Regulatory guidance on the manner in which stability should be assessed is vague and non-specific. Grounded on a literature review, attrition analysis is proposed as one possible methodology to estimate the effective maturity and stability of balances deposited in retail indeterminate maturity products (deposit products without an explicit contractual maturity date), such as savings and cheque accounts. An algorithm is provided to perform the attrition analysis, which is used as input in the estimation of the effective maturity and the *Annually Compounded Attrition Rate* (ACAR). The ACAR is then used as a measure to segment the deposit balances into a stable and a volatile proportion. The volatile proportion of balances are directly equated to the ACAR, while the remainder of the balances are regarded as stable. The proposed methodology is applied to anonymised month-end balance data of an indeterminate maturity product, sourced from a large retail bank in South Africa. The results are useful and can act as a departing point for more sophisticated methods, which are proposed in the conclusion.

Key words: Indeterminate maturity, attrition analysis, stable deposits, volatile deposits, Liquidity Coverage Ratio (LCR).

1 Introduction

A bank functions by obtaining funds from clients, both retail and wholesale, repo-style funding, interbank loans and funding from other business lines within the bank itself. The bank then utilises these funds for investment purposes (buildings, business entities), lending of funds to customers and businesses (mortgages, overdrafts, student loans) and to finance their trading, hedging and insurance activities, to name but a few examples [6]. This study will focus on the funding side of the banking book with specific reference to the funding obtained from clients in the form of retail indeterminate maturity deposits. The classification of these deposits into a stable proportion and a less stable proportion is essential for several reasons that will be elucidated.

*Centre for Business Mathematics and Informatics, North-West University, Private bag X6001, Potchefstroom, 2531, Republic of South Africa, email: wd.schutte@nwu.ac.za

The phrase "indeterminate maturity deposits" has been mentioned, but no formal definition has been given. Although the next section will deal with the problem description and some formal definitions of concepts used throughout, the concept of indeterminate maturity deposits must be clarified. Instruments with an indeterminate maturity can be defined as funding instruments with no predetermined or negotiated contractual maturity [9]. This definition implies, in terms of application in the banking environment, that the balances of indeterminate maturity instruments may be indefinitely invested with the bank or withdrawn (partially or completely) without prior notice given to the bank. Thus, the instrument has an unknown lifetime and unknown associated cash flows. Examples include savings accounts, cheque accounts, call accounts and any positive client balance on a credit card.

The remainder of this paper is structured as follows: In §2 a problem description is provided to understand the importance and role of indeterminate maturity deposits in the reporting activities of banks. Section 3 contains a literature review, which substantiate the solution methodology proposed in §4. Section 5 presents the results following the application of the methodology and §6 concludes the paper with some future research ideas to expand the methodology.

2 Problem description and regulatory setting

Globally, the banking industry is meticulously regulated. The *Bank for International Settlements* (BIS) is an international organisation that strives towards the establishment of worldwide regulations that promote global monetary and financial stability. The *Basel Committee on Banking Supervision* (BCBS) is housed within the BIS and is the primary global standard setter for the prudential regulation of banks. On the local front, the South African banking system is systematically organised and regulated by the *South African Reserve Bank* (SARB). Legislation governing the financial sector in South Africa is primarily the Banks Act [1], which propagates the achievement of a stable and sound banking system that offers efficient and reliable services in the interest of the depositors and lenders (the clients) of the banks and the economy as a whole. South African banks also need to adhere to international regulations such as Basel III [2], in a manner prescribed by the SARB. With respect to the achievement of financial stability, banks are obliged to report vast amounts of detail to the SARB, monthly and annually (and in some instances, daily). Two of the risk reporting instruments relevant for this study are the liquidity risk return (BA 300) and the reporting of the Basel III *Liquidity Coverage Ratio* (LCR), including the Net Stable Funding Ratio [3].

First, banks need to report on a monthly basis to the SARB their liquidity risk position (among others) on a standard reporting template called the BA 300 [12]. Amongst others, this template contains items that relate to the stability and volatility of deposits (lines 6 and 7 of the BA 300). The directives, definitions and interpretations for completion of this monthly return concerning liquidity risk provide limited guidance on how the stability or volatility should be determined as evident from the following quotations from SARB [12]:

> "...volatile deposits shall include any deposit likely to be quickly withdrawn in

> a stress situation, including deposits received from private individuals."
>
> "...stable deposits shall include any deposit deemed by the reporting bank to be less liquid, that is, deposits other than volatile deposits, including deposits received from private individuals."
>
> "...a bank shall duly document the specific definitions and/or criteria applied by the bank to distinguish between 'stable deposits' and 'volatile deposits' and, at the request of the Registrar, the bank shall in writing submit to the Registrar the said specific definitions and/or criteria."

Secondly, when banks report their LCR, certain line items (lines 115, 116, 120 and 121) refer to "stable funding" and "less stable funding" [12]. The directives in this regard are slightly clearer and certain minimum thresholds are provided depending on characteristics of the deposit instrument. Yet, banks must ensure that these minimum thresholds are prudent and should validate its appropriateness.

In summary, banks need to report monthly on the segmentation of deposit balances as stable or volatile, without receiving explicit guidance from the regulatory authority. Therefore, banks need to develop sound and straightforward methodologies to substantiate the classification of certain deposit balances (or portions thereof) as stable and volatile (or less stable). For deposit instruments without an explicit maturity date (indeterminate maturity), such classification is especially relevant since these deposits can be indefinitely invested with the bank or withdrawn (partially or completely) without prior notice given to the bank.

