– Proceedings –

42nd Annual Conference of the Operations Research Society of South Africa

15–18 September 2013
Protea Hotel, Technopark, Stellenbosch, South Africa

ISBN: xxx-xxxxx-xxxx-x

Proceedings of the 42^{nd} Annual Conference of the Operations Research Society of South Africa

ISBN: xxx-xxxxx-xxxx-x

Editorial Board

Review process

Thirty seven (37) manuscripts were submitted for possible inclusion in the *Proceedings of the 42^{nd} Annual Conference of the Operations Research Society of South Africa, 2012*. All submitted papers were double-blind peer-reviewed by at least two independent reviewers. Papers were reviewed according follow criteria: technical quality, i.e. correct use of language, clarity of expression, quality and justification of arguments; and on the contribution to Operations Research, i.e. knowledge of field, quality and consistency of referencing, significance of contribution and suitability for conference proceedings. The full set of criteria can be found at the end of the proceedings.

Of the thirty seven (37) submitted papers, fourteen (14) were ultimately, after consideration and incorporation of reviewer comments, judged to be suitable for inclusion in the proceedings of the conference. The proceedings is published online at
`http://www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf`.

Reviewers

The editorial would like to thank the following reviewers:

A Adewumi	University of Kwazulu Natal, South Africa
A Rademeyer	OPSI Systems, South Africa
A De Villiers	Stellenbosch University, South Africa
D Evans	Development bank of Southern Africa, South Africa
D Lubinsky	OPSI Systems, South Africa
E Willemse	University of South Africa, South Africa
H Ittmann	HWI Consulting, South Africa
H Kruger	North-West University, South Africa
I Durbach	University of Cape Town, South Africa
M Fisher	Sasol, South Africa
N Pillay	University of Kwazulu Natal, South Africa
R Koen	Council for Scientific and Industrial Research, South Africa
R Bennetto	OPSI Systems, South Africa
S Das	Council for Scientific and Industrial Research, South Africa
T Lane-Visser	Stellenbosch University, South Africa

Elias J Willemse
(e) ejwillemse@gmail.com
(t) +27 71 890 2714
Editor-in-Chief: ORSSA 2013 Proceedings

Table of contents

Proceedings of the 2013 ORSSA Annual Conference
pp. 1–10
www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf

ORSSA Proceedings
ISBN
XXX-XXXXX-XXXX-X

An algorithm for $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination of a graph

A Roux* & JH van Vuuren†

Abstract

Consider a network facility location problem that is modelled by a graph on n vertices labelled $v_1, \ldots, v_n$. Let $\boldsymbol{r} = [r_1, \ldots, r_n]$ and $\boldsymbol{s} = [s_1, \ldots, s_n]$ be vectors of non-negative integers and consider the problem of locating the minimum number of units of some commodity on the vertices of the graph such that at least s_i units are located in the vicinity (*i.e.* closed neighbourhood) of v_i, with no more than r_i units placed at the vertex v_i itself, for all $i = 1, \ldots, n$. The smallest number of units that may be placed on the vertices of the graph satisfying the above requirements is called the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$*-domination number* of the graph. In this paper we present an exponential-time algorithm which requires polynomial space for determining the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination number of an arbitrary graph.

In the proposed algorithm the problem of finding the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination number of a graph is modelled by the *set multicover with multiplicity constraints* (SMCM) problem. The only known exact algorithm for solving the SMCM problem runs in exponential time and requires exponential space. Its excessive memory usage renders the existing algorithm impractical and inspired us to pursue the more practical algorithm put forward in this paper.

The proposed algorithm follows a branch-and-reduce approach similar to the approach adopted for determining the classical domination number of a graph (*i.e.* in the case where $\boldsymbol{r} = [1, \ldots, 1]$ and $\boldsymbol{s} = [1, \ldots, 1]$) and employs two reduction rules that improve the running time of the algorithm considerably.

1 Introduction

Let $G = (V, E)$ be a simple graph of order n with vertex set $V = \{v_1, \ldots, v_n\}$ and define the *open neighbourhood* $N(v_i)$ of a vertex $v_i \in V$ as the set $\{v_j : \ v_i v_j \in E\}$ and its *closed neighbourhood* $N[v_i]$ as the set $N(v_i) \cup \{v_i\}$. A set $S \subseteq V$ is a *dominating set* of G if every vertex in V is either in S or adjacent to a vertex in S. The *domination number* $\gamma(G)$ is the minimum cardinality of a dominating set of G.

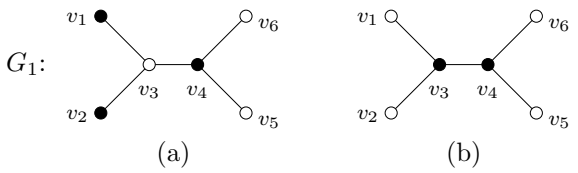


Figure 1: *(a) The solid vertices denote a dominating set of G_1. (b) The solid vertices denote a dominating set of G_1 of minimum cardinality, showing that $\gamma(G_1) = 2$.*

*Corresponding author: Department of Logistics, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: rianaroux@sun.ac.za

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

Consider the graph G_1 in Figure 1(a). The open neighbourhood of the vertex v_3 is the set of all vertices of G adjacent to v_3, that is $N(v_3) = \{v_1, v_2, v_4\}$, while its closed neighbourhood is $N[v_3] = N(v_3) \cup \{v_3\} = \{v_1, v_2, v_3, v_4\}$. The solid vertices in Figure 1(a) form a dominating set of G_1 since every vertex of G_1 is either solid or adjacent to a solid vertex. However, this is not a dominating set of minimum cardinality. As shown in Figure 1(b), the graph G_1 can be dominated by two vertices. Since the graph clearly cannot be dominated by only one vertex, $\gamma(G_1) = 2$.

Recently Cockayne [3] introduced a more general framework for domination in graphs. Let $\boldsymbol{r}$ and $\boldsymbol{s}$ be vectors of non-negative integers, called the *multiplicity constraint vector* and the *coverage requirement vector*, respectively. As in [3], we define an *$\boldsymbol{r}$-function* of G as a function $f : V \mapsto \mathbb{N}_0$ satisfying $f(v_i) \leq r_i$ for all $i = 1, \ldots, n$. For every $v \in V$, let $f[v] = \sum_{u \in N[v]} f(u)$. An $\boldsymbol{r}$-function f is called *$\boldsymbol{s}$-dominating* if $f[v_i] \geq s_i$ for each $i = 1, \ldots, n$. The *weight* of an $\boldsymbol{r}$-function f is defined as $|f| = \sum_{v \in V} f(v)$. The smallest weight of an $\boldsymbol{s}$-dominating $\boldsymbol{r}$-function of G is called the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination number of G and is denoted by $\gamma^{\boldsymbol{s}}_{\boldsymbol{r}}(G)$.

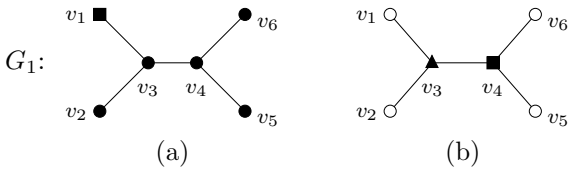


Figure 2: *Consider the graph G_1 and vectors $\boldsymbol{r} = [2, 1, 3, 2, 1, 1]$ and $\boldsymbol{s} = [3, 2, 4, 3, 2, 1]$. An $\boldsymbol{s}$-dominating $\boldsymbol{r}$-function of weight 7 is depicted in (a), where one unit of the commodity is placed on each solid round vertex and two units of the commodity are placed on the square vertex, while an $\boldsymbol{s}$-dominating $\boldsymbol{r}$-function of minimum weight is shown in (b), where two units of the commodity are placed at the square vertex and three units of the commodity are placed at the triangular vertex.*

Consider the graph G_1 in Figure 2 together with the vectors $\boldsymbol{r} = [2, 1, 3, 2, 1, 1]$ and $\boldsymbol{s} = [3, 2, 4, 3, 2, 1]$. Two $\boldsymbol{s}$-dominating $\boldsymbol{r}$-functions are shown in Figure 2. The function in (a) has a weight of 7 while the function in (b) has a weight of 5. The function in Figure 2(b) has the smallest possible weight of an $\boldsymbol{s}$-dominating $\boldsymbol{r}$-function of G_1 and therefore $\gamma^{\boldsymbol{s}}_{\boldsymbol{r}}(G_1) = 5$.

As noted in [3], an $\boldsymbol{s}$-dominating $\boldsymbol{r}$-function of G exists if and only if $\sum_{v_j \in N[v_i]} r_j \geq s_i$ for all $i = 1, \ldots, n$. Note that if $\boldsymbol{r} = \boldsymbol{s} = [1, \ldots, 1]$, then the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination number of G is its (classical) domination number, $\gamma(G)$.

After outlining a few applications of $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination in §2, we introduce a number of preliminary definitions and notations related to the celebrated set cover problem in §3, of which the domination problem is a special case, after which a branch-and-reduce algorithm for determining the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination number of an arbitrary graph is presented in §4. Two reduction rules are suggested in §5 to speed up the algorithm and the paper closes in §6 with two ideas for related future work.

2 Applications of $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination

The problem of $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination has numerous and widespread application in, among others, facility location and covering integer programs. Consider, as an example, the centre selection problem [7, 15] in which a network of candidate locations is modelled by a graph G. Centres must be placed in the network such that each vertex of G has a centre or is adjacent to at least one centre. As a fault tolerance requirement, it is required that every vertex should still have at least one functioning centre in its neighbourhood even after centres at b nodes have failed. To meet this requirement, each neighbourhood must contain at least $b + 1$ centres. The

$\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination problem models the situation where the objective is to place the minimum number of centres in the network.

Another example concerns file distribution [13] in a network of processors. The network of processors may be represented by a graph G. The $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination problem models the situation where memory devices are assigned to the vertices of G such that the sum of the memory in the neighbourhood of any vertex of G is at least of size k. The objective is to minimize the total size of memory in the network.

Other problems that can also be formulated as $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination problems include labour scheduling [6] and combinatorial problems that arise in the reverse engineering of protein and gene networks [1].

3 Preliminaries

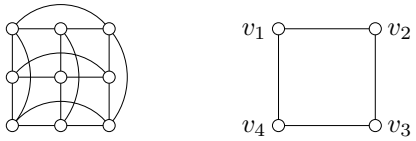
The domination problem was first formulated as a set cover problem by Kann [9] in 1992. This formulation has led to the currently fastest exact algorithm for computing the domination number of a graph in [17]. In the set cover problem, we are given a universe $\mathcal{U}$ of n elements and a family of sets $\mathcal{S}$, where each set $S \in \mathcal{S}$ is a subset of $\mathcal{U}$, and the objective is to find a sub-family $\mathcal{S}' \subseteq \mathcal{S}$ of minimum cardinality such that $\mathcal{U} = \bigcup_{S \in \mathcal{S}'} S$. A solution to the set cover problem also solves the (classical) domination problem for a graph G if the universe $\mathcal{U}$ is taken as the vertex set of G and the family of sets $\mathcal{S}$ is taken as the closed neighbourhoods of the vertices of G. The $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination problem, however, cannot be modelled by the set cover problem since vertices may have to be dominated more than once. Fortunately, the more general *set multicover with multiplicity constraints* (SMCM) problem [8] takes this factor into account.

In the SMCM problem, a universe $\mathcal{U}$ of n elements and a family of sets $\mathcal{S}$ are again specified, where each set $S \in \mathcal{S}$ is a subset of $\mathcal{U}$. The objective is to find a sub-family $\mathcal{S}' \subseteq \mathcal{S}$ of minimum cardinality such that each element $i \in \mathcal{U}$ is covered at least b_i times (*i.e.* appears in at least b_i elements of $\mathcal{S}'$), with the proviso that each set $S_j \in \mathcal{S}$ is used at most d_j times. The vector $\boldsymbol{b} = [b_1, \ldots, b_n]$ is called the *coverage requirement vector* and the vector $\boldsymbol{d} = [d_1, \ldots, d_n]$ is called the *multiplicity constraint vector*. As mentioned in [14], the SMCM problem may also be described as a set cover problem with a universe $\mathcal{U}$ containing b_i copies of the element i for all $i \in \{1, \ldots, n\}$ and a family of sets $\mathcal{S}$, where each set $S_j \in \mathcal{S}$ is a subset of the set $\{1, \ldots, n\}$ and appears d_j times in $\mathcal{S}$. Then the objective is to find a sub-family $\mathcal{S}' \subseteq \mathcal{S}$ that covers $\mathcal{U}$. This description of the SMCM problem is adopted in this paper.

To reduce the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination problem for a graph G to the SMCM problem, the universe $\mathcal{U}$ is taken as the vertex set of G and the family of sets $\mathcal{S}$ is taken as the closed neighbourhoods of the vertices of G. Furthermore, the coverage requirement vector is taken as $\boldsymbol{b} = \boldsymbol{s}$ and the multiplicity constraint vector is taken as $\boldsymbol{d} = \boldsymbol{r}$.

Research in designing heuristic and approximation algorithms for the SMCM and for covering integer programs, which may also be applied to SMCM problems, is widespread [5, 10, 11, 12]. However, the only known exact algorithm for solving the SMCM problem appears in [8] and solves the SMCM problem in $\mathcal{O}^*((b+1)^n)$ time[1] using $\mathcal{O}^*((b+1)^n)$ space, where $b = \max_i\{b_i\}$. This algorithm can be adopted for the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination problem and is therefore also the first exact algorithm for finding the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination number of a graph. We implemented the algorithm in [8] in Wolfram's `Mathematica 7` [19] and executed it on an Intel(R) Core(TM)2 Duo 3.16 GHz processor with 3.23 GB RAM running on `Microsoft Windows XP`. We found, however, that it is an impractical algorithm because of its excessive memory usage. For example, the algorithm failed to produce a result for the graph G_2 of order 9 in Figure 3 with $b = 6$ because of insufficient memory. This inspired us to pursue the algorithm put forward in this paper.

[1] The $\mathcal{O}^*$-notation is similar to the usual big $\mathcal{O}$-notation, but it suppresses all polynomial factors [18].



(a) G_2 (b) C_4

Figure 3: *The graph G_2 and the 4-cycle C_4.*

The notation related to multisets specified in [16] is adopted here. A *multiset* $\mathcal{A}$ over A is a pair $\langle A, f \rangle$, where A is a set and $f : A \mapsto \mathbb{N}_0$ is a function indicating the number of times the element $a \in A$ appears in the multiset $\mathcal{A}$. Let $\mathcal{A} = \langle A, f \rangle$ and $\mathcal{B} = \langle A, g \rangle$ be two multisets. The *sum* of two multisets $\mathcal{A}$ and $\mathcal{B}$, denoted by $A \uplus B$, is defined as the multiset $\mathcal{C} = \langle A, h \rangle$ where $h(a) = f(a) + g(a)$ for all $a \in A$. The *removal* of the multiset $\mathcal{B}$ from $\mathcal{A}$, denoted by $\mathcal{A} \ominus \mathcal{B}$, is defined as the multiset $\mathcal{C} = \langle A, h \rangle$ where $h(a) = \min\{(f(a) - g(a)), 0\}$ for all $a \in A$. Finally, if the multiset $\mathcal{B}$ is *deleted* from $\mathcal{A}$, denoted by $\mathcal{A} - \mathcal{B}$, the multiset $\mathcal{C} = \langle A, h \rangle$ is produced where $h(a) = 0$ for all $a \in A$ where $g(a) \neq 0$.

4 A branch-and-reduce algorithm for $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination

The branch-and-reduce algorithm proposed in this section for computing the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination number of an arbitrary graph was inspired by the work of Fomin *et al.* [4] and of Van Rooij and Bodlaender [17] who solved the problem of computing a minimum dominating set of a graph by reducing it to the set cover problem. Since the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination problem may reduced to the SMCM problem, a similar approach to that of [4, 17] may be used to solve the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination problem.

Our proposed algorithm is similar to the standard recursive algorithm for the set cover problem presented in [4, 17], except for an alteration to account for the use of multisets. The algorithm is called with a universe consisting of s_i copies of the vertex v_i for each $i \in \{1, \ldots, n\}$ and the multiset of sets $\mathcal{S}$ consisting of r_i copies of the neighbourhood of v_i for each $i \in \{1, \ldots, n\}$ as input.

Algorithm 1 takes a universe $\mathcal{U}$ and a multiset of sets $\mathcal{S}$ as input. If both $\mathcal{U}$ and $\mathcal{S}$ are empty, the algorithm returns the empty set, while if $\mathcal{S}$ is empty and $\mathcal{U}$ is not empty, the algorithm terminates. When $\mathcal{S}$ is not empty, the algorithm chooses an element $S \in \mathcal{S}$ of maximum cardinality and highest multiplicity in $\mathcal{S}$ and two subproblems are considered within a branching paradigm: one where S is part of the multicover and one where S is not part of the multicover. In the case where S is part of the multicover, the elements of S are removed from $\mathcal{U}$ (since $\mathcal{U}$ is a multiset this means that the multiplicities of the elements of S in $\mathcal{U}$ decrease by one) and S is removed from $\mathcal{S}$. Furthermore, for every $S' \in \mathcal{S}$, all the elements in S' not in the universe are removed from $\mathcal{S}$ as well. In the case where S is not part of the multicover, S is deleted from $\mathcal{S}$. The algorithm recursively solves both subproblems and returns the smallest multicover found by the recursive calls. The algorithm terminates if no set $S \in \mathcal{S}$ remains to branch on.

The working of Algorithm 1 is illustrated in Figure 4 which contains a part of the branching search tree constructed by the algorithm when computing the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination number $\gamma^{\boldsymbol{s}}_{\boldsymbol{r}}(C_4)$ of the cycle C_4 in Figure 3(b) for $\boldsymbol{r} = [1, 1, 2, 3]$ and $\boldsymbol{s} = [2, 1, 3, 2]$.