3 Literature review

The broad problem of modelling indeterminate maturity deposit balances has been studied by several authors [5, 6, 7, 10, 11]. The overarching problem that is investigated by these authors relates to the calculation of the effective maturity of indeterminate maturity deposits. The effective maturity of an instrument is an important concept since it represents (on average) the duration until the balance in that instrument will reduce to zero. Effective maturity can thus be defined as the maturity value assigned to an instrument after the observation of the behaviour of that instrument over a period of time. The effective maturity of a deposit instrument, therefore, conveys information about the stability of the balances in that instrument since the maturity date signifies the point in time where the balances will be withdrawn. While it is simple to derive the contractual maturity of, say, a fixed deposit, it is much more difficult to obtain an observed or realised value for the maturity of an instrument with no contractual maturity, such as a call deposit or a savings account.

Research relating to indeterminate maturity deposits also spans the slightly more complex issue of segmenting balances into a stable or core part, and a more volatile or non-core part [6, 9, 10]. The core part of balances is said to be a stable, "long-term" (still indeterminate maturity) foundation amounts residing within these deposit instruments. From another perspective, core deposits can be defined as the balances in accounts that will remain with the bank during a time of a financial distress [9]. It is also possible to link the concept of calculating the effective maturity (as described above) to segmenting balances into a stable and volatile part. This is similar to the proposed method, which will be elaborated on in §4.

Also, when the literature is considered, it is evident that two types of modelling strategies should be considered:

- Models relating to the balances of indeterminate maturity deposits; and
- Models relating to the interest rates offered on indeterminate maturity deposits.

Due to constraints on the length of the paper, only models relating to the balances of indeterminate maturity deposits will be considered.

Matz [10], however, encourages the use of balance and interest rate models simultaneously to ensure that the interaction effect of interest rates on deposit balances is accounted for. On a practical level, this supports the assumption that indeterminate maturity deposits are replaceable, and that this funding can be relied on as long as the bank is willing to pay attractive interest rates compared to alternative instruments.

Bessis [6] proposes several strategies to deal with indeterminate maturity deposits, also referring to both balance and interest rate models:

- Make assumptions with respect to the amortisation of indeterminate maturity deposits, *e.g.,* using a yearly amortization rate of 5% or 10%.
- Divide deposits into stable (core) and unstable (non-core) balances. The core deposits remain constantly as a permanent resource. The volatile fraction is treated as short-term debt. This approach is closer to reality, although no technique is proposed for splitting deposits into core and non-core fractions.
- Make projections by modelling some observable variable correlated with the outstanding balances of indeterminate maturity deposits. Such variables include the trend of economic conditions and possible interest rate variables. The limitation of this method is that not all parameters that impact indeterminate maturity deposits can be considered, as there may be a limited amount of data available on these factors. Forecasting of these factors might also pose a challenge. This approach is, however, regarded as the closest to reality.

Furthermore, Matz [10] identifies methods that can be used as tools to establish accurate maturity assumptions for a bank's deposit balances:

- Decay or attrition analysis, where the attrition behaviour or roll-out rate of deposit balances are observed over time,
- segmented volume analysis, where various statistical methods are applied to identify distinct segments of the deposit balances,
- replicating portfolios, which implies a statistical method that matches known (and easily priced) portfolios with known maturities to the indeterminate maturities portfolio,
- a combination of replicating portfolios and segmented volume analysis, and
- highly quantitative statistical modelling which incorporates internal and external variables.

In terms of reporting requirements, more recent literature was published by the *European Banking Authority* (EBA) in a discussion paper on retail deposits for the purposes of LCR reporting [8]. Bašič [5] also investigated methods of assessing the run-off factor of retail deposits for the purpose of LCR reporting. These authors do not address the problem of segmenting balances into a stable and volatile part. In the next section, attrition analysis is proposed as a potential deposit balance model to aid in the segmentation of balances into stable and volatile proportions.

4 Potential solution methodology

Attrition analysis is simply the statistical analysis of the rate at which deposit volumes reduce over time [10]. Attrition may be caused by the withdrawal of funds from still-existing accounts or from the closure of accounts. Attrition analysis involves the calculation of an attrition rate representing the annual decay of deposit balances. This attrition rate can be interpreted as the balances that are assumed to "mature" each year. In essence, the attrition rate can therefore be converted to an effective maturity (the point in time when all balances have been withdrawn), which can then be linked to the concept of segmenting deposit balances into a stable and volatile proportion. This procedure is proposed as one possible solution to the described problem and will be discussed in detail.

To estimate the attrition rate for specific retail deposit products, a random sample of accounts should be selected. Attrition analysis then looks at each account individually. The analysis therefore, requires historical deposit balances for every account in the sample. Month-end or daily balances averaged over each calendar month can be used. The time period for the attrition study needs to be considered carefully. If the period is too long, historical behaviour will affect the data even if the data are no longer relevant. On the other hand, if the period is too short, it may not be representative of both rising and falling interest rate environments. The best strategy is to select a time period that captures the most recent complete interest rate cycle — the period from the most recent rate high through the most recent rate low, or *vice versa*. Such a period will most probably include rising and falling rate environments. The rate of decline in the sample account balances is then expressed as an annual percentage of change. However, the rate of change is not the same in every month; therefore the function is curvilinear. In other words, the rate of decrease is larger in the earlier months and smaller in the later months. Consequently, attrition rates over time can most clearly be seen when it is graphed. In §5 the results are discussed using several of these graphs.

For this study, anonymised data were obtained from a large retail bank containing the month-end balances of retail clients for an indeterminate maturity product. The following algorithm, illustrated in Figure 1, is proposed to perform attrition analysis on such a data set.