Algorithm 1 is called with the family of sets $\mathcal{S} = \{\{1, 2, 4\}, \{1, 2, 3\}, \{2, 3, 4\}, \{2, 3, 4\}, \{1, 3, 4\}, \{1, 3, 4\}, \{1, 3, 4\}\}$ and the universe $\mathcal{U} = \{1, 1, 2, 3, 3, 3, 4, 4\}$ as input. Since all the sets in $\mathcal{S}$ have cardinality 3, the algorithm branches on the set with the highest multiplicity, $S = \{1, 3, 4\}$. Two multisets, A_1 and A_2, are constructed. The multiset A_1 consists of the set $S = \{1, 3, 4\}$ and the output of Algorithm 1, when called with the family of sets $\mathcal{S}_1 = \{\{1, 2, 4\}, \{1, 2, 3\},$

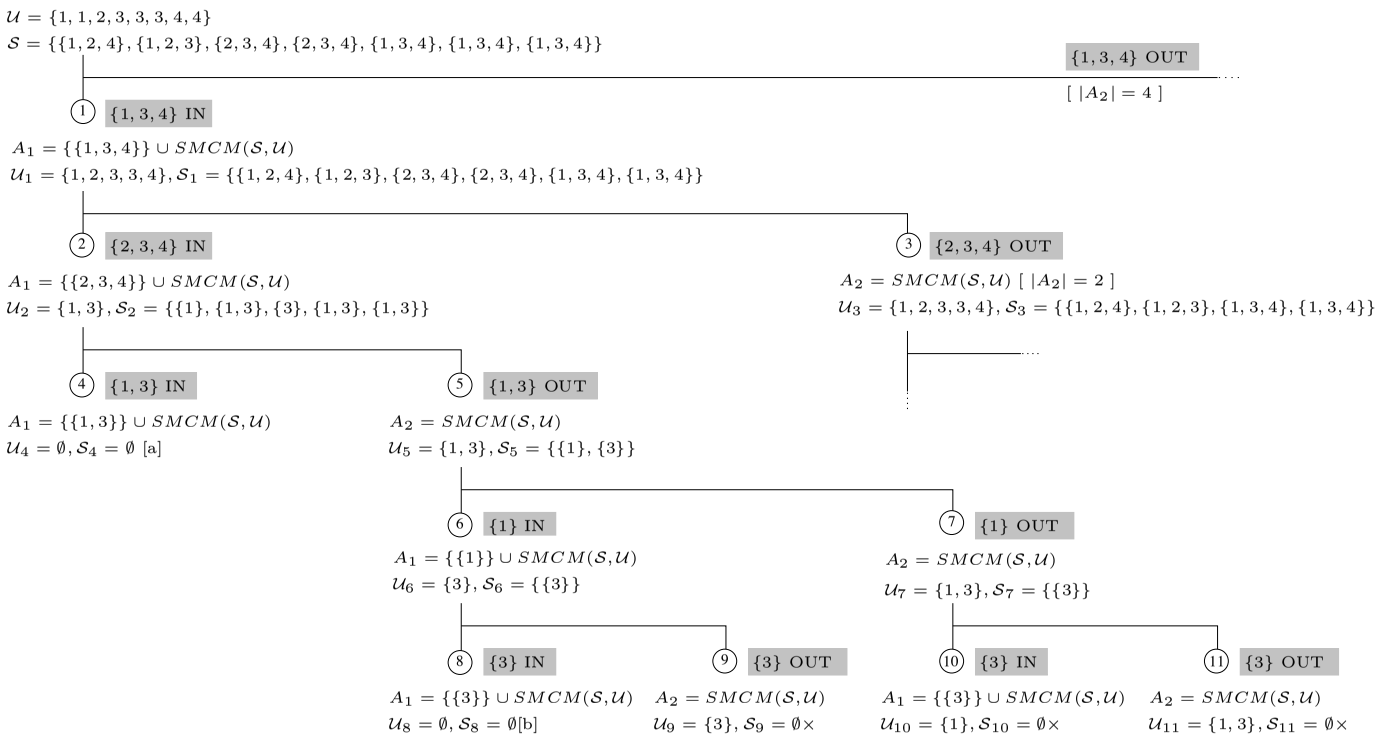


Figure 4: *Part of the branching search tree produced by Algorithm 1 when computing $\gamma_{\boldsymbol{r}}^{\boldsymbol{s}}(C_4)$ for the 4-cycle C_4 in Figure 3(b), where $\boldsymbol{r} = [1, 1, 2, 3]$ and $\boldsymbol{s} = [2, 1, 3, 2]$. The algorithm returns the minimum set multicover $\{\{1, 3, 4\}, \{2, 3, 4\}, \{1, 3\}\}$ which can be translated to either the $\boldsymbol{s}$-dominating $\boldsymbol{r}$-function $[0, 1, 1, 1]$ or $[0, 0, 1, 2]$ of weight $\gamma_{\boldsymbol{r}}^{\boldsymbol{s}}(C_4) = 3$. In the figure, [a] and [b] denote the end-vertices of the branches that produce multicovers for the universe, while $\times$ denotes branches that do not yield multicovers for the universe.*

Algorithm 1: SMCM($\mathcal{S}$,$\mathcal{U}$) A trivial set multicover with multiplicity constraint algorithm

Input : A set multicover instance ($\mathcal{S}$,$\mathcal{U}$).
Output: A minimum set multicover of ($\mathcal{S}$,$\mathcal{U}$).

```
if S = ∅ and U ≠ ∅ then
    return False
else
    if S = ∅ then
        return ∅
    else
        Let S ∈ S be a set of maximum cardinality and largest multiplicity in S
        A_1 = {S} ⊎ SMCM({S' ∩ (U ⊖ S) | S' ∈ S ⊖ {S}}, U ⊖ S)
        A_2 = SMCM(S − {S}, U)
return The smallest multiset from A_1 and A_2
```

$\{2,3,4\},\{2,3,4\},\{1,3,4\},\{1,3,4\}\}$ and the universe $\mathcal{U}_1 = \{1,2,3,3,4\}$ as input. The universe $\mathcal{U}_1$ is formed by deleting each element of S once from the original universe. Since $\mathcal{U}_1$ still contains all the vertices of the cycle C_4, only S is removed from $\mathcal{S}$ to form the new family of sets. The multiset A_2 is the output of Algorithm 1, when called with the family of sets from which all the copies of S have been removed, *i.e.* $\mathcal{S} = \{\{1,2,4\},\{1,2,3\},\{2,3,4\},\{2,3,4\}\}$ and the universe $\mathcal{U} = \{1,1,2,3,3,3,4,4\}$ as input. Figure 4 only depicts the branch of the search tree in which A_1 is constructed.

To construct A_1 at the node labelled 1, the algorithm is called with $\mathcal{S}_1 = \{\{1,2,4\},\{1,2,3\},\{2,3,4\},\{2,3,4\},\{1,3,4\},\{1,3,4\}\}$ and $\mathcal{U}_1 = \{1,2,3,3,4\}$ as input. Since $\{2,3,4\}$ and $\{1,3,4\}$ have the same cardinality and multiplicity in $\mathcal{S}$, either one can be chosen as S; in this example $S = \{2,3,4\}$. The algorithm now constructs the two multisets A_1 and A_2 as shown at the nodes labelled 2 and 3 in Figure 4. The multiset A_1 consists of the set $S = \{2,3,4\}$ and the output of Algorithm 1 when called with the universe $\mathcal{U}_2$ and the family of sets $\mathcal{S}_2$ as input, where $\mathcal{U}_2 = \{1,3\}$ is formed by deleting each element of S once from the universe $\mathcal{U}_1$ at node 1. The multiset $\mathcal{S}_2 = \{\{1\},\{1,3\},\{3\},\{1,3\},\{1,3\}\}$ is formed by removing one copy of S and deleting every element of $S' \in \mathcal{S}_1$ that is not in $\mathcal{U}_2$. With this call of the algorithm the multisets A_1 and A_2 at nodes 4 and 5 in Figure 4 are constructed with the choice of $S = \{1,3\}$.

At node 4 in Figure 4 the multiset A_1 consists of the set $S = \{1,3\}$ and the output of Algorithm 1 when called with both the universe and the family of sets equal to the empty set as input. The algorithm returns the empty set and produces the first multicover at the node labelled [a]. The multiset A_2 at node 5 is the output of Algorithm 1 when called with the universe $\mathcal{U}_5 = \{1,3\}$ and $\mathcal{S}_5 = \{\{1\},\{3\}\}$ as input, where the latter is formed by deleting all the copies of S from $\mathcal{S}_2$.

This branch leads to four leaves of the search tree (nodes 8 to 11) of which only one yields a multicover of the universe, the node labelled [b]. The leaves of the search tree labelled with the symbol '$\times$' indicate branches that do not yield multicovers of the universe.

To find a minimum multicover, consider the cardinalities of the multisets A_1 and A_2. The multiset with smallest cardinality is chosen at each branch. In Figure 4 it is shown that A_2 at node 3 has cardinality 2 and that A_2, where $\{1,3,4\}$ is not in the multicover, has cardinality 4. It is easy to see that the branch that ends at the node labelled [a] provides a minimum set multicover $\{\{1,3,4\},\{2,3,4\},\{1,3\}\}$. This multicover can be translated to one of the following $\boldsymbol{s}$-dominating $\boldsymbol{r}$-functions $[0,1,1,1]$ or $[0,0,1,2]$ (in vector form).

5 Improvement of the algorithm

To improve upon the efficiency of Algorithm 1 the following definitions are required. Let $(\mathcal{S},\mathcal{U})$ be an SMCM instance with coverage requirement vector $\boldsymbol{b}$ and multiplicity constraint vector $\boldsymbol{d}$. The entry b_i indicates the multiplicity of $i \in \{1,\ldots,n\}$ in $\mathcal{U}$. Furthermore, let $\mathcal{S}(i) = \{S \in \mathcal{S} \mid i \in S\}$ be the family of sets in $\mathcal{S}$ in which the element i occurs and define the multiplicity f_i to be the number of elements of $\mathcal{S}$ that contain i, so that $f_i = |\mathcal{S}(i)|$.

First consider the multiset A_2 at node 7 in Figure 4. It is immediately clear that this branch cannot form a multicover of $\mathcal{U}_7$ since there is no element 1 in $\mathcal{S}_7$, but there is an element 1 in the universe $\mathcal{U}_7$, in other words $b_1 > f_1$. In general it is clear that an SMCM instance cannot yield a multicover if there exists an element $i \in \mathcal{U}$ for which $b_i > f_i$. This observation inspires our first of two reduction rules.

Reduction Rule 1 (No multicover)

if *there exists an element* $i \in \mathcal{U}$ *for which* $b_i > f_i$ ***then***
  return ***false***
end if

Now consider the multiset A_2 at node 5 in Figure 4. The only way to cover $\mathcal{U}_5$ is by choosing both elements in $\mathcal{S}_5$. In general, if $b_i = f_i$ for some element $i \in \mathcal{U}$, then $\mathcal{S}(i)$ must be part of the multicover of $\mathcal{U}$. This observation inspires our second reduction rule.

Reduction Rule 2 (Unique elements)

if *there exists an element* $i \in \mathcal{U}$ *for which* $b_i = f_i$ ***then***
  return $\mathcal{S}(i) \uplus SMCM(\{S' \cap (\mathcal{U} \ominus \mathcal{S}(i)) | S' \in \mathcal{S} \ominus \mathcal{S}(i)\}, \mathcal{U} \ominus \mathcal{S}(i))$
end if

The formulation of Algorithm 1 can be improved since it is no longer necessary to test whether every computed candidate multicover in fact covers all of $\mathcal{U}$. Upon incorporation of the two reduction rules above, Algorithm 2 is obtained.

Algorithm 2: An improved version of Algorithm 1 incorporating Reduction Rules 1 and 2

Input : A set multicover instance $(\mathcal{S},\mathcal{U})$.
Output: A minimum set multicover of $(\mathcal{S},\mathcal{U})$.

$b_i \leftarrow$ the multiplicity of $i \in \{1,\ldots,n\}$ in $\mathcal{U}$
$f_i \leftarrow$ the multiplicity of $i \in \{1,\ldots,n\}$ in $\mathcal{S}$
if *there exists an element* $i \in \mathcal{U}$ *for which* $b_i > f_i$ **then**
  return False
if *there exists an element* $i \in \mathcal{U}$ *for which* $b_i = f_i$ **then**
  return $\mathcal{S}(i) \uplus SMCM(\{S' \cap (\mathcal{U} \ominus \mathcal{S}(i)) | S' \in \mathcal{S} \ominus \mathcal{S}(i)\}, \mathcal{U} \ominus \mathcal{S}(i))$
if $\mathcal{S} = \emptyset$ **then**
  return $\emptyset$
else
  Let $S \in \mathcal{S}$ be a set of maximum cardinality and largest multiplicity in $\mathcal{S}$
  $A_1 = \{S\} \uplus \text{SMCM}(\{S' \cap (\mathcal{U} \ominus S) | S' \in \mathcal{S} \ominus \{S\}\}, \mathcal{U} \ominus S)$
  $A_2 = \text{SMCM}(\mathcal{S} - \{S\}, \mathcal{U})$
return *The smallest multiset of* A_1 *and* A_2

We next estimate the space complexity and the worst-case time complexity of Algorithm 2.

Proposition 1 *If the input to Algorithm 2 is a graph G of order n, the coverage requirement vector $\boldsymbol{s} = [s_1, \ldots, s_n]$ and the multiplicity constraint vector $\boldsymbol{r} = [r_1, \ldots, r_n]$, then Algorithm 2 computes the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination number of G in $\mathcal{O}^*((r+1)^n)$ time using $\mathcal{O}(n^2+n)$ space, where $r = \max_i\{r_i\}$ and $s = \max_i\{s_i\}$.*

Proof. Algorithm 2 calls itself recursively at most $\prod_{i=1}^{n}(r_i+1)-1$ times, since the algorithm considers all possible combinations of sets in $\mathcal{S}$. The desired result follows since the remaining steps of the algorithm may be performed in polynomial time.

To determine the space complexity, first consider the input variables. The universe $\mathcal{U}$ may be stored as a vector of size n, where each entry indicates the multiplicity of the element in the universe. Furthermore, the multiset of sets $\mathcal{S}$ may be stored as a list of length n together with a vector, indicating the multiplicity of each element, of size n. Each element of $\mathcal{S}$ is of size at most n. The occupied space is reused in the subsequent calls to the algorithm in Steps 6, 11 and 12. The algorithm also stores two n-vectors $\boldsymbol{b}$ and $\boldsymbol{f} = [f_1, \ldots, f_n]$, while the sets A_1 and A_2 may each occupy at most the same amount of space as the multiset of sets $\mathcal{S}$. This results in a space complexity of $\mathcal{O}(n^2+n)$. ■

Since the inclusion of the two reduction rules is not reflected in the time complexity estimate of Proposition 1, the execution times of the algorithm before and after the inclusion of the reduction rules are compared in Table 1. The algorithm was implemented in `Mathematica 7` [19] and cycles of orders 6 to 15 were considered for the comparison. For each graph the algorithm was executed for five different pairs of $\boldsymbol{r}$ and $\boldsymbol{s}$ vectors. The vector $\boldsymbol{s}$ was taken as a randomly generated vector[2] with values between 1 and 6, while the vector $\boldsymbol{r}$ was taken as the first random $\boldsymbol{s}$-dominating vector with values between 1 and 6. Table 1 contains a summary of the average execution times and average number of calls of Algorithms 1 and 2 as well as the difference in and percentage decrease in the execution time and number of calls for the two algorithms. Both the execution times and number of calls decreased considerably upon the inclusion of the reduction rules. A percentage decrease of more than ninety nine percent in the execution time was thus recorded for cycles of order 12 and up.

		Algorithm 1		Algorithm 2		Percentage decrease	
Graph	Ave $\gamma_{\boldsymbol{r}}^{\boldsymbol{s}}$	Ave Time	Ave no of Calls	Ave Time	Ave no of Calls	Ave Time	Ave no of Calls
C_6	10.00	0.19	1 365	0.03	121	85.70%	91.11%
C_7	10.20	0.51	3 769	0.05	253	89.66%	93.29%
C_8	12.40	0.66	4 884	0.04	176	94.27%	96.39%
C_9	13.60	4.78	37 189	0.18	867	96.31%	97.67%
C_{10}	14.60	21.35	156 716	0.52	2 440	97.56%	98.44%
C_{11}	15.60	36.05	259 595	0.54	2 464	98.50%	99.05%
C_{12}	17.00	244.31	1 734 313	1.48	6 760	99.39%	99.61%
C_{13}	18.60	262.07	1 925 558	0.56	2 563	99.78%	99.87%
C_{14}	20.20	338.59	2 502 738	1.13	5 058	99.66%	99.80%
C_{15}	24.00	3 097.67	22 713 275	3.32	14 163	99.89%	99.94%

Table 1: *A comparison between the efficiencies of the SMCM algorithm with reduction rules and the algorithm without reduction rules. All times are measured in seconds.*

Algorithm 2 was able to compute the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination number of the graph G_2 of order 9 in Figure 3 with $b = 6$ with ease, unlike the algorithm of Hua *et al.* [8].

The algorithm was validated in five fundamentally different ways. First, we manually examined the output of the algorithm for a number of randomly generated graphs of varying order with different pairs of $\boldsymbol{r}$ and $\boldsymbol{s}$ vectors and satisfied ourselves that the multicover returned indeed covered the universe in each case. Secondly, we verified that the algorithm returned values that

[2]Drawn from a uniform distribution.

fall between the theoretical bounds given in [2]. The output of the algorithm was also compared with the output of the algorithm of Hua *et al.* [8].

In addition, we verified that the algorithm produced the same results when a set other than the one of maximum cardinality and largest multiplicity is used at the first branch. Finally, we satisfied ourselves that the algorithm produced the correct values for the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination number of graph classes for which the value of the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination number is known theoretically, such as cycles and complete graphs (when $\boldsymbol{r} = [r, \ldots, r]$ and $\boldsymbol{s} = [s, \ldots, s]$).

6 Conclusion

A branch-and-reduce algorithm was presented in this paper for computing the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination number of an arbitrary graph. The algorithm runs in exponential time and requires polynomial space. Its working was illustrated by means of an example in §4.

In [17] Van Rooij and Bodlaender used a measure-and-conquer analysis as a design tool to refine their algorithm for the set cover problem. Future work may include pursuit of a similar approach towards refining the branch-and-reduce algorithm in §4. Such a measure-and-conquer analysis may lead to a better estimation of the worst-case time complexity of the branch-and-reduce algorithm.

The algorithm in §4 was implemented in Mathematica. The execution-time of the algorithm may, however, be shortened dramatically if implemented in a low-level programming language such as C or C++.

References

[1] Berman P, DasGupta B & Sontag E, 2007, *Randomized approximation algorithms for set multicover problems with applications to reverse engineering of protein and gene networks*, Discrete Applied Mathematics, **155(6–7)**, pp. 733–749.

[2] Burger AP & Van Vuuren JH, 2008, *On the $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination number of a graph*, Ars Combinatoria, **88**, pp. 275–271.

[3] Cockayne EJ, 2009, *Towards a theory of $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$-domination in graphs*, Utilitas Mathematica, **80**, pp. 97–114.

[4] Fomin FV, Grandoni F & Kratsch D, 2009, *A measure & conquer approach for the analysis of exact algorithms*, Journal of the Association of Computing Machinery, **56(5)**, pp. 25:1–25:32.

[5] Hall NG & Hochbaum DS, 1986, *A fast approximation algorithm for the multicovering problem*, Discrete Applied Mathematics, **15(1)**, pp. 35–40.

[6] Hochbaum DS & Levin A, 2006, *Cyclical scheduling and multi-shift scheduling: Complexity and approximation algorithms*, Discrete Optimization, **3(4)**, pp. 327–340.

[7] Hochbaum DS & Shmoys DB, 1986, *A unified approach to approximation algorithms for bottleneck problems*, Journal of the Association for Computing Machinery, **33(3)**, pp. 533–550.

[8] Hua QS, Wang Y, Yu D & Lau FCM, 2010, *Dynamic programming based algorithms for set multicover and multiset multicover problems*, Theoretical Computer Science, **411(26–28)**, pp. 2467–2474.

[9] Kann V, 1992, *On the approximability of NP-complete optimization problems*, Doctoral Dissertation, Royal Institute of Technology, Stockholm.

[10] Kolliopoulos SG, 2003, *Approximating covering integer programs with multiplicity constraints*, Discrete Applied Mathematics, **129(2–3)**, pp. 461–473.

[11] Kolliopoulos SG & Young NE, 2005, *Approximation algorithms for covering/packing integer programs*, Journal of Computer and System Sciences, **71(4)**, pp. 495–505.

[12] Kulik A, Shachnai H, Shmueli O & Sayegh R, 2011, *Approximation schemes for deal splitting and covering integer programs with multiplicity constraints*, Theoretical Computer Science, **412(52)**, pp. 7087–7098.

[13] Naor M & Roth RM, 1991, *Optimal file sharing in distributed networks*, SIAM Journal on Computing, **24**, pp. 158–183.

[14] Nederlof J, 2008, *Inclusion exclusion for hard problems*, Doctoral Dissertation, Utrecht University, Utrecht.

[15] Peleg D, Schechtman G & Wool A, 1997, *Randomized approximation of bounded multicovering problems*, Algorithmica, **18(1)**, pp. 44–66.

[16] Syropoulos A, 2001, *Mathematics of multisets*, pp. 347–358 in Calude C, Paun G, Rozenberg G & Salomaa A (Eds), *Multiset Processing*, Volume 2235 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg.

[17] Van Rooij MM & Bodlaender HL, 2011, *Exact algorithms for dominating set*, Discrete Applied Mathematics, **159(17)**, pp. 2147–2164.

[18] Woeginger GJ, 2003, *Exact algorithms for NP-hard problems: A survey*, pp. 185–207 in Jünger M, Reinelt G & Rinaldi G (Eds), *Combinatorial optimization—Eureka, you shrink!*, Springer-Verlag, New York (NY).

[19] Wolfram Research, 2013, *Mathematica Edition: Version 9.0.1.0*, Wolfram Research, Inc., Champaign (IL).

Proceedings of the 2013 ORSSA Annual Conference
pp. 11–18
www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf

ORSSA Proceedings
ISBN
XXX-XXXXX-XXXX-X

An analytic testing framework for traffic control strategies

AP Burger*, MD Einhorn† & JH van Vuuren‡

Abstract

It is often desirable to form a preliminary idea about the potential effectiveness of a novel traffic control strategy before engaging in the time-consuming process of building a realistic traffic simulation model in which to test the control strategy extensively. In this paper, a simple analytic testing framework is suggested for this purpose. The framework may be used to ascertain the potential effectiveness of a traffic control strategy by comparing it to a known optimum control strategy in terms of its ability to minimise the average control delay of vehicles arriving at an isolated intersection. Three known self-organising traffic control strategies are implemented in this testing framework and compared to a known, optimum, pre-timed traffic control strategy for both undersaturated and oversaturated deterministic traffic conditions.

Key words: Traffic control, self-organisation, analytic models.

1 Introduction

When developing novel traffic control strategies for implementation at signalised intersections, a natural step is to test these novel traffic control strategies against previously proposed strategies and industry standards. It is common practice to perform these tests in a simulated environment [6]. There are many advantages to using simulation in this context such as its cost-effectiveness when compared to the implementation of an equivalent real-world model, as well as its ability to compress and expand time to allow for a detailed and thorough analysis of the system under investigation. However, in spite of these advantages it can be very time consuming to develop an accurate and reliable simulation model in which to implement these traffic control strategies. For this reason, we aim to provide a simple analytic testing framework in this paper according to which the potential effectiveness of traffic control strategies may be tested speedily and with relative ease under simplified, deterministic conditions, for which there is a known optimal pre-timed signal control strategy. Should the novel control strategy achieve the same, or desirably similar results to that of the optimum control strategy, further and more extensive testing of the strategy is warranted in a simulated environment under more realistic, stochastic conditions, for which there is no known optimal strategy. However, if the novel control strategy

*Department of Logistics, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: apburger@sun.ac.za

†Corresponding author: Department of Logistics, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: einhorn@sun.ac.za

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

is unable to achieve desirable results under the imposed simplified conditions it may be concluded that it will potentially not perform desirably under more complex and realistic stochastic conditions either and may therefore be disregarded in terms of future investigation without the need for testing it in a simulated environment. In this case, however, one should be able to identify fundamental short-comings of the control strategy and attempt to rectify them, where possible, and retest the amended version. The analytic testing framework presented in this paper may therefore be used to filter out undesirable traffic control strategies early during the analysis effort, before excessive amounts of time and effort are expended testing their efficiency by means of advanced stochastic experiments.

The analytic testing framework described above is implemented as a proof of concept, using it to determine the effectiveness of three well-known self-organising traffic control strategies by ascertaining their proficiency to minimise the average control delay experienced by vehicles arriving at a single, isolated signalised intersection. For all three approaches vehicle arrivals are assumed to be uniform, continuous and deterministic. The results obtained are compared to a known optimised, pre-timed signal control strategy for the prevailing conditions. Results are presented for undersaturated and oversaturated conditions. These results suggest that this is a suitable first-order procedure for testing the efficiency of novel self-organising traffic control strategies before deciding whether or not to implement them in a simulated environment.

2 A delay model from the literature

For each signalised intersection, the *phase plan* adopted, the *type* of signal control implemented and the allocation of green time among the various phases have to be considered [8]. The *phase plan* comprises the number of signalling phases to be implemented at the intersection as well as the order or sequence in which they are implemented. The type of control may be either *pre-timed* or *actuated* control. *Pre-timed* signal control involves a fixed sequence of phases which are repeated continually, the green times of which are fixed, resulting in a constant cycle length. *Actuated* vehicle control typically involves some sort of vehicle detection mechanism which is used to determine the amount of green time allocated to an approach. In this paper, the simplest possible analysis scenario is assumed, namely that of deterministic and uniform vehicle arrivals. Therefore, although the self-organising traffic control strategies are a form of actuated control, they appear to be pre-timed under the assumption mentioned above.

The terminology used to describe vehicle delays at a signalised intersection are defined in Table 1. The delay model presented in [8] for an isolated intersection is given by

$$d_i = d_{1,i} + d_{2,i} + d_{3,i}. \tag{1}$$

Here, d_i is the average control delay experienced by vehicles along approach i to a signalised intersection. Furthermore, $d_{1,i}$ is the *average uniform delay* which accounts for the deterministic delay experienced under the assumption of a uniform arrival rate, while $d_{2,i}$ is the *average incremental delay* and accounts for the delay due the presence of oversaturation queues (which occur when the demand for service along an approach exceeds the capacity) and for the effect of random arrivals (should a deterministic arrival rate not be assumed). Moreover, $d_{3,i}$ is the *average initial queue delay* and accounts for the delay due to an initial queue being present along approach i during the analysis period.

In (1), the average uniform delay is calculated as

$$d_{1,i} = \frac{\frac{1}{2}C(1 - g_i/C)^2}{1 - (\min\{1, X_i\}g_i/C)}, \tag{2}$$

Symbol	Description	Unit
q_i	The *vehicle flow rate* along approach i. This is the rate at which vehicles enter approach i.	veh/s
s_i	The *saturation flow rate* of approach i. This is the maximum number of vehicles per second which may travel at the speed limit along approach i. It is assumed that this is the rate at which queued vehicles depart an intersection upon the commencement of a green signal.	veh/s
C	The *total cycle length*. This is the time required to complete one full cycle of all phases in the phase plan.	s
τ	The length of the amber and all-red phase, also known as the *setup time* or *inter-green time*. This is the length of time during which no vehicles may pass through the intersection while service is changed from one approach to another and is adjusted to incorporate all time losses due to finite accelerations.	s
T	The length of the *analysis period*. The analysis period therefore consists of T/C cycles.	s
g_i	The *effective green time* afforded to approach i. This is the length of time during which vehicles along approach i proceed through the intersection.	s
c_i	The *capacity* of approach i. This is the number of vehicles that are able to pass through the intersection at the rate of s_i during the effective green time g_i per cycle C and is calculated as $s_i g_i / C$.	veh/s
X_i	The *volume-to-capacity-ratio* of approach i. This is a measure of the degree of saturation along approach i and is calculated as q_i/c_i, which is equivalent to $q_i C/s_i g_i$ (*i.e.* the total number of vehicles that require service along approach i per cycle, divided by the number of vehicles that may be served along approach i per cycle). If $X_i \leq 1$, the approach is said to be undersaturated. Otherwise, the approach is considered oversaturated.	None

Table 1: *Basic notation required to describe traffic dynamics at signalised intersections.*

with the meanings of the various symbols as defined in Table 1.

This expression for $d_{1,i}$ assumes uniform, undersaturated traffic flow conditions and for this reason, values of X_i greater than 1 are not considered.

The average incremental delay, on the other hand, is given by

$$d_{2,i} = \frac{T}{4}\left((X_i - 1) + \sqrt{(X_i - 1)^2 + \frac{8kIX_i}{c_i T}}\right). \tag{3}$$

Here, k is an incremental delay calibration constant that depends on the type of control implemented at the intersection (pre-timed or vehicle actuated) and assumes a random arrival process. Furthermore, I is an upstream metering adjustment factor which incorporates the effects of vehicle arrivals from upstream signalised intersections. Under the assumptions of deterministic uniform arrivals at an isolated intersection the second term under the root sign in (3) assumes the value of zero.

In the event that there is no queue present at the start of the analysis period along approach i, the value of $d_{3,i}$ is also zero. Throughout this paper it is assumed that there is no initial queue present at the start of any analysis period and, thus, for the sake of conciseness the expression presented for $d_{3,i}$ in [3] is not discussed here.

A diagrammatic representation of the delays $d_{1,i}$ and $d_{2,i}$ appears in Figure 1.

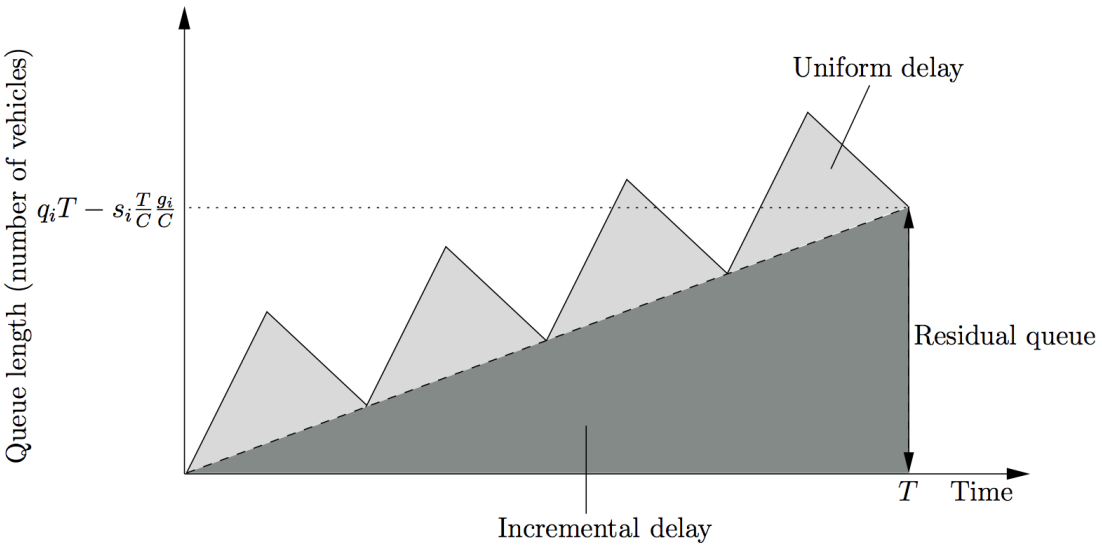


Figure 1: *Components of average delay over an analysis period of length T.*

With average control delay estimated for a single intersection approach, it is desirable to aggregate the delay along each approach so as to determine the average control delay for the intersection. This aggregation is achieved by calculating the weighted average delay of the approaches, weighted by the volumes of traffic along the approaches. The average control delay for an intersection I is therefore

$$d_I = \frac{\sum_i d_i q_i}{\sum_i q_i}. \tag{4}$$

3 Average control delay minimisation model

The following question arises for a signalised intersection with n signal phases for which no two approaches are served during the same phase: Which cycle length and corresponding green times, $g_1, \ldots, g_n$, minimise this expected average control delay? These decision variables may be found by minimising d_I in (4) subject to the constraint set

$$C = \sum_{i=1}^{n} g_i + n\tau, \quad i = 1, \ldots, n. \tag{5}$$

The above minimisation problem may be solved using a standard numerical solver. For this paper, the *Minimize* function in *Mathematica 8.0* [7] was used. The *Minimize* function has several different algorithms at its disposal which can be used, depending on the type of problem at hand [7]. The most generally used method is based on the *cylindrical algebraic decomposition* (CAD) algorithm [2], which is applied when the objective function and the constraints are real algebraic functions, as is the case above. Under these conditions the CAD algorithm is always able to compute global extrema.

4 Self-organising traffic control algorithms

This section contains an overview of the philosophies and the workings of three self-organising traffic control strategies found in the literature. The first two approaches were proposed by

Gershenson [4], who describes a self-organising system as one in which elements of the system interact with one another in order to achieve a global objective, without this objective being imposed by a single or several elements, nor determined hierarchically. The first method proposed by Gershenson, which is referred to here as *Gersh1* for the remainder of the paper, relies on a counting mechanism. For the case that approach i is currently not receiving service at time t, a variable $\kappa_i(t)$ keeps a count of all the vehicles present along approach i multiplied by the amount of time for which they have been present along the approach. When this value of $\kappa_i(t)$ reaches a predetermined threshold value, θ_i, service to approach i commences and $\kappa_i(t)$ is reset back to zero. This method allows for vehicle platoons to form along intersection approaches. If the value of θ_i is, however, chosen too large, then unnecessarily long waiting times may be experienced by vehicles along approach i under light traffic conditions, while under heavier traffic conditions, too frequent switching of service from one approach to another may occur if the value of θ_i is chosen too small (this may cause considerable vehicle queues to build up along the intersection approaches).

The second method proposed in [4], referred to here as *Gersh2*, differs from the first in that it introduces a minimum allowable green time, ρ_i^{min}, such that service to approach i must last at least ρ_i^{min} time steps. In other words, once approach i has received service for a time period of $\rho_i \geq \rho_i^{min}$, service of approach i ends when $\kappa_i(t) \geq \theta_i$.

The third self-organising traffic control method considered in this paper was presented by Lämmer and Helbing [5], and is referred to here as *LamHel*. This method relies on the anticipation of green times required to clear expected vehicle queues before they form and combines an *optimisation strategy* with a *stabilisation strategy*. Each approach (or group of approaches which is served during the same green period) to an intersection is assigned a dynamic priority index. These dynamic priority indices are compared during implementation of the optimisation strategy and the approach (or group of approaches) achieving the highest priority index receives service (*i.e.* a green signal). The priority index of approach i at time t is calculated as

$$\pi_i(t) = \frac{\hat{n}_i(t)}{\tau_i^{pen}(t) + \tau_i(t) + \hat{g}_i(t)}, \tag{6}$$

where $\hat{n}_i(t) = \hat{g}_i(t)s_i$ is the anticipated number of vehicles expected to be served at the saturation flow rate s_i upon commencement of the anticipated green time. This value of $\hat{n}_i(t)$ accounts for all the vehicles currently queued along approach i at time t as well as all the vehicles expected to become queued during the subsequent setup phase and green time phase [5]. Furthermore, $\tau_i^{pen}(t)$ is a penalty term associated with discontinuing service to approach i. If approach i is not receiving service at time t, then $\tau_i^{pen}(t) = 0$. Moreover, $\tau_i(t)$ is the remaining setup time required before the onset of service to approach i. Finally, $\hat{g}_i(t)$ is the green time required to clear all the vehicles arriving at the intersection along approach i during a time interval of length $t + \tau_i(t) + \hat{g}_i(t)$ and may be found by using standard bisection methods [1].

This prioritisation approach is complemented by a stabilisation strategy aimed at ensuring that vehicle queues do not grow excessively long and is achieved by introducing a *critical queue length* as well as a *maximum allowable green time*. Approach i is scheduled for service as soon as the anticipated number of vehicles requiring service along it at time t, namely $\hat{n}_i(t)$, exceeds the critical queue length. Approach i then continues to receive service either until the anticipated queue of vehicles has been dissipated completely or it has received a green signal for the maximum allowable green time. The critical queue length and maximum allowable green time are calculated using user-defined parameters S and S^{max} which ensure that each traffic flow shall be served once, on average, within a desired service interval of length $S > 0$ and that each traffic flow shall be served at least once within a maximum service interval of length $S^{max} \geq S$.

5 Algorithmic comparison results

The self-organising traffic control strategies presented in §4 were implemented in the same analytic environment as that of the average control delay minimisation model presented in §3 under the same assumption that vehicle arrivals are uniform, deterministic and continuous. The purpose of the tests was to investigate whether or not the self-organising control strategies were capable of achieving the optimal fixed-time cycle settings under varying degrees of saturation. The tests were conducted for a single, isolated intersection with two conflicting traffic flows along approach 1 and approach 2. The results are summarised in Table 2.