Algorithm 1 (Effective maturity using attrition analysis) *Estimate the effective maturity of an indeterminate maturity product by applying attrition analysis to the month-end balances of retail clients.*

1. *Choose a starting month, say SM1, to generate an attrition "pattern" for only those accounts presently open in that month, containing a positive account balance. Only these accounts will be considered across time in the remaining steps.*
2. *On an account level, record the balance of each account monthly for the duration of the data sample. For each account, ensure that the balance in each subsequent month is always less than or equal to the balance in the previous month (i.e., an increase in the account's balance from one month to the next is not tolerated — if present, assign the same balance in that month as in the previous month). The reason behind this assumption is to ensure that the estimates are conservative*[1] *and that an actual attrition rate is observed (and not a growth rate).*
3. *Calculate the total monthly balance across all the accounts considered to obtain a series of calculated month-end balances for each month in the data sample.*
4. *Take every balance in the series generated in step 3 and divide it by the balance of the starting month. This will translate into a retention rate for a growing time period, i.e., a one-month retention rate, a two-month retention rate, etc.*
5. *Select a different starting month than the one selected in step 1, say SM2 and repeat steps 1–4 until several attrition patterns are obtained, each for a different starting month. These starting months should be chosen while considering the interest rate cycle, i.e., choose starting months at the beginning of an increasing interest rate cycle and at the beginning of a decreasing interest rate cycle, and at various other points in the interest rate cycle. Furthermore, dependence among these starting months can be avoided by selecting only a sample (without replacement) of accounts per starting month.*
6. *Select the starting month with the largest number of observations in the attrition pattern (from Table 1, this will be the attrition pattern starting in March 2015).*
7. *Use the last value in the attrition series (i.e., the terminal retention percentage) TR (from Table 1, this is 64.31%) to calculate the annually compounded attrition rate ACAR as*

   $$ACAR = 1 - (TR)^{\left(\frac{1}{n}\right)}$$

   where n represent the time interval (in years) over which the terminal retention rate was calculated (from Table 1, $n = 3.833$ years resulting in $ACAR = 0.1087$).
8. *The effective maturity EM (in years) is then the reciprocal of ACAR, i.e., $EM = 1/ACAR$. From Table 1, $EM = 9.193$ years.*

Table 1 represents potential results that could be obtained after selecting three different starting months and executing the algorithm described above. The entries in the table represent the percentage of the original balance that was retained at a specific point in time.

[1] Although regulators prefer conservatism, a bank would strive towards models that resemble reality. A possible methodology to assess the level of conservatism is to allow account balances to increase month-on-month (see step 2 of the algorithm) and to calculate the retention rate (which might be greater than 100%). The difference between the retention rate calculated this way and the retention rate calculated with the described algorithm may provide some assessment of the level of conservatism.

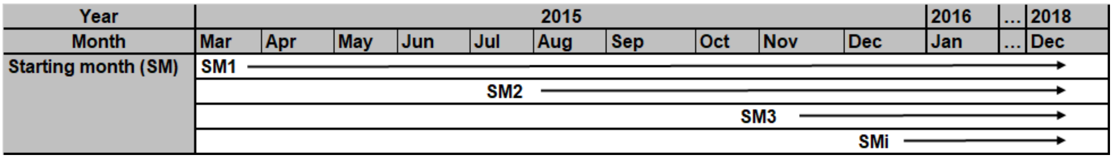


Figure 1: *Illustrating the algorithm to develop attrition patterns for different starting months.*

Retention rate %		Mar-15	Apr-15	May-15	Jun-15	Jul-15	Aug-15	Sep-15	Oct-15	Nov-15	Dec-15	...	Dec-18
Starting month (SM)	Mar-15	100.00%	94.71%	93.78%	93.69%	93.55%	92.34%	91.88%	91.22%	89.03%	88.01%	...	64.31%
	Aug-15						100.00%	98.71%	96.78%	94.69%	94.55%	...	70.00%
	Nov-15									100.00%	98.20%	...	75.66%

Table 1: *Concatenated potential result after three iterations of the algorithm.*

Furthermore, segmenting deposit balances into stable and volatile parts can be done naturally using the annual compounded attrition rate. If one assumes that the volatile proportion of balances represents those balances that will be withdrawn within the next year, it is just to equate the $ACAR$ to the volatile proportion of the deposit balances. The stable proportion of the deposit balances will then be equal to the remainder (*i.e.*, $1 - ACAR$). Mathematically, one could define *Volatile deposit balance* $= ACAR \times$ *Deposit balance* and *Stable deposit balance* $= (1 - ACAR) \times$ *Deposit balance*. From the above example, suppose $ACAR = 0.1087$; this methodology could be used to motivate that 10.87% of the deposit balances are volatile while the remaining 89.13% is stable. In the next section, the results of the application of the proposed methodology to retail indeterminate maturity products are discussed.

5 Results

As mentioned, anonymised data containing the month-end balances of retail clients for an indeterminate maturity product were obtained from a large retail bank. The data covered a period of forty-three months. Figure 2 represents the attrition patterns developed from the data for eleven different starting months, following the methodology proposed in §4.

The annually compounded attrition rate is estimated at 16.72%, yielding an estimated effective maturity of 5.97 years. Although these results are useable, much greater insight could be obtained from investigating qualitative aspects of the attrition patterns. For example, it is evident that there is a significant month-effect present in the attrition patterns, such as the January-effect where there is a noticeable drop in the percentage of the balances retained.

The use of this attrition analysis can be further expanded beyond what was discussed in

§4, in order to allow for some more advanced modelling opportunities. One option is to disregard the month-effect and align all the attrition patterns to the same starting position (*i.e.,* by months since starting month).