Inputs	$T = 900$, $\tau_i = 4$, $\tau_2 = 4$, $q_1 = 0.4$, $q_2 = 0.2$, $s_1 = 0.8$, $s_2 = 0.8$					
Control Strategy	$g_1(s)$	$g_2(s)$	X_1	X_2	$d_I(s)$	Parameter settings
Optimum	16	8	1	1	9.33	None
Gersh1	16	8	1	1	9.33	$\theta_1 = 57.6$, $\theta_2 = 80$
Gersh2	16	8	1	1	9.33	$\rho_1^{min} \leq 16$, $\rho_2^{min} \leq 8$
LamHel	16	8	1	1	9.33	$S \geq 32$

Table 2: *Low traffic demand scenario*

Inputs	$T = 900$, $\tau_i = 4$, $\tau_2 = 4$, $q_1 = 0.7$, $q_2 = 0.6$, $s_1 = 0.8$, $s_2 = 0.8$					
Control Strategy	$g_1(s)$	$g_2(s)$	X_1	X_2	$d_I(s)$	Parameter settings
Optimum	50.32	40.68	1.72	1.83	429.48	None
Gersh1	50.32	40.68	1.72	1.83	429.48	$\theta_1 = 1\,397.11$, $\theta_2 = 1\,770.24$
Gersh2	50.32	40.68	1.72	1.83	429.48	$\rho_1^{min} \leq 50.32$, $\rho_2^{min} \leq 40.68$
LamHel	84.49	38.25	1.35	2.56	506.17	$S = 370$

Table 3: *High traffic demand scenario*

It may be observed in Table 2 that for the first scenario, in which there is a relatively low traffic demand, all three self-organising traffic control strategies are able to obtain the optimal green time settings, provided that all the necessary user-specified parameters are chosen correctly. It may also be noted that the optimal green times found result in a degree of saturation of $X = 1$ for both approaches, indicating that it is optimal to provide just enough green time to clear the number of vehicles arriving during a cycle and not to provide excessive green time to either of the approaches. With the optimal green times known, it was possible to reverse engineer the values of θ_1 and θ_2 such that Gersh1 was able to achieve the optimum settings, where $\theta_1 = 57.6$ veh·s and $\theta_2 = 80$ veh·s. If these values were any smaller, the corresponding green times would not be long enough and if they were any larger, the green times would be too long. For Gersh2, the optimum was found provided that $\theta_1 = 57.6$ veh·s and $\theta_2 = 80$ veh·s and that the minimum allowable green times for each approach were at least as large as the green times found by the optimisation strategy, *i.e* $\rho_1^{min} \leq 16$ s and $\rho_2^{min} \leq 8$ s. For LamHel, the optimisation strategy was able to find the optimal green times provided that S was greater than 32 s, which is the optimal cycle length.

For the second scenario in Table 3, oversaturation along both approach 1 and 2 was unavoidable as indicated by both X_1 and X_2 values being greater than 1. Gersh1 was able to achieve the optimal green time settings provided for the choice of parameter values $\theta_1 = 1397.11$ veh·s and $\theta_2 = 1770.24$ veh·s. Both these threshold values are relatively high, however, and are associated with excessively long vehicle queues which may be too long to be accommodated physically, depending on the length of the intersection approaches. Gersh2 was able to achieve optimality provided that $\rho_1^{min} \leq 50.32$ s and $\rho_2^{min} \leq 40.68$ s and for parameter values satisfying $\theta_1 \leq 1397.11$ veh·s and $\theta_2 \leq 1770.24$ veh·s. LamHel was unable to achieve the optimum signal timing settings. The best it could do was set S to a value of 370 s, which resulted in a maximum green time for approach 1 of 84.49 s and for approach 2 of 38.25 s. This resulted in an average

control delay of 506.17 s per vehicle for the analysis period T, which is 17.8% larger than that achieved by the optimum signal timings (*i.e.* by solving the model in §3).

6 Conclusion

It is acknowledged that the true value and power of self-organising traffic control algorithms emerge under stochastic conditions with random, discrete vehicle arrivals due to the ability of self-organising traffic control algorithms to utilise free intersection capacity much more effectively than even an optimised pre-timed approach. However, a truly effective self-organising traffic control algorithm should at least be able to achieve the best possible signal timings under simplified deterministic conditions. For this reason, an analytic testing procedure was developed in this paper which allows for a proposed traffic control strategy to be implemented easily and quickly, and compared to a known optimal pre-timed control strategy. This allows one to determine whether or not it is likely to be worthwhile to proceed to testing the proposed strategy in a more advanced and realistic simulated environment.

Three well-known self-organising traffic control strategies were implemented in this analytic test facility and compared to a known optimum for both undersaturated and oversaturated traffic conditions. For the case of undersaturated conditions, all three algorithms were able to achieve the optimum signal timing settings. For oversaturated traffic conditions it was found that the approach of Lämmer and Helbing [5] was unable to find the optimal signal timing settings, and, while both approaches proposed by Gershenson [4] were able to achieve the optimal signal timing settings, the parameter values required to do so were undesirably large. These results highlight the potential of the self-organising traffic control strategies, but they also show that there is room for improvement in all three algorithmic approaches, particularly for oversaturated traffic conditions.

7 Recommendations for future work

The next logical step in testing self-organising traffic control approaches is to test the effectiveness of the optimised pre-timed signal settings in a simulated environment for stochastic, random arrivals generated according to a Poisson distribution with a mean equal to the deterministic arrival rate. This will also provide a test bench on which self-organising traffic control approaches may be compared in a more realistic environment.

It is also recommended that further research be undertaken into developing a self-organising traffic control approach which improves upon those reviewed in this paper. One particular area of improvement may involve the prioritisation strategy of Lämmer and Helbing [5]. One suggestion is for the priorities of each approach to be based on economic order quantity models in which approaches to an intersection effectively "order" green time from the intersection, where the order size is determined by the number of vehicles requiring service as well as their distance from the intersection and the speed with which they are approaching it. To account for problems that occur during periods of high traffic volumes, an osmosis-based approach may also be adopted in which the "pressures" on either side of an intersection are considered. This would ensure that the green time provided to an approach by an intersection is only long enough to ensure that the number of vehicles that pass through the intersection onto an adjoining approach does not exceed the available capacity of the adjoining approach.

References

[1] Arfken GB & Weber HJ, 2001, *Mathematical methods for physicists*, Fifth edition, Academic Press, San Diego (CA).

[2] Arnon DS, Collins GE & McCallum, S, 1984, *Cylindrical algebraic decomposition I: The basic algorithm*, SIAM Journal on Computing, **13(4)**, pp. 865–877.

[3] Fambro DB & Rouphail NM, 1997, *Generalized delay model for signalized intersections and arterial streets*, Transportation Research Record: Journal of the Transportation Research Board, **1572(1)**, pp. 112–121.

[4] Gershenson C, 2004, *Self-organising traffic lights*, arXive preprint nlin/0411066.

[5] Lämmer S & Helbing D, 2008, *Self-control of traffic lights and vehicle flows in urban road networks*, Journal of Statistical Mechanics: Theory and Experiment, **2008**, p. P04019.

[6] Papacostas CS & Prevedouros PD, 2001, *Queueing and simulation*, pp. 611–625 in Curless L (Ed), *Transportation engineering and planning,* Prentice-Hall, Upper Saddle River (NJ).

[7] Wolfram Research, Inc., 2010, *Mathematica*, Version 8, Champaign (IL).

[8] Transportation Research Board, 2000, *Highway capacity manual*, Washington (DC).

Proceedings of the 2013 ORSSA Annual Conference
pp. 19–30
www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf

ORSSA Proceedings
ISBN
XXX-XXXXX-XXXX-X

A binary programming approach towards achieving effective graph protection

AP Burger*, AP de Villiers† & JH van Vuuren‡

Abstract

In this paper we adopt an integer programming approach towards computing five NP-hard parameters which frequently appear in the graph theoretic literature on the protection or safeguarding of networks. These parameters are the domination number, the total domination number, the Roman domination number, the weak Roman domination number and the secure domination number of a graph. In applications the vertices of the graph denote physical entities that are typically geographically dispersed and which have to be secured or monitored, while the graph edges model links between these entities along which patrolling guards stationed at the vertices may monitor entities or move to entities in order to resolve security threats that may occur at the entities. The five parameters mentioned above represent the minimum number of guards required to protect the entire graph of entities under different conditions (*i.e.* for different definitions of the notion of "protection").

We investigate the effectiveness of an integer programming approach towards determining these parameters for small graphs (with at most 99 vertices), medium-sized graphs (with between 100 and 999 vertices) and large graphs (1 000 vertices or more). The first three parameters above are classified as being applicable in a static protection framework, while the latter two apply to dynamic protection strategies. It is found that the three static parameters may be computed within a reasonable time for small and medium-sized graphs by a state-of-the-art commercial integer programming solvers, while the two dynamic parameters may thus be computed within a reasonable time for small graphs only. For large graphs more sophisticated solution approaches (*e.g.* column generation or approximate, metaheuristic solution approaches) are required to determine upper bounds on all five parameters.

1 Introduction

During the fourth century A.D., the Roman Empire had a total of twenty five legions at its disposal to defend its territories. Each legion consisted of various infantry and cavalry units [18]. A grouping of six legions, called a *field army*, was deemed sufficient to secure any one of the eight regions represented by the vertices of the graph Ξ superimposed on the map of the Empire in Figure 1. Emperor Constantine the Great (274–337 A.D.) therefore commanded four field armies and had to decide how to deploy these field armies. The Emperor considered a deployment capable of securing the entire Empire if every one of its eight regions was either

*Department of Logistics, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: apburger@sun.ac.za

†Corresponding author: Department of Logistics, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: antondev@sun.ac.za

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

occupied by a field army or was directly adjacent to a region occupied by two field armies [22], where adjacency is indicated by the edges of the graph Ξ in Figure 1 (these edges represented deployment routes at the time). Constantine's reasoning was that two field armies had to be stationed in a region before one would be allowed to move to a neighbouring, unoccupied region in order to deal with an internal uprising or external defence challenge there, so as to ensure that the region vacated by the moving field army could not be attacked successfully by an enemy.

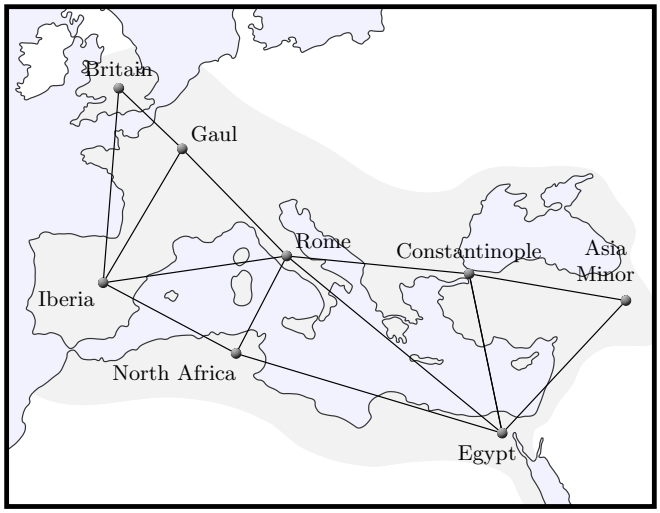


Figure 1: *A graph model Ξ of the eight main regions of the Roman Empire during the fourth century A.D.*

It is not immediately obvious whether or not the entire empire could have been defended by only four field armies. Emperor Constantine, in fact, chose to sacrifice Britain by securing the central regions of the Empire when he stationed two field armies in Rome and two in Constantinople, as shown in Figure 2(a). He could, however, have secured the entire Empire by rather stationing one field army in Britain, two in Rome and one in Asia Minor, as shown in Figure 2(b). The defence strategy of Emperor Constantine is one of the earliest recorded facility location problems and is, in fact, a generalisation of the modern notion of *domination* in graph theory, formalised in the seminal work of Berge [1] and Ore [20] during the period 1958–1962. The subfield of graph domination has numerous applications other than the safeguarding of graphs, but in keeping with the application above, we use the more generic term *guard* instead of *field army* in our explanation of the concept of graph domination and its variations.

A *dominating set* of a graph G is a subset X of the vertex set of G, where X represents those vertices of G that receive one guard each, with the property that each vertex of G which is not in X should be adjacent to at least one vertex in X. The *domination number* of G, denoted by $\gamma(G)$, is the minimum value of $|X|$, taken over all dominating sets X of G (*i.e.* the smallest number of guards that can possibly form a dominating set of G). A dominating set of smallest cardinality for the graph Ξ in Figure 1 is depicted in Figure 2(c), showing that $\gamma(\Xi) = 2$. The notion of domination in graphs has attracted significant attention in the graph theory literature since the 1970s [14, 15] and is a special case of the celebrated *set cover problem* in the operations research literature, as described in the work of Caprara *et al.* [8].

A very natural variation on the theme of domination is that of *total domination*, introduced by Cockayne *et al.* [9] in 1980. A *total dominating set* of a graph G is a subset X_t of the vertex set of G, where X_t represents those vertices of G that receive one guard each, with the property that *every* vertex of G should be adjacent to at least one vertex in X_t (*i.e.* in addition to the set being dominating, every guard should also be adjacent to at least one other guard). The *total domination number* of G, denoted by $\gamma_t(G)$, is the minimum value of $|X_t|$, taken over all

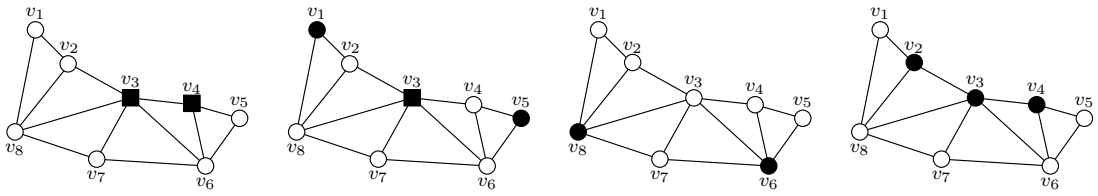


Figure 2: *(a) Emperor Constantine's defence strategy was to station two field armies in Rome and two in Constantinople. (b) A better defence strategy, avoiding the sacrifice of Britain. (c) A dominating set of minimum cardinality for the graph Ξ. (d) Both a weak Roman and secure dominating set of minimum cardinality for the graph Ξ. In all cases $v_1 \equiv$ Britain, $v_2 \equiv$ Gaul, $v_3 \equiv$ Rome, $v_4 \equiv$ Constantinople, $v_5 \equiv$ Asia Minor, $v_6 \equiv$ Egypt, $v_7 \equiv$ North Africa, $v_8 \equiv$ Iberia. Furthermore, $\circ \equiv$ unoccupied vertex, $\bullet \equiv$ vertex occupied by one guard or field army and $\blacksquare \equiv$ vertex occupied by two guards or field armies.*

total dominating sets X_t of G (*i.e.* the smallest number of guards that can possibly form a total dominating set of G). A total dominating set of smallest cardinality for the graph Ξ in Figure 1 is shown in Figure 2(d), from which it follows that $\gamma_t(\Xi) = 3$. The notion of total domination was inspired by policing and monitoring applications where the set X_t represents vertices at which guards are placed, but with the additional requirement that each guard should himself also be monitored by at least one other guard for auditing purposes in an attempt at safeguarding against corruption of the guards.

Constantine's defence strategy of the Roman Empire has inspired yet another variation on the concept of domination. A *Roman dominating set* of a graph G is a pair (X_R, Y_R) of mutually exclusive subsets of the vertex set of G, where X_R represents those vertices of G that receive one guard each and where Y_R represents those vertices of G that receive two guards each, with the property that each vertex which is neither in X_R nor in Y_R should be adjacent to at least one vertex in Y_R. The *Roman domination number* of G, denoted by $\gamma_R(G)$, is the minimum value of $|X_R|+2|Y_R|$, taken over all Roman dominating set pairs (X_R, Y_R) of G (*i.e.* the smallest number of guards that can possibly form a Roman dominating set of G). It is not difficult to show that $\gamma_R(\Xi) = 4$, *i.e.* that the Roman defence strategy in Figure 2(b) is best possible. The notion of Roman domination in graphs has been studied by various authors [10, 11, 23, 25].

Under the assumption that no two regions of the Roman Empire would be attacked simultaneously, Emperor Constantine could have defended the empire using even fewer than four of his thinly stretched field armies. This observation led Henning and Hedetniemi [16] to introduce the notion of weak Roman domination in 2003. A *weak Roman dominating set* of a graph G is a pair (X_r, Y_r) of mutually exclusive subsets of the vertex set of G, where X_r again represents those vertices of G that receive one guard each and where Y_r again represents those vertices of G that receive two guards each, but with the property that $X_r \cup Y_r$ forms a dominating set of G and additionally, for each vertex u in neither X_r nor Y_r, there exists a vertex $v \in X_r$ such that the *swap set* $((X_r \cup \{u\}) - \{v\}) \cup Y_r$ is again a dominating set of G, or a vertex $v \in Y_r$ such that the *swap set* $(Y_r - \{v\}) \cup (X_r \cup \{u, v\})$ is again a dominating set of G. The notion of a swap set models the situation where a guard moves from a single occupied vertex v or a doubly occupied vertex v' to an unoccupied vertex u in order to deal with a problem at u, but leaving the resulting configuration a dominating set of G again. The *weak Roman domination number* of G, denoted by $\gamma_r(G)$, is the minimum value of $|X_r| + 2|Y_r|$, taken over all weak Roman dominating set pairs (X_r, Y_r) of G (*i.e.* the smallest number of guards that can possibly form a weak Roman dominating set of G). The minimum total dominating set in Figure 2(d) is incidently also a weak Roman dominating set of minimum cardinality for the graph Ξ in Figure 1. To see this, note that the guard at v_2 (field army in Gaul) is able to move to either of the vertices v_1

or v_8 (Britain or Iberia) if a security threat were to occur there. The guard at v_3 (field army in Rome) can similarly defend the unoccupied vertex v_2 (North Africa), while the guard at v_4 (field army in Constantinople) can defend the unoccupied vertices v_5 or v_6 (Asia Minor or Egypt). It is not too difficult to show that the weak Roman dominating set in Figure 2(d) is best possible and hence that $\gamma_r(\Xi) = 3$. Note that a total dominating set of a graph G is not always a weak Roman dominating set of G; it is a mere coincidence for the graph Ξ.

Whereas the possibility of placing two guards at a vertex within the context of Roman domination is historically well-founded, this seems to be an artificial construct in the relaxed setting of weak Roman domination, as was noted by Cockayne *et al.* [11] in 2004, thus leading to the simpler notion of *secure domination* where each vertex of the graph can accommodate at most one guard. A *secure dominating set* of a graph G is therefore a subset X_s of the vertex set of G, where X_s represents those vertices of G that receive one guard each, with the property that X_s forms a dominating set of G and additionally, for each vertex u not in X_s, there exists a vertex $v \in X_s$ such that the *swap set* $(X_s - \{v\}) \cup \{u\}$ is again a dominating set of G. Here the swap set again models the situation where a guard moves from an occupied vertex v to an unoccupied vertex u, again leaving the resulting configuration a dominating set. The *secure domination number* of G, denoted by $\gamma_s(G)$, is the minimum value of $|X_s|$, taken over all secure dominating sets X_s of G (*i.e.* the smallest number of guards that can form a secure dominating set of G). Since each vertex in the weak Roman dominating set of the graph Ξ shown in Figure 2(d) accommodates a single guard already, it is also a secure dominating set of G. It is possible to show that $\gamma_s(\Xi) = 3$. However, the parameters $\gamma_r(G)$ and $\gamma_s(G)$ differ for graphs in general.

In this paper we formulate the problems of computing the parameters $\gamma(G)$, $\gamma_R(G)$, $\gamma_t(G)$, $\gamma_r(G)$ and $\gamma_s(G)$ for a general graph G as binary programming problems and we investigate the effectiveness of state-of-the-art software available for solving our binary programming models for graphs of different sizes. We continue our exposition in §2 with a brief review of algorithms available in the literature for computing these parameters, after which we formulate the various computation problems in §3. Numerical results obtained by solving the binary programs of §3 for the classes of square grid and hexagonal graphs are presented in §4, after which the paper is closes in §5 with some ideas for future work.

2 Algorithms for computing $\gamma(G)$, $\gamma_R(G)$, $\gamma_t(G)$, $\gamma_r(G)$ and $\gamma_s(G)$

The decision problem associated with computing the value of $\gamma(G)$ for a general graph G is NP-complete [13] and so are the decision problems associated with computing the four related parameters $\gamma_R(G)$, $\gamma_r(G)$, $\gamma_s(G)$ and $\gamma_t(G)$. There is, however, a significant difference in practical terms between the problems of computing the parameters of domination, total domination and Roman domination on the one hand and those of computing the parameters of weak Roman domination and secure domination on the other, as noted by Burger *et al.* [2, 3]. This difference is rooted in the inherent static nature of the former three problems (in the sense that no guard moves are required) and the dynamic element of a guard swap set introduced into the latter two problems (requiring a dominating set both before and after the guard move). It is therefore expected that the computational burden involved in computing $\gamma_r(G)$ or $\gamma_s(G)$ should be larger than that of computing $\gamma(G)$, $\gamma_t(G)$ or $\gamma_R(G)$ for a general graph G. The relationships

$$\gamma(G) \leq \begin{cases} \gamma_t(G) \leq 2n/3 \\ \gamma_r(G) \leq \begin{cases} \gamma_R(G) \leq 2\gamma(G) \\ \gamma_s(G) \leq n-1 \end{cases} \end{cases} \tag{1}$$

nevertheless hold for any graph G of order $n \geq 3$ [10, 11, 16] and may be used to bound the hard parameters by the easier ones.

The domination number $\gamma(G)$ of a general graph G may be computed in exponential time by a branch-and-reduce approach which has a worst-case time complexity of $\mathcal{O}(1.4969^n)$ [28]. This parameter may, however, be computed in linear time if G is a tree [19]. An upper bound on $\gamma(G)$ may therefore be computed for an order n graph G of size m in $\mathcal{O}(nm)$ time by first computing a spanning tree T of G using a depth-first search in $\mathcal{O}(nm)$ time [26] and then computing $\gamma(T)$ in $\mathcal{O}(n)$ time. The resulting polynomial-time bound $\gamma(G) \leq \gamma(T)$ is expected to be good for graphs G with few cycles. Similarly, Laskar *et al.* [17] developed a linear algorithm for determining $\gamma_t(G)$ if G is a tree.

The problem of graph domination may be modelled by the celebrated unicost set cover problem [27], whose universe is the vertex set of G which must be covered by the set $\mathcal{A}$ of closed neighbourhoods of the vertices of G. The goal in the set cover problem is to find a smallest subset of $\mathcal{A}$, the union of which contains all elements of the universe. Caprara *et al.* [8] presented exact and efficient heuristic solution approaches for the set cover problem.

Exponential time (branch-and-bound and branch-and-reduce) algorithms are also available for computing the secure domination number $\gamma_s(G)$ of a general graph G [5], while this parameter can also be computed in linear time for trees [4].

As far as we are aware, there have been no attempts in the literature at designing purpose-made algorithms for computing the parameters $\gamma_R(G)$ and $\gamma_r(G)$.

3 Binary programming problem formulations

Let G be a graph of order n with vertex set $V(G) = \{v_1, \ldots, v_n\}$. Furthermore, suppose the entry in row i and column j of the adjacency matrix of G is denoted by a_{ij} for all $i \neq j$, with the convention that $a_{ii} = 1$ for all $i = 1, \ldots, n$.

3.1 Formulations for the static domination problems

Let X be the set of vertices containing exactly one guard and let Y be the set of vertices containing two guards (when applicable) in the five computation problems described in the introduction. Define the binary decision variables

$$x_i = \begin{cases} 1 & \text{if } v_i \in X \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad y_i = \begin{cases} 1 & \text{if } v_i \in Y \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

for all $i = 1, \ldots, n$.

3.1.1 Domination

The problem of computing a dominating set of minimum cardinality for G may be formulated as a binary program in which the objective is to

$$\text{minimise} \quad z = \sum_{i=1}^{n} x_i \tag{3}$$

subject to the constraints

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1, \qquad i = 1, \ldots, n. \tag{4}$$

Constraint set (4) ensures that each vertex of G is adjacent to at least one vertex in the set X.

3.1.2 Total domination

The problem of computing a total dominating set of minimum cardinality for G may be formulated as a binary program in which the objective is to

$$\text{minimise} \quad z_t = \sum_{i=1}^{n} x_i \tag{5}$$

subject to the constraints

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1, \qquad i = 1, \ldots, n, \tag{6}$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j \geq x_i, \qquad i = 1, \ldots, n. \tag{7}$$

Constraint set (6) ensures that each vertex of G is adjacent to at least one vertex in the set X (*i.e.* that X is a dominating set of G). In (7) it is additionally ensured that each occupied vertex v_i (*i.e.* for which $x_i = 1$) is adjacent to at least one vertex in X other than v_i.

3.1.3 Roman domination

The problem of computing a Roman dominating set of minimum cardinality for G may be formulated as a binary program in which the objective is to

$$\text{minimise} \quad z_R = \sum_{i=1}^{n} (x_i + 2y_i) \tag{8}$$

subject to the constraints

$$\sum_{j=1}^{n} a_{ij} y_j \geq 1 - x_i - y_i, \qquad i = 1, \ldots, n, \tag{9}$$

$$x_i + y_i \leq 1, \qquad i = 1, \ldots, n. \tag{10}$$

Constraint set (9) ensures that each unoccupied vertex v_i (*i.e.* for which $x_i = y_i = 0$) is dominated by at least one vertex in the set Y (*i.e.* containing two guards). Constraint set (10) ensures that the sets X and Y are mutually exclusive (*i.e.* that no vertex is interpreted as being occupied by both one and two guards). Revelle & Rosing [23] proposed a similar but different formulation for Roman domination in 2000.

3.2 Formulations for the dynamic domination problems

Define, in addition to the binary decision variables in (2), the auxiliary variables

$$z_{k\ell} = \begin{cases} 1 & \text{if vertex } v_k \ (x_k = 0 \text{ and } y_k = 0) \text{ and } v_\ell \ (x_\ell = 1 \text{ or } y_\ell = 1) \text{ form a swap set} \\ 0 & \text{otherwise} \end{cases}$$

for all $k, \ell = 1, \ldots, n$ and $k \neq \ell$.

3.2.1 Weak Roman domination

The problem of computing a weak Roman dominating set of minimum cardinality for G may be formulated as a binary program in which the objective is to

$$\text{minimise} \quad z_r = \sum_{i=1}^{n} (x_i + 2y_i) \tag{11}$$

subject to the constraints

$$\sum_{j=1}^{n} a_{ij}(x_j + y_j) \geq 1, \qquad i = 1, \ldots, n, \tag{12}$$

$$x_k + y_k \leq 1, \qquad k = 1, \ldots, n, \tag{13}$$

$$\sum_{\substack{\ell=1 \\ \ell \neq k}}^{n} z_{k\ell} \geq 1 - x_k - y_k, \qquad k = 1, \ldots, n, \tag{14}$$

$$a_{k\ell}(y_\ell + x_\ell - x_k - y_k + 1) \geq 2z_{k\ell}, \qquad \begin{array}{l} k = 1, \ldots, n, \\ \ell = 1, \ldots, n, \quad \ell \neq k, \end{array} \tag{15}$$

$$y_\ell a_{ik} + a_{i\ell} + \sum_{\substack{j=1 \\ j \neq k \\ j \neq \ell}}^{n} a_{ij}(x_j + y_j) \geq z_{k\ell}, \qquad \begin{array}{l} i = 1, \ldots, n, \\ k = 1, \ldots, n, \\ \ell = 1, \ldots, n, \quad \ell \neq k. \end{array} \tag{16}$$

Constraint set (12) ensures that each vertex v_i is adjacent to at least one vertex in X. Constraint set (13) ensures that the sets X and Y are mutually exclusive (*i.e.* that no vertex is interpreted as being occupied by both one and two guards). Constraint set (14) ensures, if the vertex v_k is unoccupied (*i.e.* $x_k = y_k = 0$), that there is a swap set involving v_k, while constraint set (15) ensures that each swap set from v_ℓ to v_k ($z_{k\ell}$) is valid (*i.e.* that v_k is unoccupied and there is an occupied vertex v_ℓ adjacent to v_k from which a guard can move to v_k). Constraint set (16) ensures that the guard configuration remains dominating after any single guard swap from v_ℓ to v_k is performed.

3.2.2 Secure domination

The problem of computing a secure dominating set of minimum cardinality for G may be formulated as a binary program in which the objective is to

$$\text{minimise} \quad z_s = \sum_{i=1}^{n} x_i \tag{17}$$

subject to the constraints

$$\sum_{j=1}^{n} a_{ij}x_j \geq 1, \qquad i = 1,\ldots,n, \tag{18}$$

$$\sum_{\substack{\ell=1\\ \ell\neq k}}^{n} z_{k\ell} \geq 1 - x_k, \qquad k = 1,\ldots,n, \tag{19}$$

$$a_{k\ell}(x_\ell - x_k + 1) \geq 2z_{k\ell}, \qquad \begin{array}{l} k = 1,\ldots,n,\\ \ell = 1,\ldots,n, \quad \ell \neq k, \end{array} \tag{20}$$

$$a_{ik} + \sum_{\substack{j=1\\ j\neq k\\ j\neq \ell}}^{n} a_{ij}x_j \geq z_{k\ell}, \qquad \begin{array}{l} i = 1,\ldots,n,\\ k = 1,\ldots,n,\\ \ell = 1,\ldots,n, \quad \ell \neq k. \end{array} \tag{21}$$

Constraint set (18) ensures that each vertex v_i is adjacent to at least one vertex in X (*i.e.* an occupied vertex). Constraint set (19) ensures, if the vertex v_k is unoccupied (*i.e.* $x_k = 0$), that there is a swap set involving v_k, while constraint set (20) ensures that each swap set from v_ℓ to v_k ($z_{k\ell}$) is valid (*i.e.* that v_k is unoccupied and there is an occupied vertex v_ℓ adjacent to v_k). Finally, constraint set (21) ensures that the guard configuration remains dominating after any single guard swap from v_ℓ to v_k is performed.

3.3 Formulation characteristics

The formulations of the three static parameters require significantly fewer variables and constraints than the formulations for the two dynamic parameters for any graph, as shown in Table 1.

§	Problem	Variables	Constraints
3.1.1	Domination	n	n
3.1.2	Total domination	n	$2n$
3.1.3	Roman domination	$2n$	$2n$
3.2.1	Weak Roman domination	$n^2 + 2n$	$n^3 + n^2 + 3n$
3.2.2	Secure domination	$n^2 + n$	$n^3 + n^2 + 2n$

Table 1: *The number of variables and constraints in the problem formulations of §3.1–§3.2 for a graph of order n.*

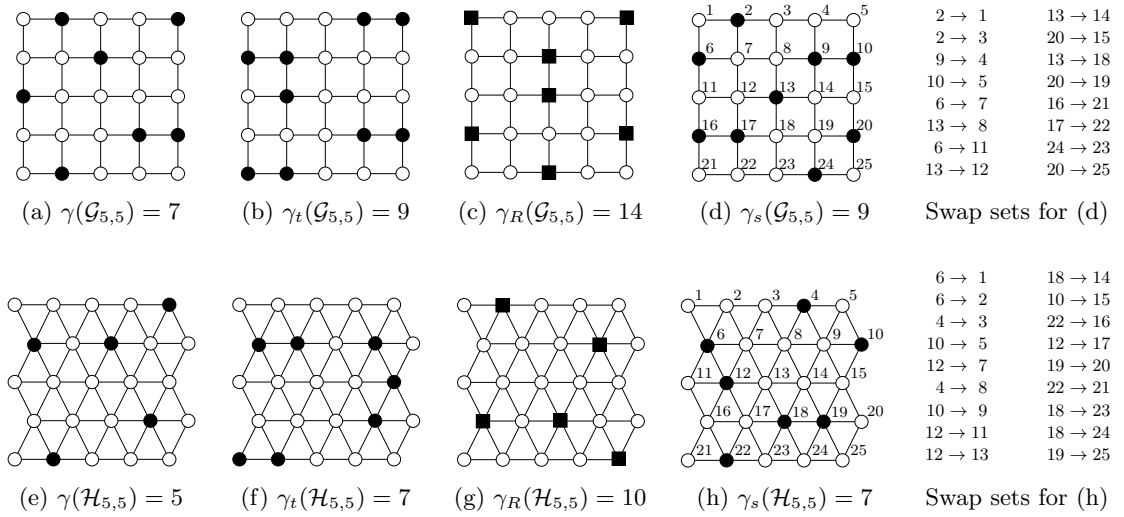
4 Test results

The models of §3 were solved for the families of square grid graphs in the plane and square hexagonal graphs using CPLEX [12]. These graph classes were chosen as test instances because of their frequent use in war games [21, pp. 116] and geographic information system applications [6, pp. 23]. The results thus obtained for the 5×5 grid graph $\mathcal{G}_{5,5}$ and the 5×5 hexagonal graph $\mathcal{H}_{5,5}$ are shown as an example in Figure 3. Results for grid graphs and hexagonal graphs of other orders are summarised in Table 2. All numerical results were computed on an Intel(R) Core(TM)i7-3770 CPU 3.40GHz processor with 8.0 Gb RAM running in Linux Ubuntu [7].

Soltankhah [24] determined the total domination number of a square grid graph analytically as

$$\gamma_t(\mathcal{G}_{n,n}) = \begin{cases} \frac{n^2+2n}{4} & \text{if } n \equiv 0, 2 \pmod 4 \\ \frac{n^2+2n+1}{4} & \text{if } n \equiv 1 \pmod 4 \\ \frac{n^2+2n-3}{4} & \text{if } n \equiv 3 \pmod 4. \end{cases} \tag{22}$$

We used this result as a means of validation, by verifying that the numerical results in Table 2 correspond with the values in (22).



(a) $\gamma(\mathcal{G}_{5,5}) = 7$ (b) $\gamma_t(\mathcal{G}_{5,5}) = 9$ (c) $\gamma_R(\mathcal{G}_{5,5}) = 14$ (d) $\gamma_s(\mathcal{G}_{5,5}) = 9$ Swap sets for (d)

(e) $\gamma(\mathcal{H}_{5,5}) = 5$ (f) $\gamma_t(\mathcal{H}_{5,5}) = 7$ (g) $\gamma_R(\mathcal{H}_{5,5}) = 10$ (h) $\gamma_s(\mathcal{H}_{5,5}) = 7$ Swap sets for (h)

Figure 3: *(a)–(d) Examples of a dominating, total dominating, Roman dominating and secure dominating set of the square grid graph of order 25 in a plane. (e)–(h) Examples of a dominating, total dominating, Roman dominating and secure dominating set of the square hexagonal graph of order 25. The swap sets corresponding to the secure dominating sets in (d) and (h) are also shown.*

For the purposes of this analysis, graphs of order at most 99 were considered *small graphs*, graphs of orders 100 to 999 were considered *medium-sized graphs*, and graphs of order 1 000 and up were considered *large graphs*. As may be seen in the table, the static parameters, γ, γ_t and γ_R, could readily be found by a binary programming solution approach for small and medium sized graphs, while the dynamic parameters, γ_r and γ_s, could only be found for small graphs within a reasonable time frame. For larger graphs a more sophisticated solution approach is required. The results for γ_r and γ_s in Table 2 are identical for square grid graphs and square hexagonal graphs; this result may be established analytically. There are, however, a large number of graph classes for which these two parameters are not equal. The formulation for Roman domination by Revelle & Rosing [23] was used to validate the results in Table 2, and the execution times of their formulation are similar to the execution times of the formulation presented in this paper.

5 Conclusion

Binary programming formulations were presented in this paper for computing five parameters related to the protection of a graph. The five graph protection paradigms considered were classified as being either static (domination, total domination and Roman domination) or dynamic (weak Roman and secure domination). The formulations for both the static and dynamic domination problems were presented in §3, followed by a brief discussion on the characteristics of these formulations. Numerical test results were established in §4 for the families of square grid graphs in the plane and hexagonal graphs. Using a state-of-the-art binary solver (CPLEX) on a relatively fast computing platform, it was found that the static parameters may be found within

	Results for the static domination problems																	
	Results for the grid graph $\mathcal{G}_{n,n}$									Results for the hexagonal graph $\mathcal{H}_{n,n}$								
	Domination			Total domination			Roman domination			Domination			Total domination			Roman domination		
n	LP	γ	Time	LP	γ_t	Time	LP	γ_R	Time	LP	γ	Time	LP	γ_t	Time	LP	γ_R	Time
2	1.33	2	0.01	1.33	2	0.01	2.00	3	0.01	1.00	1	0.01	1.00	2	0.01	2	2	0.01
3	2.50	3	0.01	2.50	3	0.01	4.00	6	0.01	2.00	2	0.01	2.00	2	0.01	4	4	0.01
4	4.00	4	0.01	4.00	6	0.01	6.89	8	0.01	4.00	4	0.01	4.00	4	0.01	6	7	0.01
5	6.27	7	0.01	6.27	9	0.01	10.00	14	0.02	5.00	5	0.01	5.00	7	0.01	9	10	0.01
6	8.75	10	0.01	8.75	12	0.01	14.00	19	0.02	6.00	6	0.01	6.75	8	0.01	12	12	0.01
7	11.50	12	0.04	11.50	15	0.01	18.67	24	0.04	9.00	9	0.01	9.00	12	0.04	16	18	0.01
8	14.81	16	0.05	14.81	20	0.02	23.93	32	0.05	10.33	12	0.01	11.50	16	0.08	20	23	0.01
9	18.25	20	0.08	18.25	25	0.03	29.90	38	0.03	13.50	14	0.02	13.75	18	0.16	25	28	0.02
10	22.39	24	0.09	22.39	30	0.02	36.52	48	0.11	16.86	18	0.02	16.92	23	0.16	30	35	0.04
11	26.81	29	0.22	26.81	35	0.10	43.75	57	0.19	20.13	21	0.03	20.13	28	0.44	36	41	0.04
12	31.57	35	0.25	31.57	42	0.07	51.11	67	0.22	23.78	24	0.05	23.85	32	0.59	42	48	0.07
13	36.93	40	2.76	36.93	49	0.08	60.34	78	0.27	27.54	29	0.08	27.56	38	3.96	49	58	0.19
14	42.40	47	5.02	42.40	56	0.11	69.60	88	0.26	31.66	34	0.18	31.58	44	37.72	56	66	0.22
15	48.53	53	5.82	48.53	63	0.15	79.47	102	0.63	36.11	38	0.18	36.19	50	113.91	64	75	0.24
16	54.81	60	33.17	54.81	72	0.29	90.14	115	1.21	40.67	43	0.24	40.70	56	769.09	72	85	0.59
17	61.71	68	803.47	61.71	81	0.41	101.44	129	1.70	45.85	48	3.37	45.93	63	21 157.32	81	96	5.38
18	69.09	76	16 037.67	69.09	90	0.59	113.35	144	6.00	50.79	54	1.78	50.82	$72^{\dagger}$	MO	90	107	6.67
19	76.53	$84^{\dagger}$	MO	76.53	99	0.37	126.00	158	0.98	56.56	60	7.02	—	—	—	100	119	13.44
20	—	—	—	84.67	110	0.89	139.26	176	18.38	62.31	66	12.30	—	—	—	110	131	57.98
21	—	—	—	92.98	121	1.11	153.23	193	27.30	68.37	72	96.41	—	—	—	121	144	144.93
22	—	—	—	101.80	132	1.37	167.88	211	58.35	75.00	79	150.48	—	—	—	132	158	2 801.59
23	—	—	—	111.01	143	1.40	183.09	230	105.57	81.34	86	358.93	—	—	—	144	170	89.27
24	—	—	—	120.65	156	0.93	199.03	248	103.52	88.63	93	1 159.16	—	—	—	156	186	9 495.27
25	—	—	—	130.83	169	2.13	215.75	270	559.34	95.62	101	5 887.20	—	—	—	169	200	3 537.32
26	—	—	—	141.09	182	2.75	232.96	291	571.61	103.34	$109^{\dagger}$	MO	—	—	—	182	$216^{\dagger}$	MO
27	—	—	—	151.94	195	1.60	250.92	313	963.33	—	—	—	—	—	—	—	—	—
28	—	—	—	163.15	210	4.09	269.61	336	3 848.79	—	—	—	—	—	—	—	—	—
29	—	—	—	174.78	225	4.27	288.84	358	734.03	—	—	—	—	—	—	—	—	—
30	—	—	—	186.84	240	6.45	308.81	384	14 378.69	—	—	—	—	—	—	—	—	—