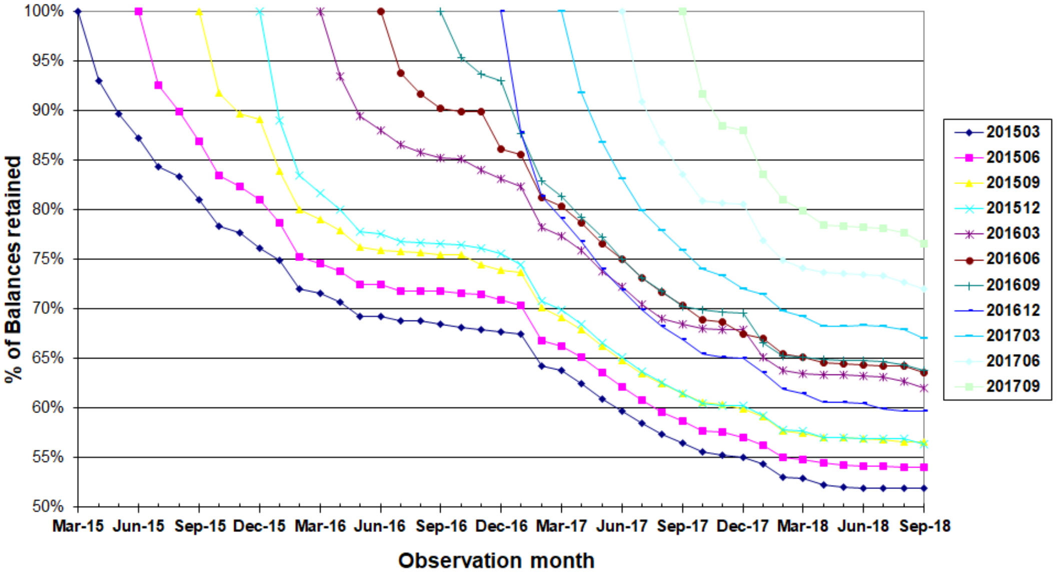


Figure 2: *Attrition patterns generated for eleven different starting months.*

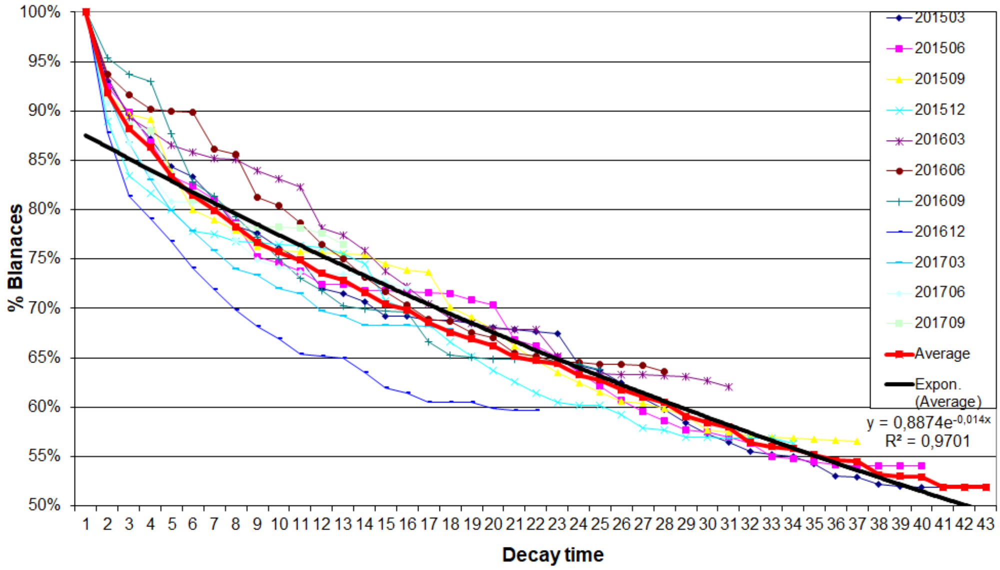


Figure 3: *Attrition patterns aligned to attrition time, with average attrition pattern and exponential curve fitted.*

Figure 3 contains the graphical representation of aligning the attrition patterns. It is now possible to create an "averaged" attrition pattern, followed by fitting different functional curves (*i.e.,* power, logarithmic, exponential, *etc.*) to this average pattern. The results of this process are also illustrated in Figure 3 after fitting an exponential curve.

The extrapolation of the functional curve can be utilised to generate another estimate for the effective maturity. Suppose a conservative estimation is desired: One could assume that when the remaining balances reach a level of, say, 15%, it is considered to be an indication that no stable balances remain. In the provided results, this extrapolation can easily be done, resulting in an estimated effective maturity of 10.87 years ($ACAR = 9.19\%$).

6 Conclusions and further research

The proposed methodology to utilise attrition analysis to estimate effective maturity and the annually compounded attrition rate in assessing the stability of retail indeterminate maturity deposits is understandable and straightforward. Banks are obliged to report and substantiate to the *South African Reserve Bank* (SARB) the methodology followed in classifying parts of their deposit balances as stable. In this regard, the SARB prefers simplified methods above sophisticated methods that could easily become “black boxes” from which it is difficult to comprehend the results produced by the model [4]. Attrition analysis is, however, not without weakness [9], and banks should acknowledge these limitations, such as the fact that no new business or activity is taken into account. The proposed solution in this paper also did not address the interest rate effect.