		Results for the dynamic domination problems					
		Weak Roman domination			Secure domination		
	n	LP	γ_r	Time	LP	γ_s	Time
$\mathcal{G}_{n,n}$	2	1.33	2	0.01	1.33	2	0.01
	3	2.50	4	0.02	2.50	4	0.02
	4	4.00	7	0.05	4.00	7	0.07
	5	6.27	9	0.15	6.27	9	0.17
	6	8.75	13	0.53	8.75	13	0.39
	7	11.50	18	1.93	11.50	18	2.90
	8	14.81	23	221.94	14.81	23	72.16
	9	18.25	29	13 974.94	18.25	29	23 356.24
	10	22.39	$36^{\dagger}$	MO	22.39	36^{*}	TO
$\mathcal{H}_{n,n}$	2	1.00	2	0.01	1.00	2	0.01
	3	2.00	3	0.01	2.00	3	0.01
	4	4.00	5	0.04	4.00	5	0.04
	5	5.00	7	0.09	5.00	7	0.08
	6	6.00	10	0.37	6.00	10	0.27
	7	9.00	13	6.25	9.00	13	7.41
	8	11.33	17	160.10	11.33	17	110.00
	9	13.50	21	859.64	13.50	21	1 083.17
	10	16.86	$27^{\dagger}$	MO	16.86	$27^{\dagger}$	MO

Table 2: *Results obtained by solving the formulations in §3 for square grid graphs in the plane ($\mathcal{G}_{n,n}$) and "square" hexagonal graphs ($\mathcal{H}_{n,n}$). The values shown in columns labelled 'LP' are the associated linear programming relaxation lower bounds. All times are measured in seconds. A time-out bound of 8 hours (28 800 seconds) was enforced. If a problem instance required more than 8 hours of computation time, the acronym TO is shown in the time column and the time-out upper bound value on the parameter is accompanied by an asterisk. A memory-out bound of 8 Gb was enforced. If a problem instance required more than 8 Gb of memory, the acronym MO is shown in the time column and the memory-out upper bound value on the parameter is accompanied by a dagger.*

a reasonable timeframe for graphs of order at most 1 000, while the dynamic parameters can only be found for graphs of order at most 100.

Further work may include adopting a more sophisticated exact solution approach (*e.g.* a column generation approach or the addition of valid inequalities) to find the values of the parameters for graphs of order more than 1 000 or to adopt an approximate solution approach (*e.g.* a local search or metaheuristic, such as those by Caprara *et al.* [8] and Yagiura *et al.* [29] for the set cover problem) in order to establish good upper bounds on the values of the parameters.

Acknowledgements

The research towards this paper was supported financially by the South African National Research Council under grant numbers 70593, 77248 and 81558.

References

[1] C Berge, 1962, *Theory of graphs and its applications*, Methuen, London.

[2] AP Burger, EJ Cockayne, WR Gründlingh, CM Mynhardt, JH van Vuuren & W Winterbach, 2004, *Finite order domination in graphs*, Journal of Combinatorial Mathematics and Combinatorial Computing, **49**, pp. 159–175.

[3] AP Burger, EJ Cockayne, WR Gründlingh, CM Mynhardt, JH van Vuuren & W Winterbach, 2004, *Infinite order domination in graphs*, Journal of Combinatorial Mathematics and Combinatorial Computing, **50**, pp. 179–194.

[4] AP Burger, AP de Villiers & JH van Vuuren, *A linear algorithm for secure domination in trees*, Discrete Applied Mathematics, To appear.

[5] AP Burger, AP de Villiers & JH van Vuuren, 2013, *Two algorithms for secure graph domination*, Journal of Combinatorial Mathematics and Combinatorial Computing, **85**, pp. 321–339.

[6] PA Burrough & RA McDonnell, 1998, *Principles of geographical information systems*, Oxford University Press, Oxford.

[7] Canonical, 2004, *Ubuntu Edition: Version 10.04*, [Online], [Cited April 15th, 2013], Available from `http://www.canonical.com/`.

[8] A Caprara, P Toth & M Fischetti, 2000, *Algorithms for the set covering problem*, Annals of Operations Research, **98**, pp. 353–371.

[9] EJ Cockayne, RM Dawes & ST Hedetniemi, 1980, *Total domination in graphs*, Networks, **10**, pp. 211–219.

[10] EJ Cockayne, PA Dreyer, SM Hedetniemi & ST Hedetniemi, 2004, *Roman domination in graphs*, Discrete Mathematics, **278**, pp. 11–22.

[11] EJ Cockayne, PJP Grobler, WR Gründlingh, J Munganga & JH van Vuuren, 2005, *Protection of a graph*, Utilitas Mathematica, **67**, pp. 19–32.

[12] IBM ILOG CPLEX Optimization Studio, 2012, *CPLEX Edition: Version 12.5*, [Online], [Cited April 15th, 2013], Available from `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`.

[13] MR Garey & DS Johnson, 1979, *Computers and intractability: A guide to the theory of NP-completeness*, Freeman, New York (NY).

[14] TW Haynes, ST Hedetniemi & PJ Slater, 1997, *Fundamentals of domination in graphs*, Marcel Dekker, New York (NY).

[15] TW Haynes, ST Hedetniemi & PJ Slater, 1997, *Domination in graphs: Advanced topics*, Marcel Dekker, New York (NY).

[16] MA Henning & ST Hedetniemi, 2003, *Defending the Roman Empire — A new strategy*, Discrete Mathematics, **266**, pp. 239–251.

[17] R Laskar, J Pfaff, SM Hedetniemi & ST Hedetniemi, 1984 *On the algorithmic complexity of total domination*, Algebraic and Discrete Methods, **5(3)**, pp. 420–425.

[18] EN Luttwak, 1976, *The grand strategy of the Roman Empire*, Johns Hopkins University Press, Baltimore (MD).

[19] SL Mitchell, EJ Cockayne & ST Hedetniemi, 1979, *Linear algorithms in recursive representations of trees*, Journal of Computer and System Sciences, **18**, pp. 76–85.

[20] O Ore, 1962, *Theory of graphs*, American Mathematical Society, Colloquium Publications, Providence (RI).

[21] PP Perla, 1990, *The art of wargaming*, Naval Institute Press, Annapolis (MD).

[22] I Peterson, 2000, *Math Trek: Defending the Roman Empire*, Science News Online, September 9th, [Online], [Cited 2013, January 3rd], Available from: `http://www.sciencenews.org/view/generic/id/837/description/Defending_the_RomanEmpire`.

[23] CS Revelle & KE Rosing, 2000, *Defendes imperium Romanum: A classical problem in military strategy*, American Mathematical Monthly, **107(7)**, pp. 585–594.

[24] N Soltankhah, 2010, *Results on total domination and total restrained domination in grid graphs*, International Mathematical Forum, **5**, pp. 319–332.

[25] I Stewart, 1999, *Defend the Roman Empire!*, Scientific American, December Issue, pp. 136–138.

[26] R Tarjan, 1972, *Depth-first search and linear graph algorithms*, Journal of Computing, **1(2)**, pp. 146–160.

[27] C Toregas, R Swain, C ReVelle & L Bergman, 1971, *The location of emergency service facilities*, Operations Research, **19(6)**, pp. 1363–1373.

[28] JMM van Rooij & HL Bodlaender, 2011, *Exact algorithms for dominating set*, Discrete Applied Mathematics, **159**, pp. 2147–2164.

[29] M Yagiura, M Kishida & I Ibaraki, 2006, *A 3-flip neighborhood local search for the set covering problem*, European Journal of Operational Research, **172**, pp. 472–499.

Proceedings of the 2013 ORSSA Annual Conference
pp. 31–39
www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf

ORSSA Proceedings
ISBN
xxx-xxxxx-xxxxx-x

A Comparative Study of Genetic Algorithms Using a Direct and Indirect Representation in Solving the South African School Timetabling Problem

Rushil Raghavjee[1] Nelishia Pillay[2]

Abstract

Previous work applying genetic algorithms to solve the school timetabling problem have generally used a direct representation, in which each chromosome represents a timetable directly. This study proposes and evaluates a genetic algorithm employing an indirect chromosome representation. Each chromosome is a string comprised of instructions which are used to build a timetable. The fitness of each chromosome is a function of the hard and soft constraint violations of the timetable constructed using the chromosome. Tournament selection is used to choose parents which the mutation and crossover operators are applied to in order to create successive generations. The performance of the genetic algorithm using an indirect representation (IGA) was compared to that using a direct representation (DGA) in solving the school timetabling problem for a South African primary and high school. Both genetic algorithms were able to produce feasible timetables of good quality with the IGA performing better than the DGA. The difference in performance was found to be statistically significant.

Key words: School timetabling, genetic algorithms, indirect representations

1 Introduction

The school timetabling problem is an optimization problem faced by all schools. It essentially involves the allocation of classes, teachers, venues and other resources to a school timetable in a manner such that the requirements specified by the school have been met. A single meeting

[1]School of Management, Information Technology and Governance, University of KwaZulu-Natal, Pietermaritzburg, KwaZulu-Natal, South Africa, email: raghavjee@ukzn.ac.za
[2] School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Pietermaritzburg, KwaZulu-Natal, South Africa, email: pillayn32@ukzn.ac.za

between a class and a teacher that involves a subject and a venue is referred to as a tuple and all tuples specified by the school must be allocated to a period on the timetable. Requirements specified by the school are divided into two categories, namely hard constraints and soft constraints. Hard constraints specified by the school must be met in order for the timetable to be usable. Soft constraints specify the requirements that affect timetable quality. The greater the number of soft constraints satisfied, the better the quality of the timetable.

Several techniques have previously been applied to the school timetabling problem. Abramson [1], Avella et al. [4], and Liu et al. [14] used simulated annealing to solve the school timetabling problem for schools in Australia, Italy and Greece respectively. Tabu search was used by Alvarez-Valdez et al. [3], Schaerf [15] and Desef et al. [10] while Beligiannis et al. [6] solved the school timetabling problem using a particle swarm optimization approach. Colorni et al. [9] compared a simulated annealing approach with a tabu search when solving the school timetabling problem while Jacobsen [13] compared tabu search with a constraint programming approach.

A genetic algorithm is an algorithm based on Darwin's theory of evolution [11]. An initial population of individuals is created and is then iteratively refined over a number of generations. This process continues until either a generation limit has been reached or when a solution is found. During each generation, individuals are selected as parents and genetic operators are applied to the selected parents resulting in the creation of offspring. These offspring become individuals in the new generation. The most common methods for selection of parents are either fitness proportionate selection or tournament selection. Offspring are created using genetic operators such as reproduction, crossover and mutation. Beligiannis et al. [7] and Bedoya et al. [5] have solved the school timetabling problem using a genetic algorithm with only a mutation operator. Abramson et al. [2] solved an Australian school timetabling problem using a parallel genetic algorithm. Wilke et al. [16] solved the German school timetabling problem with a genetic algorithm that used hybrid repair operators. For each of these genetic algorithms, a direct representation was used where each individual was represented using a timetable.

This paper evaluates the performance of two genetic algorithms, one using a direct representation (DGA) and the second an indirect representation (IGA) in solving the school timetabling problem. The performance of both GAs was tested on a South African primary and high school timetabling problem. The IGA was found to perform better than the DGA in solving these problems. Section 2 describes the DGA and section 3 the IGA. Section 4 covers the methodology used and describes the two school timetabling problems that the two approaches were applied to. Section 5 provides the results found and a discussion of the results. Finally, Section 6 provides conclusions made in this study and outlines ideas for future research.

2 A Direct Representation Genetic Algorithm (DGA) to Solve the School Timetabling Problem

This genetic algorithm approach consists of two phases. Phase 1 uses a genetic algorithm to search for feasible timetables. Phase 2 uses a genetic algorithm to improve the quality of the feasible timetables found during Phase 1.

The population is comprised of individuals representing a timetable. Individuals are also referred to as chromosomes or candidate solutions. Each individual is represented using a two dimensional matrix. The rows of the matrix represent the periods of the week and columns represent the classes. Each cell stores the teacher that will teach a particular class during a specific period. The cell will also store the subject that is being taught as well as the venue (if applicable). An initial population is created using a sequential construction method (SCM) where for each individual in the initial population, a group of timetables are created and the best (fittest) timetable is added to the initial population. The number of elements in the group is referred to as the SCM size and is the same for each group. This value is problem dependent and is set at the beginning of each run. Each timetable in the group is created by firstly ordering all lessons using the saturation degree heuristic [8]. The saturation degree is calculated to be the number of feasible periods, i.e. periods that do not result in hard constraint violations, available to which the tuple can be allocated. Tuples are then allocated to a feasible period. If there is more than one feasible period, the tuple is allocated to the period which results in the lowest soft constraint cost. If there are no feasible periods available the tuple is allocated to a randomly selected period.

Each timetable is evaluated by counting the number of constraint violations. A feasible timetable is one that has no (zero) hard constraint violations. In terms of timetable quality, a lower soft constraint cost indicates a better quality timetable. Thus the objective is to minimize the soft constraint cost.

For Phase 1, tournament selection is used when selecting parents and mutation is the only genetic operator applied. For Phase 2, two tournament selection methods were considered, namely the standard tournament selection and a variant tournament selection where the selected individual is not always the fittest individual. When comparing individuals in variant tournament selection, an individual is selected either by its fitness or by chance. Like the standard tournament selection, variant tournament selection returns a single parent. Selection is with replacement, thus an individual can be chosen more than once as a parent. The algorithm for the variant tournament selection is shown below:

```
Input: Tournament size t and current population P (t < |P|)
Output: Parent
1  best = Individual randomly chosen from P
2  for J = 2[3] to  t
3     next= Individual randomly chosen from population
4     n = Random number from 1 to 3
5     If n = 1
6        best = fitter individual between best and  next
7     Else if n = 2
8      best = next
9   endfor
10  return best
```

Figure 1: Variant tournament selection algorithm

[3] Note the loop begins at 2 as the first element of the tournament is selected prior to the loop.

For Phase 1, four mutation operators were tested and involved searching for tuples that cause constraint violations and swapping them with randomly chosen tuples. The four operators were named 1VH, 2VH, 1VNH and 2VNH. The 1VH operator searches for a constraint violating tuple and swaps it with a randomly chosen tuple. The 2VH operator swaps two constraint violating tuples. Both these operators use hill climbing where only swaps that reduce the cost of the timetable are accepted. The 1VNH and 2VNH operators are similar to the 1VH and 2VH but do not use hill climbing.

For Phase 2, four mutation operators were also tested and involved the swapping of tuples between periods. The four mutation operators considered for Phase 2 were a random swap, a row swap as well as the 1VH and 2VH operators used in Phase 1. All the operators incorporate hill climbing.

3 An Indirect Representation Genetic Algorithm (IGA) to Solve the School Timetabling Problem

Similar to the DGA, the IGA approach also consists of two phases that separately address hard constraints and soft constraints. Phase 1 will attempt to find a feasible timetable while Phase 2 (if needed) will focus on improving the quality of the feasible timetable found in Phase 1.

Each individual in the population is represented by a string of characters. The length of each chromosome is randomly chosen to be in the inclusive range of 1 and the maximum number of tuples to be allocated. Each character represents an instruction that is capable of building or modifying a timetable. The following list of characters in Table 1 represent each of the low-level heuristics that are randomly allocated to each chromosome.

Table 1: Instructions used by the IGA

A	Allocation – Add an unallocated tuple to the timetable
D	De-allocation – Remove an allocated tuple from the timetable
1..4	Set of mutation operators for Phase 1
5..8	Set of mutation operators for Phase 2

The "A" instruction, when called, allocates a tuple with the lowest saturation degree to the timetable. If there are no tuples to be allocated this operator has no effect and is essentially an intron, i.e. redundant code. The "D" instruction randomly selects a tuple that has already been allocated and removes it from the timetable. The remaining instructions are mutation operators (discussed in the section on the DGA) that make changes to the timetable by moving tuples between periods. If the timetable is empty, i.e. there are no allocations, the deallocate and mutation operators have no effect. An example of an instruction string is *ADA3DA4AA12DDDA*. The first instruction executed is "A" and a tuple is allocated to the timetable. The "D" instruction is then executed and as a result, the tuple is removed from the

timetable. This is followed by the "A" instruction and a tuple is then re-allocated to the timetable. The "3" instruction indicates that one of the mutation operators, namely the 1VH mutation operator, will be executed. This is followed by a de-allocation, an allocation, the execution of mutation operator "4" (2VH), 2 more allocations, the execution of mutation operators 1 (1VNH) and 2 (2VNH), 3 de-allocations and finally an allocation instruction.