In addressing these limitations, attrition analysis forms the basis for the investigation of more sophisticated approaches. As noted by Matz [10], both balance and interest rate models should be considered. It is proposed to investigate regression analysis of the average attrition pattern against predictor variables that are representative of the interest rates offered on the deposit instruments. Time series analysis could also be performed by decomposing the attrition series into a long term trend, a seasonal component, a cyclical component and some irregular variation. The long-term trend can potentially be used to represent the stable part of the deposits balances.

References

[1] Banks Act 94 of 1990, South Africa, [Statute], Government printer, **10272**, Pretoria.

[2] Basel Committee on Banking Supervision (BCBS), 2011, *Basel III: A global regulatory framework for more resilient banks and banking systems*, [Online], [Cited May 24th, 2019], Available from `https://www.bis.org/publ/bcbs189.pdf`

[3] Basel Committee on Banking Supervision (BCBS), 2013a, *Basel III: The Liquidity Coverage Ratio and liquidity risk monitoring tools*, [Online], [Cited May 24th, 2019], Available from `https://www.bis.org/publ/bcbs238.pdf`

[4] Basel Committee on Banking Supervision (BCBS), 2013b, *The regulatory framework: balancing risk sensitivity, simplicity and comparability*, [Online], [Cited June 4th, 2019], Available from `https://www.bis.org/publ/bcbs258.pdf`

[5] Bašič, AM, 2015, *Model for determining the stability of retail deposits with higher outflow rates*, Naše Gospodarstvo / Our Economy — Journal of Contemporary Issues in Economics and Business, **61(5)**, pp. 12–22.

[6] Bessis, J, 2015, *Risk management in banking*, 4th edition, John Willey & Sons Ltd, West Sussex.

[7] Cipu, EC & Udriste, S, 2009, *Estimating non-maturity deposits*, Proceedings of the 9th WSEAS international conference on simulation, modelling and optimization, Budapest, pp. 369–374.

[8] European Banking Authority (EBA), 2013, *Consultation paper: Draft guidelines on retail deposits subject to different outflows for purposes of liquidity reporting under Regulation (EU) No 575/2013 (Capital Requirements Regulation – CRR) EBA/CP/2013/34*, [Online], [Cited May 23rd, 2019], Available from `http://www.eba.europa.eu`

[9] Matz, LM, 2002, *Liquidity risk management*, Sheshunoff Information Services Inc, Austin (TX).

[10] Matz, LM, 2004, *Interest rate risk management*, Sheshunoff Information Services Inc, Austin (TX).

[11] Musakwa, FT, 2013, *On measuring bank funding liquidity risk*, Paper presented at the Actuarial Society of South Africa's 2013 Convention, Sandton.

[12] South African Reserve Bank (SARB), 2012, *Government notice: Banks act 1990 (act no. 94 of 1990)*, Government Gazette, **35950**, Pretoria.

Proceedings of the 48th ORSSA Annual Conference
pp. 124–131
http://nand8a.com/orssa/conferences/2019/ORSSA2019_Proceedings.pdf

ORSSA Proceedings
ISBN 978–1–86822–701–3
©2019

Using structural semi-supervised learning to tag resources for resource scarce languages

AC Griebenow[*] DP Snyman[*] GR Drevin[*]

Abstract

Natural language processing often makes use of machine learning-based classification techniques for the analysis of text-based language. These approaches rely on pre-annotated training data from which underlying patterns in the data can be utilised to predict classes for unseen examples. In the case of resource scarce languages, little to no pre-annotated data is readily available. This study investigates and applies techniques for optimising the use of available annotated resources for Afrikaans. The aim is to determine whether structural semi-supervised learning can be effectively applied to improve the accuracy, given the underlying resource scarcity, of a supervised learning part of speech tagger. Existing semi-supervised learning algorithms, with specific focus on structural semi-supervised learning, are investigated. Structural semi-supervised learning is implemented by means of an *Alternating Structure Optimisation* (ASO) algorithm based on *Singular Value Decomposition* (SVD). The SVD-ASO algorithm attempts to extract the shared structure of untagged data using auxiliary problems before training a tagger. The use of untagged data during the training of a tagger leads to an error rate reduction of 1.67% compared to the baseline. Even though the error rate reduction does not prove to be statistically significant in all cases, the results show that it is possible to improve the accuracy in some cases.

Key words: Resource-scarce language, machine learning, semi-supervised learning, structural learning, human language technology, natural language processing.

1 Introduction

The development of resources in any language is an expensive process. The result is that, due to this high cost, many languages, including the indigenous languages of South Africa, can be classified as being resource scarce, or lacking in natural language processing resources. This study investigates and applies techniques and methodologies for optimising the use of available resources and improving the accuracy of a part of speech tagger using Afrikaans as resource scarce language and aims to determine whether structural semi-supervised learning can be effectively applied to improve the accuracy of a supervised learning tagger for Afrikaans.

[*] School of Computer Science and Information Systems, North-West University, Private Bag X6001, Potchefstroom, 2520, Republic of South Africa, email: agriebenow@gmail.com, dirk.snyman@nwu.ac.za, gunther.drevin@nwu.ac.za

The semi-supervised algorithm that is implemented is based on the structural learning method proposed by Ando & Zhang [1, 2].

2 Stuctural learning

Structural learning explores the pre-existing structures that are present in the data by exploiting those structures that are prone to make good predictions, as discussed in §3. In this case it is specifically applied to the semi-supervised learning problem as valuable information in the form of structures is extracted from unlabelled data. These newly learned structures have good prediction ability and is then carried over to the target problem which is the supervised learning of labelled data. Structural learning is an attempt to improve a classifier by identifying similarities or shared structures between multiple prediction problems. This is also referred to as multi-task learning [3, 5]. These classification tasks are generated from unlabelled data and the assumption is made that if the structures that are present in the unlabelled data are useful and has good discriminatory prediction ability, then it will also be useful for labelled data.