The initial population is created by randomly generating each string (individual). Each individual is evaluated based on the cost of the timetable that has been created using the instructions of the string. Similar to the DGA, the timetable cost (and thus the fitness of the individual) is calculated by counting the number of hard constraint and soft constraint violations. Since there is a possibility that incomplete timetables could be induced, the number of unallocated tuples is included in the hard constraint cost, i.e. chromosomes producing timetables that do not contain all the required tuples are penalized and have a poor fitness.

Tournament selection is used when selecting individuals as parents and the genetic operators applied are mutation and crossover. For mutation, a single instruction is selected and is replaced by a randomly chosen instruction. For crossover, the cut-and-splice crossover operator [12] is implemented. A crossover point is selected for each individual and the string fragments are swapped as shown in Figure 2. Each parent is selected using tournament selection. A crossover point in each parent is randomly chosen to be between 1 and the length of the parent chromosome. Note that the last instruction in the first offspring is the deallocate instruction which will result in an incomplete timetable being formed and the chromosome will be penalized accordingly. Also note that application of the crossover operator produces offspring of variable length, with the offspring possibly being of different lengths.

```
Parent 1:  AD1234AAA|DDD121ADAAA        | = crossover point
Parent 2:  12ADDAAA343|4AD                 | = crossover point
Resultant offspring A:  AD1234AAA4AD
Resultant offspring B:  12ADDAAA343DDD121ADAAA
```

Figure 2: Cut-and-Splice Crossover

The Phase 1 genetic algorithm ends when an individual string produces a feasible timetable, i.e. a timetable with no hard constraint violations. This feasible timetable is carried over to Phase 2 which follows a similar approach to Phase 1. The objective of Phase 2 is to improve the quality of the feasible timetable found in Phase 1. An initial population of individuals are randomly generated. The set of instructions are the allocation instruction, de-allocation instruction and the soft constraint mutation operators considered when using the DGA. Each instruction within an individual is applied to the feasible timetable produced in Phase 1. Parents are selected using tournament selection and the genetic operators are mutation and the cut-and-splice crossover operator. The genetic algorithm in Phase 2 continues until a generation limit is reached.

4 Experimental Setup

The main aim of this study is to compare two genetic algorithm approaches that use different representations. The two genetic algorithm approaches described in the two previous sections are each applied to two South African school timetabling problems. The first problem is a

primary school problem with 19 teachers, 16 classes and 14 subjects. A school week is comprised of 5 days and a maximum of 11 periods (the number of periods per day varies for each day and for each grade). The hard (HC) and soft (SC) constraints for this problem are listed below:

- No teacher clashes, class clashes and for some problem instances venue clashes (HC)
- For all classes, mathematics must be taught in the morning periods (HC)
- Co-teaching requirements must be met i.e. for certain lessons, classes are allocated two teachers instead of one (HC)
- Double period requirements must be met (HC)
- Subjects taught to each class must be evenly distributed throughout the week (SC)

The second problem is a secondary school problem with 30 classes, 40 teachers and 44 subjects. A school week is comprised of seven periods in a day and there are six days in a school week. The hard and soft constraints for the problem are as follows:

- No teacher clashes or class clashes (HC)
- All subclass and co-teaching requirements must be met. Subclasses occur when several classes of the same grade are split into two or more groups with each group being taught by a different teacher. (HC)
- Class-period allocation preferences should be met (SC)
- Teacher-period preferences should be met (SC)
- Co-teaching-period and subclass-period preferences should be met (SC)

For the DGA, trial runs were conducted to determine the best mutation operator to use as well as the best genetic parameter values to use. The 1VH, 2VH, 1VNH and 2VNH mutation operators were all tested individually and the 1VH operator performed the best in terms of finding feasible timetables for the secondary school problem. For the primary school problem, a combination of 1VH, 2VH and a random swap performed best in terms of producing feasible timetables. For Phase 2, a random swap operator performed the best for the primary school problem while a combination of the 1VH operator and the row swap performed the best for the secondary school problem. The list of processes and genetic parameters used are listed below:

Table 2: Processes and Parameter values used - DGA

	Primary school problem	Secondary school problem
Selection Method (Phase 1)	Standard tournament selection	Standard tournament selection
Mutation operator (Phase 1)	1VH, 2VH and random swap	1VH
Selection Method (Phase 2)	Variant tournament selection	Standard tournament selection
Mutation operator (Phase 2)	Random swap	1VH row swap
SCM Size	20	20
Population Size	500	750
Tournament Size	10	10
Number of swaps	200	150
Number of generations	75	50

For the IGA, the set of instructions used for the primary school problem were A (Allocate), D (De-allocate), 2 (2VNH), 3 (1VH) and 4 (2VH). For the secondary school problem, the set of instructions used were A, D, 1 (1VNH), 2 (2VNH), 3 (1VH) and 4 (2VH). The parameter values were the same as the values used by the DGA (see Table 2 above) and were determined using trial runs. Please note that the "Number of swaps" refers to the number of swaps performed by a single application of a mutation operator/instruction.

Both approaches were developed using Visual C++ 2010. The random number generator function available in C++ was used to generate random number. Thirty runs were performed with each run using a different random number generator seed. The DGA was run using an Intel Core i7 870 CPU @ 2.93 GHz, 4.00 GB RAM on Windows 7. The IGA was run using the Center for High Performance Computing [17].

5 Results and discussion

This section covers the performance of both the DGA and IGA when solving the two school timetabling problems. Thirty runs were performed for each approach. The success rates indicating the number of feasible timetables found over thirty runs as well as the average timetable quality (average number of soft constraint violations) is shown in the table below.

Table 3: Success Rate and Average Quality

	Primary school			Secondary school		
	Success rate	Average Quality	Maximum Soft Constraint Cost	Success Rate	Average Quality	Maximum Soft Constraint Cost
DGA	60%	10.83	113	67%	4.5	20
IGA	93%	7.61	113	100%	2.37	20

As can be seen from Table 3, the IGA performs far better than the DGA. Hypothesis tests (Z-tests) were conducted to test whether these results were significant. Table 4 lists the levels of significance, critical values and decision rules for the hypothesis tests.

Table 4: Hypothesis test conditions

P	Critical Value	Decision Rule
0.01	2.33	Reject H0 if $Z > 2.33$
0.05	1.64	Reject H0 if $Z > 1.64$
0.10	1.28	Reject H0 if $Z > 1.28$

A single hypothesis was tested for feasibility for each problem, namely, that the IGA performs better than the DGA i.e. the IGA produces fewer constraint violations than the DGA. A Z-value of 2.9 (for the primary school problem) and 3.61 (for the secondary school problem) is obtained indicating that for both school timetabling problems the IGA performs better than the DGA. This was found to be significant at all levels of significance. Reasons for the IGA performing better than the DGA could include the possibility of the size of the search space being reduced when using the IGA as there are fewer combinations making up chromosomes than when using an indirect representation. Furthermore, the indirect representation could also result in a less rugged landscape thereby reducing the chances of premature convergence. Finally, small changes made in the space using the indirect representation may map to larger

changes in the actual solution space, enabling the GA to be more explorative. These reasons will be investigated further as part of future extensions of this work.

The timetables produced using the DGA and IGA were compared to the actual timetables used by the schools. For the primary school problem, the best timetable produced by the DGA was feasible with 6 soft constraint violations. The best timetable produced by the IGA was also feasible and contained 3 soft constraint violations. The actual timetable used by the school contained no clashes but there was one instance of a double period violation not being met. In addition the double periods had to be allocated manually by a staff member. For the secondary school problem, the best timetables produced by both the DGA and IGA were feasible and contained 2 soft constraint violations. The actual timetable used by the school was feasible and also contained 2 soft constraint violations.

6 Conclusion

This research looked at using two different representations for genetic algorithms when solving the school timetabling problem. The results showed that the genetic algorithm approach using an indirect representation, where each individual is a string consisting of instructions capable of building a timetable, performed far better than the genetic algorithm approach using a direct representation, where each individual represented a timetable directly. The results found were statistically significant. It is hypothesized that this could possibly be attributed to the use of an indirect representation resulting in a reduction in search space size, a smoother fitness landscape and greater exploration of the search space. Future research will include a more theoretical analysis of the search area being covered for both the IGA and DGA. Another possible reason that will be included in future research is the fitness landscape of the problems and how the IGA and DGA cover this area.

References

[1] Abramson, D. (1991). Constructing School Timetables using Simulated Annealing: Sequential and Parallel algorithms. *Management Science. 37*(1): 98 – 113.

[2] Abramson, D., Abela, J. (1991). A Parallel Genetic Algorithm for Solving the School Timetabling Problem. *Proceedings of the 15th Australian Conference: Division of Information Technology.* 1-11.

[3] Alvarez-Valdez, R., Martin, G., Tamarit, J.M. (1996). Constructing Good Solutions for the Spanish School Timetabling Problem. *Journal of the Operational Research Society.* 47(10): 1203-1215.

[4] Avella, P., D'Auria, B., Salerno, S. (2007). A Computational Study of Local Search Algorithms for Italian High-School Timetabling. *Journal of Heuristics.* 13(6): 543-556.

[5] Bedoya, C.F., Santos M. (2004). A Non-Standard Genetic Algorithm Approach to Solve Constrained School Timetabling Problems. *Computer Aided Systems Theory – Eurocast 2003, Lecture Notes in Computer Science.* 2809: 26-37.

[6] Beligiannis, G.N., Tassopoulos, I.X. (2012). Solving Effectively the School Timetabling Problem Using Particle Swarm Optimization. *Expert Systems with Applications* . 39: 6029-6040.

[7] Beligiannis, G.N., Moschopoulos, C.N., Kaperonis, G.P., Likothanassis, S.D., (2008) Applying Evolutionary Computation to the School Timetabling Problem: The Greek Case. *Computers and Operations Research.* 35: 1265-1280.

[8] Carter, M.W., Laporte, G. (1996) Recent Developments in Practical Examination Timetabling. *Practice and Theory of Automated Timetabling: Lecture Notes in Computer Science.* 1-21.

[9] Colorni, A., Dorigo, M., Maniezzo, V. (1998). Metaheuristics for High School Timetabling. *Computational Optimization and Applications.* 9: 275–298.

[10] Desef, T., Bortfeldt, A., Gehring, H. (2006). A Tabu Search Algorithm for Solving the Timetabling-Problem for German Primary Schools (Abstract).*Proceedings of the International Conference on the Practice and Theory of Automated Timetabling (PATAT).* Online. Available: http://www.asap.cs.nott.ac. uk/external/watt/patat/patat04/465.ps [Cited October 28, 2013].

[11] Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Westley. ISBN: 0-201-15767-5

[12] Goldberg, D.E., Korb, B., Deb, B. (1989). Messy Genetic Algorithms: Motivation, Analysis and First Results. *Complex Systems.*3(5): 493-530.

[13] Jacobsen, F., Bortfeldt, A., Gehring, H. (2006). Timetabling at German Schools: Tabu Search versus Constraint Programming. *Proceedings of the International Conference on the Practice and Theory of Automated Timetabling.*439-442.

[14] Liu, Y., Zhang, D., Leung, S.C.H. (2009). A Simulated Annealing Approach with a new Neighbourhood Structure for the Timetabling Problem. *Proceedings of GEC2009, First ACM/SIGEVO Summit on Genetic and Evolutionary Computing.*381-386.

[15] Schaerf, A. (1996) Tabu-Search Techniques for Large High-School Timetabling Problems. *IEEE Transactions on Systems, Management and Cybernetics: Part A.*363-368.

[16] Wilke, P., Grobner, M., Oster, N. (2002). A Hybrid Genetic Algorithm for School Timetabling. *Advances in Artificial Intelligence, Lecture Notes in Computer Science.*2557/2002: 455-464.

[17] Center for High Performance Computing. Online. Available: www.chpc.ac.za/sun [Cited October 28th, 2013].

Proceedings of the 2013 ORSSA Annual Conference
pp. 40–49
www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf

ORSSA Proceedings
ISBN
XXX-XXXXX-XXXX-X

The enumeration of k-sets of mutually orthogonal Latin squares

JG Benadé* AP Burger† JH van Vuuren‡

Abstract

Latin squares and sets of k *mutually orthogonal Latin squares* (k-MOLS) have application in various scheduling problems, from providing effective ways to access parallel memory structures to scheduling transmissions from sensor arrays. MOLS also play an important role in sports tournament scheduling where every structurally different MOLS provides the scheduler with an additional degree of scheduling freedom. The existence of 3-MOLS have been resolved for all orders of Latin squares, except for order 10. We consider a backtracking algorithm for the enumeration of structurally different MOLS which partitions the search space in such a way that it is possible to estimate bounds for the enumeration of higher-order MOLS. A contribution towards the celebrated question of the existence of a 3-MOLS of order 10 is made by investigating the feasibility of using this algorithm in conjunction with specific computing paradigms in search of such a design.

Key words: Enumeration, mutually orthogonal Latin squares (MOLS), volunteer computing.

1 Introduction

A *Latin square* of order n is an $n \times n$ array in which every cell contains a single symbol with the property that each symbol occurs exactly once in each row and column of the array [6, Definition 1.1], and two Latin squares are *orthogonal* if each of the n^2 superimposed ordered pairs of symbols, one pair for every (row, column)-position in the arrays, is distinct. Four examples of Latin squares of order 4 may be seen in Figure 1. Note that all three pairs of Latin squares from the set $\{\boldsymbol{L}_1^*, \boldsymbol{L}_2^*, \boldsymbol{L}_3^*\}$ are orthogonal.

$$\boldsymbol{L}_1^* = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \end{bmatrix} \quad \boldsymbol{L}_2^* = \begin{bmatrix} 0 & \mathbf{1} & 2 & 3 \\ 2 & 3 & 0 & \mathbf{1} \\ 3 & 2 & \mathbf{1} & 0 \\ \mathbf{1} & 0 & 3 & 2 \end{bmatrix} \quad \boldsymbol{L}_3^* = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix} \quad \boldsymbol{L}_4^* = \begin{bmatrix} 0 & 3 & 1 & 2 \\ 1 & 2 & 0 & 3 \\ 2 & 1 & 3 & 0 \\ 3 & 0 & 2 & 1 \end{bmatrix}$$

Figure 1: Some Latin squares of order 4.

Latin squares were first formally studied by Leonard Euler when he considered the so-called "36-Officers problem" asking whether it is possible to arrange thirty-six soldiers of six different ranks and from six different regiments in a square platoon with the properties that every row and

*Department of Logistics and MIH Media Lab, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: jgbenade@ml.sun.ac.za

†Department of Logistics, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: apburger@sun.ac.za

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

column contains exactly one soldier of every rank and one soldier from every regiment[1] [8]. Euler was unable to find such an arrangement of soldiers (corresponding to a pair of orthogonal Latin squares of order 6) and continued to propose what has become known as *Euler's Conjecture*, namely that no pair of orthogonal Latin squares of order n exists for $n = 4m + 2$, where m is an integer [8]. In 1900, the French mathematician Gaston Tarry proved Euler's Conjecture correct for $n = 6$, but sixty years later Bose, Shrikhande and Parker [4] showed that it is possible to construct such orthogonal pairs for all cases of Euler's Conjecture other than $n = 6$, thereby disproving the conjecture in general.

The notion of orthogonality may be generalised to sets of k *mutually orthogonal Latin squares*, abbreviated to k-MOLS, which have the property that any two distinct Latin squares in the k-MOLS $\mathcal{M} = \{\boldsymbol{L}_0, \boldsymbol{L}_1, \ldots, \boldsymbol{L}_{k-1}\}$ are pairwise orthogonal. The set $\mathcal{M}_1^* = \{\boldsymbol{L}_1^*, \boldsymbol{L}_2^*\}$ is therefore an example of a 2-MOLS of order 4, while the set $\mathcal{M}_2^* = \{\boldsymbol{L}_1^*, \boldsymbol{L}_2^*, \boldsymbol{L}_3^*\}$ is a 3-MOLS of order 4.

It has been shown that k-MOLS have important applications in coding theory [16], various subfields of statistics (including experimental design) [9, 10], distributed database systems [1] and numerous scheduling problems, including the scheduling of sports tournaments [11, 13, 20]. Moreover, every structurally different set of orthogonal Latin squares provides the scheduler with an alternative schedule, and some of these schedules may be more desirable than others due to *ad hoc* constraints or preferences in the scheduling problem. The known numbers of structurally different k-MOLS for order $n \in \{3, 4, \ldots, 8\}$ appear in Table 1. Additionally, it is known that there are 19 structurally different 8-MOLS of order 9 [19] and that no k-MOLS of order 10 exists for $k \in \{7, 8, 9\}$ [7, 15].

n	k					
	2	3	4	5	6	7
3	1					
4	1	1				
5	1	1	1			
6	0	0	0	0		
7	7	1	1	1	1	
8	2 165	39	1	1	1	1

Table 1: The numbers of structurally different k-MOLS of order n for $n \in \{3, 4, \ldots, 8\}$.

The objective in this paper is to consider an algorithm for the enumeration of structurally different k-MOLS of order n, demonstrating the correctness of this algorithm by replicating the known results in Table 1, and to produce estimates of the sizes of the search spaces for 3-MOLS of orders 9 and 10, which are yet to be enumerated. These estimates should shed light on the current and short-term future feasibility of any further enumeration attempts using this approach.

2 Mathematical preliminaries

Let $S(\boldsymbol{L})$ denote the symbol set of a Latin square $\boldsymbol{L}$ and let $R(\boldsymbol{L})$ and $C(\boldsymbol{L})$ denote its row and column indexing sets, respectively. For any $i \in R(\boldsymbol{L})$ and $j \in C(\boldsymbol{L})$, let $\boldsymbol{L}(i, j) \in S(\boldsymbol{L})$ denote the element in the i-th row and the j-th column of $\boldsymbol{L}$. In the remainder of this paper it is assumed, without any subsequent loss of generality, that $R(\boldsymbol{L}) = C(\boldsymbol{L}) = S(\boldsymbol{L}) = \mathbb{Z}_n$, the set of residues of the integers after division by the natural number n. The *transpose* of $\boldsymbol{L}$, denoted $\boldsymbol{L}^T$, is the Latin square for which $\boldsymbol{L}^T(j, i) = \boldsymbol{L}(i, j)$ for all $i \in R(\boldsymbol{L})$ and $j \in C(\boldsymbol{L})$. Note, for example, that example, $\boldsymbol{L}_1^{*T} = \boldsymbol{L}_4^*$.