3 A model for the learning of structures

The algorithm that is selected makes use of a linear formulation of structural learning. A standard linear model is proposed and is then extended for structural learning.

3.1 Standard linear model

In the standard formulation of supervised learning a classifier is sought that, given the input vector $\bar{x} \in X$, will produce the corresponding output $y \in Y$; that is

$$f(\bar{x}) = \bar{w}^T \bar{x}, \tag{1}$$

where $f(\bar{x})$ is the classifier with weight vector $\bar{w}$ and feature vector $\bar{x}$. Training data consisting of a finite set of training examples $(\bar{X}_i, Y_i)$ is usually used to find an approximation $\hat{f}(\bar{x}) \in F$ for $f(\bar{x})$. *Empirical Risk Minimisation* (ERM) is a general method for the accurate approximation of $\hat{f}$ in which a weight vector is deduced from the training data that will minimise the prediction error on the labelled training data [7, 8].

3.2 Linear model for structural learning

The assumption with the linear model for structural learning is the existence of a low dimensional prediction structure that is shared by multiple prediction problems. Let l be one of m classification problems, $l \in \{1, \ldots, m\}$. The classifier for classification problem l is then given by

$$f_l(\Theta, \bar{x}) = \bar{w}_l^T \bar{x} + \bar{v}_l^T \Theta \bar{x}, \quad \Theta\Theta^T = I. \tag{2}$$

The standard linear model in §3.1 has a single component, $\bar{w}_l^T \bar{x}$, as input for each classification problem. With the linear model for structural learning a second component, $\bar{v}_l^T \Theta \bar{x}$, that is task specific for the classification problem l is added. In (2) I is the identity matrix, the orthogonal matrix Θ is the communal structural parameter that is shared by all the classification problems, and $\bar{w}_l$ and $\bar{v}_l$ are the specific weight vectors for each classification problem l.

The premise of the ERM model is to deduce a communal low dimensional prediction structure that is parameterised by the projection matrix Θ. The aim is to find a projection matrix and weight vectors that minimise the sum of the empirical risk over all the classification problems. Θ is the low dimensional projection matrix that is shared by all the classification problems $l \in \{1, \ldots, m\}$ and is a projection of the initial high dimensional space to a low dimensional space with the aim of simplifying the training of the classifier. Given a loss function L, the optimisation is

$$[\hat{\Theta}, \{\hat{f}_l\}] = \operatorname{arg[]} \min_{[\Theta, \{f_l\}]} \sum_{l=1}^{m} \frac{1}{n_l} \sum_{l=1}^{n_l} L(f_l(\bar{x}_i^l), y_i^l), \quad i \in \{1, \ldots, n_l\}. \tag{3}$$

where n_l is the number of feature types in classification problem l.

3.3 Alternating structure optimisation

The minimisation of equation (3) is done using *Alternating Structure Optimisation* (ASO), which is a two-phase iterative process:

1. Independently train m classifiers by keeping Θ constant and finding the optimal classifier $(\bar{w}_l, \bar{v}_l)$ for each problem l.
2. Find the corresponding predictive structures in the m classifiers by keeping $\bar{w}_l$ and $\bar{v}_l$ constant and determine Θ using *Singular Value Decomposition* (SVD).

During the first phase the classification tasks are independent from each other and during the second phase all the classifiers are coupled through Θ. Dimensional reduction is obtained through SVD and is done on a set of predictors, which is called the prediction space. This is made possible as multiple predictors are observed in multiple learning tasks. In the case where the observed predictors are seen as sample points of the predictor distribution in the predictor space, the SVD-based ASO (SVD-ASO) can be interpreted as an algorithm that finds the principal components of these predictors. Therefore, the algorithm searches for low dimensional structures with the highest prediction ability. The result is that the parameter values converge to local minima as the target value of the optimisation algorithm decreases with each iteration.

4 Methodology

In this paper, the use of a structural learning algorithm for semi-supervised learning is proposed. This is done by creating multiple classification tasks or auxiliary problems with labels from unlabelled data.

The data used consist of:

- 30 000 labelled tokens provided by the *Centre for Text Technology* (CTexT®) of the North-West University. These tokens were randomly split up into 80% training data, 10% test data and 10% tuning data. The tuning data are used to prevent overfitting.
- 158 450 290 unlabelled tokens, from the newspapers *Die Burger* and *Beeld*, that were provided by the media company Media24®.

Feature vectors are generated from both the auxiliary problems as well as the unlabelled data. In this study, only the features *Token*, *Previous token* and *Next token* were extracted and presented in the feature vector. For practicality purposes only the tokens in the labelled data as well as the 1 000 most frequent words in the unlabelled data were used to generate the feature vectors.

The semi-supervised procedure used consisted of the following steps:

1. **Create auxiliary problems.** Create m auxiliary problems and assign intermediate tags to the unlabelled data. Auxiliary problems with labelled training data are created automatically from unlabelled data [1, 2] and have to conform to two criteria:

   - There must be similarities between the structures of the target problem and the auxiliary problems.
   - Labelling must take place automatically.

   The main purpose of auxiliary problems is the inference of the prediction structure Θ, from unlabelled data, which is ultimately used in the target problem. With auxiliary problems it is possible to expose useful features, from unlabelled data, that are not necessarily present in the labelled data. Both the unsupervised and semi-supervised strategies are used in the creation of auxiliary problems [1, 2].

   In the unsupervised strategy, some of the substructures of the unlabelled data are used as intermediate tags, while other parts of the unlabelled data are used to classify these labels. In this study (as in Ando & Zhang [1, 2]) only the 1 000 most frequent tokens in the unlabelled data are used for the creation of auxiliary problems. Auxiliary problems are created for each token position by using *Token* as intermediate tag, and *Previous token* and *Next token* as features. In this way binary classification cases are created from unlabelled data by hiding certain available features for the purpose of the auxiliary problem. From these classification cases auxiliary problems are generated that are used in the generation of the structure matrix. The unsupervised strategy can be summarised by the following steps:

   (i) Extract rows consisting of *Token*, *Previous token* and *Next token* from the unlabelled data;
   (ii) create an auxiliary problem for each of these rows by assigning *Token* as a intermediate tag and using *Previous token* and *Next token* as features;
   (iii) use the features *Previous token* and *Next token* to determine Θ from the auxiliary problems; and
   (iv) train the final classifier using Θ and the features *Token*, *Previous token* and *Next token*.

The semi-supervised strategy has its foundation in the principle of co-training, initially proposed by Blum & Mitchell [4]; however, it is only used in the generation of auxiliary problems and not during the training of the final classifier. An actual part of speech tag is now assigned as an intermediate tag instead of a token. The first step of this strategy is as follows:

(i) Train Classifier A using labelled training data. Only use the feature *Token.*

After this, the strategy continues with steps (ii), (iii) and (iv) of the unsupervised strategy.

2. **Generating the structure matrix.** Determine Θ — the shared structure — by means of shared SVD by using auxiliary problems extracted from unlabelled data. This entails a joint ERM over all the auxiliary problems. The implementation of the algorithm is based on Ando & Zhang [1, 2] and the following parameters are used:

h — The dimension h determines the number of rows in Θ. A larger h will imply less reduction, in other words, a higher dimensional problem.

λ — The regularisation parameter λ is to prevent the model from becoming too complex.

η — The learning rate for *Stochastic Gradient Descent* (SGD).

The output of the process is the matrix Θ with h rows and the same number of columns as the length of the feature vector. The purpose of Θ is to reduce a high dimensional problem to a low dimensional problem with the most important information retained as well as the omission of the information that does not contribute to the accuracy of the classifier.

The algorithm is a two-phase iterative process based on SVD-ASO. In the first phase, m classifiers are trained independently using SGD [9]. An SGD algorithm evaluates the current data point and updates the weight vector accordingly. SGD is effective in applications with many data points and a large dimensional problem. The optimal result can be achieved with a small number of iterations, controlled by η, over the training data. The loss function used for the SGD method is the adapted Huber loss function

$$L(p, y) = \begin{cases} \max(0, 1 - py)^2 & \text{if } py \geq -1, \\ -4py & \text{otherwise.} \end{cases} \tag{4}$$

for predictor p and $y \in \{\pm 1\}$. The SGD method moves in the direction of the steepest descent; therefore, the derivative of $L(p, y)$ to p is needed:

$$\frac{dL(p, y)}{dp} = \begin{cases} 0 & \text{if } py \geq 1, \\ -4py & \text{if } py < -1, \\ -2y(1 - py) & \text{otherwise.} \end{cases} \tag{5}$$

3. **Training the classifier.** During the training of the classifier the focus is moved from the unlabelled data to the labelled data, while Θ represents the useful information that has been extracted from the unlabelled data. A predictor for the target problem,

viz. the assignment of a part of speech tag, is trained by applying ERM on the original labelled data. Now Θ is kept constant while the optimisation is done in respect of $\bar{w}$ and $\bar{v}$.

The predictor is a standard linear classification problem in which additional features are added to $\Theta\bar{x}$ as in

$$f(\Theta, \bar{x}) = \bar{w}^T\bar{x} + \bar{v}^T\Theta\bar{x}. \tag{6}$$

The classifier iterates through all possible labels for each token to which a label has to be assigned and assigns a value to each label using equation (3). The label with the highest value is the one that is eventually assigned to the token.

5 Results

The SVD-ASO algorithm was first only applied to labelled data to form the supervised learning baseline from which the error-rate reductions for the various semi-supervised learning variations were calculated. The variations entailed different combinations of the types of auxiliary problems used and the types of features considered. Auxiliary problems were extracted from unlabelled data by the unsupervised strategy (**Aux1**) and the semi-supervised strategy (**Aux2**), after which a structure matrix Θ was calculated for each of the problem solving strategies. For the training of the supervised learning classifier, the structure matrix was set to $\Theta = 0$. For the training of the semi-supervised learning classifier, **Aux1** & **Aux2**, two structure matrices were used. The features that were taken into account were also alternated between: *Token*, *Previous token* and *Next token* (**x1**); *Previous token* and *Next token* (**x2**); and *Token* (**x3**). The latter (**x3**) was only used for the training of a supervised learning classifier during the generation of the **Aux2** auxiliary problems. Note that for the training and eventual classification in which **x3** is applicable, only *Previous token* and *Next token* were used with Θ, since only these features were used in the generation of Θ. *Token* was only used as a feature in the supervised training phase. The accuracy of the different classifiers (%), as well as the error rate reduction with respect to the supervised learning baseline (Δ_{Err}), is shown in Table 1.

SVD-ASO Results	**x1**		**x2**		**x3**
	%	Δ_{Err}	%	Δ_{Err}	%
Supervised learning	82.77%	-	37.80%	-	78.05%
Semi-Supervised learning **Aux1**	82.91%	0.78%	38.84%	1.67%	78.05%
Semi-Supervised learning **Aux2**	82.98%	1.17%	-	-	-
Semi-Supervised learning **Aux1** & **Aux2**	82.84%	0.39%	-	-	-

Table 1: *Tagging accuracy and error rate reduction.*

All variations of the semi-supervised learning classifier show an error rate reduction with respect to the supervised learning baseline. The classifier with **x2** features and **Aux1** that generates auxiliary problems using the unsupervised strategy has the highest error rate reduction of 1.67%. For classifiers using **x1** features, the performance of **Aux2**, using the semi-supervised strategy to generate auxiliary problems, was better with an error rate reduction of 1.17%. Combining auxiliary problems with **Aux1** and **Aux2** results in an error rate reduction of 0.39%. However, this is not — as might be expected — a larger error rate reduction than when **Aux1** and **Aux2** are used separately.

The statistical significance of the difference between the supervised learning baseline and the various semi-supervised learning variations discussed above, is determined using McNemar test [6]. Both classifiers were trained with the same training data and then tested with the same test data T. The results of each classification case in T that were classified with Classifier A and Classifier B were tallied according to the confusion matrix

Number of cases classified incorrectly by both Classifier A and Classifier B	Number of cases classified incorrectly by Classifier A but correctly by Classifier B
Number of cases classified correctly by Classifier A but incorrectly by Classifier B	Number of cases classified correctly by both Classifier A and Classifier B

with the notation

$$\begin{array}{|c|c|}\hline n_{00} & n_{01} \\ \hline n_{10} & n_{11} \\ \hline \end{array}\ .$$

The total number of classification cases in T is

$$n = n_{00} + n_{00} + n_{00} + n_{00}. \tag{7}$$

The McNemar statistic is then calculated as

$$\frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}}\ . \tag{8}$$

The McNemar statistic's distribution is approximately χ^2 with 1 degree of freedom. Therefore, 1 minus the value of the cumulative density function gives the probability that the error rates of Classifier A and Classifier B are the same (indicated as $\rho(A = B)$ in Table 2).

Table 2 lists the controlled learning baseline classifier (Classifier A) with the semi-supervised classifier for which the statistical significance is to be determined (Classifier B).

Classifier A (supervised)	Classifier B (semi-supervised)	n_{00}	n_{10}	n_{01}	n_{11}	McNemar statistic	$\rho(A = B)$
x1	**Aux1** & **x1**	507	3	7	2467	0.90	34.28%
x2	**Aux1** & **x2**	1822	3	34	1125	24.32	0.00%
x1	**Aux2** & **x1**	506	3	2	2468	2.50	11.38%
x1	**Aux(1** & **2)** & **x1**	504	8	10	2984	0.06	81.37%

Table 2: *Difference between supervised and semi-supervised learning,* $n = 2984$.

For all practical purposes the probability that Classifier B is the same as Classifier A is 0% for **Aux1** & **x2**; therefore, the difference between the two classifiers is statistically significant. It can, therefore, be said with great confidence that, in this case, the auxiliary problems generated — with the unsupervised strategy — from the unlabelled data have definitely made a positive difference.

There is no statistically significant difference for the other variations of the semi-supervised classifiers. The probability that Classifier B is the same as Classifier A is 11.38% for **Aux2** & **x1**. This increases to 34.28% for **Aux1** & **x1** and 81.37% for **Aux(1** & **2)** & **x2**.

6 Conclusion

In this paper, structural learning was suggested using a model for the learning of structures. An *Alternating Structure Optimisation* (ASO) algorithm is proposed that applies structural learning to the problem of semi-supervised learning for classification. Dimensional reduction occurs through *Singular Value Decomposition* (SVD) which finds the most similar prediction structures. The experimental setup and implementation of the SVD-ASO algorithm is summarised as follows:

- Auxiliary problems are extracted from unlabelled data through an uncontrolled or semi-supervised strategy;
- a structure matrix is generated from the auxiliary problems; and
- the classifier is trained on labelled data and the structure matrix.

From the results it was shown that in some cases structural semi-supervised learning does give a better result compared to the supervised learning baseline. The limitations of the method can be ascribed to the use of only a small fraction of the available auxiliary problems for practicality purposes. For future work, better features, such as a 5-token sliding window, can be investigated as well as parallelising the learning algorithm to make use of more of the available auxiliary problems in the unlabelled data.

References

[1] Ando RK & Zhang T, 2005, *A framework for learning predictive structures from multiple tasks and unlabeled data*, J. Mach. Learn. Res., **6**, pp. 1817–1853.

[2] Ando RK & Zhang T, 2005, *A high-performance semi-supervised learning method for text chunking*, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Ann Arbor (MI), pp. 1–9.

[3] Argyriou A, Evgeniou T & Pontil M, 2006, *Multi-task feature learning*, Proceedings of the 19th International Conference on Neural Information Processing Systems, Vancouver, pp. 41–48.

[4] Blum A & Mitchell T, 1998, *Combining labeled and unlabeled data with co-training*, Proceedings of the 11th Annual Conference on Computational Learning Theory, Madison (WI), pp. 92–100.

[5] Caruana R, 1997, *Multitask learning*, Machine Learning, **28(1)**, pp. 41–75.

[6] Dietterich TG, 1998, *Approximate statistical tests for comparing supervised classification learning algorithms*, Neural Computation, **10(7)**, pp. 1895–1923.

[7] Escudero G, Màrquez L & Rigau G, 2000, *Boosting applied to word sense disambiguation*, Proceedings of the 11th European Conference on Machine Learning, Barcelona, pp. 129–141.

[8] Vapnik V, 2000, *The nature of statistical learning theory*, Springer, New York (NY).

[9] Zhang T, 2004, *Solving large scale linear prediction problems using stochastic gradient descent algorithms*, Proceedings of the 21st International Conference on Machine Learning, Banff, pp. 919–926.