The notion of a universal was introduced by Burger *et al.* [5] in 2010 to facilitate the enumeration of specific classes of Latin squares. A *universal* of a Latin square $\boldsymbol{L}$ is a set of n distinct, ordered pairs (i, j), one from each row and column, all containing the same symbol. Universals may be expressed in permutation form such that the *universal permutation* u_ℓ *of the symbol* ℓ maps i to j if $\boldsymbol{L}(i, j) = \ell$ and, as such, it is possible to find the cycle structure

[1] A similar and even older puzzle posed by Claude Gaspard Bachet in 1624 concerns the number of fundamentally different ways in which the 16 court cards from a standard deck of cards may be arranged in a 4×4 square such that every row and column contains a card of every rank and every suit [3].

and inverse of any universal. The *relative cycle structure* of any pair of universals u_1 and u_2 of the same order is defined to be the cycle structure of $u_2 \circ u_1^{-1}$, where $\circ$ is the traditional composition operator for permutations. In $\boldsymbol{L}_2^*$, for example, the entries in boldface correspond to the universal $\{(0,1),(1,3),(2,2),(3,0)\}$ of the symbol 1, which may be written in permutation notation as $\left(\begin{smallmatrix}0&1&2&3\\1&3&2&0\end{smallmatrix}\right)$, abbreviated here as $\langle 1320\rangle$. Let $U(\mathcal{M})$ denote the set of universal permutations of some k-MOLS $\mathcal{M}$, and let $u_\ell(m) \in U(\mathcal{M})$ denote the universal permutation of the symbol ℓ in the m-th square $\boldsymbol{L}_m \in \mathcal{M}$. The set of all universals of $\mathcal{M}_1^*$ is therefore $U(\mathcal{M}_1^*) = \{\langle 0312\rangle, \langle 1203\rangle, \langle 2130\rangle, \langle 3021\rangle, \langle 0231\rangle, \langle 1320\rangle, \langle 2013\rangle, \langle 3102\rangle\}$, while the universal permutation of the symbol 2 in the third Latin square of $\mathcal{M}_2^*$ is $u_2(2) = \langle 2301\rangle$. The relative cycle structure of $u_0(1) \in U(\mathcal{M}_1^*)$ and $u_3(0) \in U(\mathcal{M}_1^*)$ is the cycle structure of the permutation $\langle 3021\rangle \circ \langle 0231\rangle^{-1} = \langle 2013\rangle$, which may be denoted as $z_1^1 z_3^1$ as it consists of one cycle of length 1 and one cycle of length 3.

Latin squares which can be generated from one another by changing the order of their rows and/or columns, and/or by renaming their symbols, are said to be *isotopic*, while Latin squares formed by uniformly applying a permutation to all n^2 3-tuples $(i, j, \boldsymbol{L}(i, j))$ are called *conjugates*. For example, applying the permutation $\left(\begin{smallmatrix}0&1&2\\1&0&2\end{smallmatrix}\right)$ to the 3-tuple $(i, j, \boldsymbol{L}(i, j))$ yields the transpose $(j, i, \boldsymbol{L}(i, j))$ of $\boldsymbol{L}$. A maximal set of isotopic Latin squares, together with all their conjugates, form a *main class* of Latin squares. It is possible to show that $\boldsymbol{L}_1^*$, $\boldsymbol{L}_2^*$, $\boldsymbol{L}_3^*$ and $\boldsymbol{L}_4^*$ are all in the same main class by reordering the rows of $\boldsymbol{L}_1^*$ to find $\boldsymbol{L}_2^*$, transposing $\boldsymbol{L}_1^*$ to form $\boldsymbol{L}_4^*$ and, finally, reordering the columns of $\boldsymbol{L}_4^*$ to form $\boldsymbol{L}_3^*$.

The notions of isotopic and conjugate Latin squares, as well as that of main classes, may be extended to k-MOLS. All k-MOLS which may be generated by row, column and symbol permutations from a given k-MOLS are isotopic, with the additional constraint that the same row or column permutation must be applied to all k Latin squares in the k-MOLS in order to maintain orthogonality (the symbol sets, however, may be renamed independently). Conjugates, in this case, are k-MOLS formed by uniformly applying permutations to the $(k+2)$-tuples $(i, j, \boldsymbol{L}_0(i, j), \ldots, \boldsymbol{L}_{k-1}(i, j))$ and a main class consists of a given k-MOLS, together with its $(k+2)!$ conjugates as well as their respective isotopic k-MOLS.

It is possible to define a *lexicographical ordering*, denoted by the symbol $\prec$, on a main class of k-MOLS by comparing the universals lexicographically in such a way that every main class has a unique smallest element, called the *class representative*. Two k-MOLS, $\mathcal{M}$ and $\mathcal{M}'$ are ordered in this way by comparing corresponding universals, starting with $u_0(0) \in U(\mathcal{M})$ and $u'_0(0) \in U(\mathcal{M}')$, followed by $u_0(1) \in U(\mathcal{M})$ and $u'_0(1) \in U(\mathcal{M}')$, *etc.* until it is either found that the one k-MOLS is lexicographically smaller than the other, or until all universals have been compared, in which case $\mathcal{M}$ and $\mathcal{M}'$ are lexicographically equal and therefore the same k-MOLS. For example, when comparing the two 2-MOLS $\mathcal{M}^* = \{\boldsymbol{L}_1^*, \boldsymbol{L}_2^*\}$ and $\mathcal{M}^{*\prime} = \{\boldsymbol{L}_1^*, \boldsymbol{L}_4^*\}$ of order 4, it is seen that $u_0(0) = u'_0(0)$, $u_0(1) = u'_0(1)$ and $u_1(0) = u'_1(0)$ but $u_1(1) = \langle 1320\rangle \prec u'_1(1) = \langle 2013\rangle$, implying that $\mathcal{M}^* \prec \mathcal{M}^{*\prime}$.

3 Exhaustive enumeration of k-MOLS

An exhaustive enumeration of k-MOLS of order n may be carried out by the orderly generation of the class representatives of every main class. The pseudo-code of such an enumeration procedure is given as Algorithm 1. A backtracking tree-search is implemented in Algorithm 1 for constructing k-MOLS of order n, one universal at a time, in such a way that, for $i \in \mathbb{Z}_n$ and $m \in \mathbb{Z}_k$, the active nodes on level $i.m$ of the search tree correspond to the lexicographically smallest partial k-MOLS whose Latin squares $\boldsymbol{L}_0, \ldots, \boldsymbol{L}_m$ each contains $i+1$ universals and whose Latin squares $\boldsymbol{L}_{m+1}, \ldots, \boldsymbol{L}_k$ each contains i universals. The inactive nodes in the search tree represent those partial k-MOLS which cannot be completed to a class representative or in

which the partial Latin squares are no longer pairwise orthogonal. On level $i.(k-1)$ of the search tree the universal for the symbol i has been inserted in all the Latin squares $\boldsymbol{L}_0, \ldots, \boldsymbol{L}_{k-1}$ of the partial k-MOLS and the next universal to insert is $u_{i+1}(0)$; as this level marks the completion of the partial k-MOLS up to the symbol i, it is also referred to simply as level i.

Suppose that the partial k-MOLS $\mathcal{P}$ has been constructed on level $i.\ell$ of the search tree, in other words, the next universal to insert into $\mathcal{P}$ is $u_i(\ell+1)$, or $u_{i+1}(0)$ if $\ell = k-1$. Let $U(\mathcal{P})$ be the set of all universals in the partial k-MOLS $\mathcal{P}$, $U(\mathcal{P}_{\ell+1})$ the set of all universals in $\mathcal{P}$, excluding the universals of $\boldsymbol{L}_{\ell+1}$ (the Latin square into which a universal is currently being added) and denote the set of feasible candidate universals by $\mathcal{C}(\mathcal{P})$. The node in the search tree representing $\mathcal{P}$ thus has $|\mathcal{C}(\mathcal{P})|$ children, any number of which may be inactive.

To verify orthogonality in a child $\mathcal{P} \cup c$ of $\mathcal{P}$, for some candidate universal $c \in \mathcal{C}(\mathcal{P})$, it is necessary to confirm that the relative cycle structure of c and every permutation $p \in U(\mathcal{P}_{\ell+1})$ has exactly one fixed point. The following result by Kidd *et al.* [12, Theorem 4.3.2] provides an easy way of determining whether a partial k-MOLS $\mathcal{M}$ is the lexicographically smallest partial k-MOLS in its main class.

Theorem 1 *[12, Theorem 4.3.2] If $\mathcal{M} = (\boldsymbol{L}_0, \ldots, \boldsymbol{L}_{k-1})$ is the lexicographically smallest k-MOLS of order n in its main class, then (a) $u_0(0)$ is the identity permutation, (b) $u_0(1)$ is a cycle structure representative, and (c) the relative cycle structure of two universal permutations $u_i(j), u_\ell(m)$ is not lexicographically smaller than the cycle structure of $u_0(1) \in U(\mathcal{M})$ for all $i, j \in \mathbb{Z}_n$ and $j, m \in \mathbb{Z}_n$.*

According to Theorem 1 (a) and (b) there is a very limited number of feasible zero universals in $\boldsymbol{L}_0$ and $\boldsymbol{L}_1$, and by Theorem 1 (c) no relative cycle structure calculated while verifying orthogonality may be smaller than the cycle structure of $u_0(1)$ if $\mathcal{P} \cup c$ is to be the lexicographically smallest partial k-MOLS in its main class.

If $\mathcal{P} \cup c$ passes this test, then all possible pairs of universals $u_a(j), u_b(m)$ in $\mathcal{P} \cup c$ or its transpose $(\mathcal{P} \cup c)^T$ with a relative cycle structure equal to the cycle structure of $u_0(1)$ are mapped to the pair of universals $u_0(0), u_0(1)$ to form a new partial k-MOLS $(\mathcal{P} \cup c)'$ in the same main class, which is then subjected to a restricted number of row, column and symbol permutations in an attempt to find a lexicographically smaller partial MOLS. More specifically, in order to ensure that the universal $u_0(0)$ in $(\mathcal{P} \cup c)'$ remains unchanged, it is necessary to apply any potential permutation

Algorithm 1: enumerateMOLS($\mathcal{P}$)

input : A partial k-MOLS $\mathcal{P}$
output: All completed class representatives in the subtree rooted at $\mathcal{P}$

```
1  begin
2      if P is complete then
3          if none of the conjugates of P has smaller isotopics then
4              output P as class representative
5              return
6          else
7              return
8      for every candidate universal c do
9          if c preserves orthogonality and is valid by Theorem 1 (c) then
10             if P ∪ c has no smaller isotopic k-MOLS then
11                 enumerateMOLS(P ∪ c)
```

to both the row and column indices. However, for $u_0(1)$ with cycle structure $z_1 z_2^{n_2} \dots z_p^{n_p}$ to be unaffected by the permutations, the set of potential permutations is restricted to only the $\prod_{i=1}^{i \leq p} i^{n_i} n_i!$ permutations that are found by rotating and reordering the cycles of $u_0(1)$. This step of the enumeration process, referred to in line 10 of Algorithm 1, is called the `isSmallest` test. If such a smaller partial MOLS is found, the node representing $\mathcal{P} \cup c$ becomes inactive and the next candidate universal is inspected for insertion into $\mathcal{P}$. Otherwise, a new list of candidate universals are generated for insertion into $\mathcal{P} \cup c$ and the search restarts one level lower down the tree. Whenever there are no more candidate universals to inspect, the search returns to the previous level. For a completed k-MOLS $\mathcal{P}$ on level $n-1$, the mappings and transformations described above are performed on all of the conjugates of $\mathcal{P}$ to confirm that none of these conjugates have a lexicographically smaller isotopic k-MOLS than $\mathcal{P}$.

This enumeration process for 2-MOLS of order 5 is represented in Figure 2 (the same example may be found in [12]). According to Theorem 1, $u_0(0)$ must be the identity permutation and $u_0(1)$ a cycle structure representative, of which there are two possibilities for order 5, namely $z_1 z_2^2$ and $z_1 z_4$ (note that there must be exactly one 1-cycle to ensure orthogonality with the identity permutation). Two partial k-MOLS are said to be in the same *section* of the search tree if the respective $u_0(1)$ universals are the same cycle structure representative; the enumeration of 2-MOLS of order 5 therefore consists of two sections. Where branches become inactive it is indicated that either (a) no candidate universals preserve orthogonality, (b) a lexicographically smaller partial MOLS has been found in the same main class, or (c) a class representative has been found. One 2-MOLS is found in the section of $z_1 z_1^2$ and no structurally different 2-MOLS is found in the section of $z_1 z_4$, although a completed candidate 2-MOLS is uncovered which, upon inspection, is shown to be in the same main class as the first one, but lexicographically larger.

The known results in Table 1 were replicated in a validation attempt and details on the enumeration results for 3-MOLS of order 8 are given in Table 2. The number of active nodes found on every level is identical to that found by Kidd [12], while the serialized runtime has been improved from approximately 36 days to just under 10 days, although this improvement may be partially due to the use of different computing platforms. There are 45 active nodes on level 0 (after all of the zero universals have been inserted and an `isSmallest` test has been performed) and these nodes were given as the starting positions from which all of the subtrees were enumerated. It was found that there are 259 and 1 700 active nodes on level 0 of the search trees for orders 9 and 10, respectively, which may be partitioned into 7 and 8 sections, respectively. Interestingly, the runtime increased from 6 seconds for the enumeration of 3-MOLS of order 7 to just under 10 days for the 3-MOLS of order 8, raising serious concerns over the feasibility of the enumeration of 3-MOLS of order 9 and higher.

Section	Level								Time (s)
	0	1	2	3	4	5	6	7	
$z_1 z_2^2 z_3$	17	12 501 028	1 484 518 094	18 814 494	55	23	22	20	775 321
$z_1 z_2^1 z_5$	14	3 358 273	61 708 802	63 157	97	92	84	17	60 011
$z_1 z_3 z_4$	5	52 059	5 283	1	0	0	0	0	93
$z_1 z_7$	9	37 403	9 079	82	64	53	53	2	111
Total	45	15 948 763	1 546 241 258	318 877 734	216	168	159	39	835 537

Table 2: The number of active nodes in every section and on every level of the search tree for enumerating 3-MOLS of order 8, together with the time in seconds that the enumeration of every section took on a 3.2 GHz processor with 8 Gb of RAM.

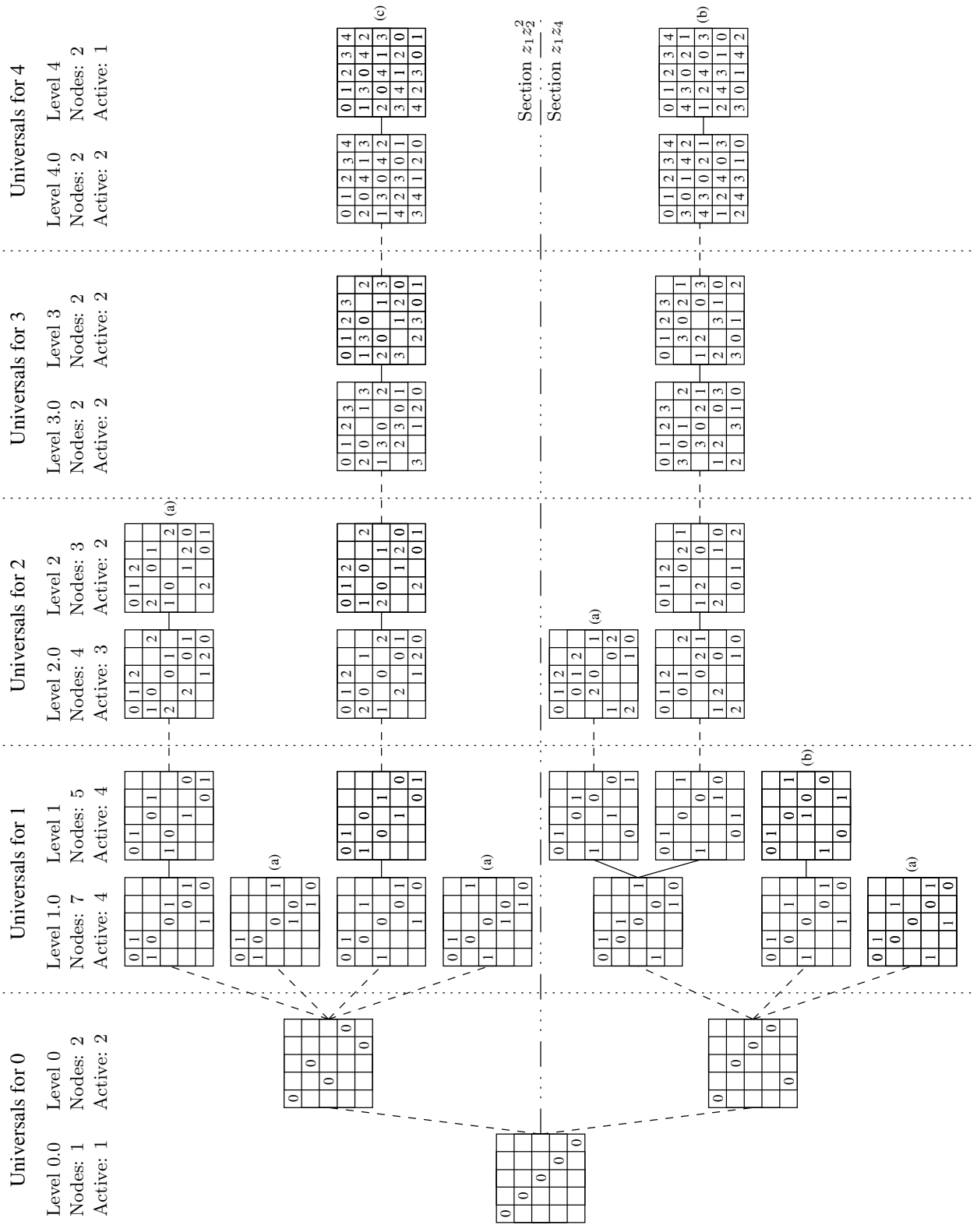


Figure 2: The backtracking enumeration search tree for 2-MOLS of order 5. At every leaf it is either indicated that (a) no candidate universals preserve orthogonality, or that (b) a lexicographically smaller partial MOLS has been found in the same main class, or that (c) a class representative has been found.

4 On the enumerability of larger order search spaces

In order to determine the feasibility of enumerating 3-MOLS of orders 9 and 10, the algorithm was modified so that it only examines MOLS that are isotopic to a partial MOLS $\mathcal{P}$ after universals of the i-th symbol have been inserted into every Latin square in $\mathcal{P}$. Although this increases the total number of branches of the search tree that survive to level i, it decreases the total number of `isSmallest` tests performed during the enumeration, as all branches that would otherwise have been pruned earlier must necessarily have been subjected to at least one `isSmallest` test. Furthermore, the effect on the search tree as a whole is minimised, as the exact same number of branches will pass the `isSmallest` and proceed to the next symbol. The sizes of the subsequent search trees for orders 9 and 10 were approximated by estimating the total number of nodes in the absence of the `isSmallest` test before applying the expected pruning effect of the `isSmallest` test to determine the number of active nodes on every level of the tree. Finally, a small number of nodes from one of these levels were used as starting points for the enumeration algorithm so that the total time it would take to traverse the entire trees could be estimated. The enumeration tree for 3-MOLS of order 8 was also traversed to determine the average number of universals that preserve orthogonality and are valid by Theorem 1 (c), *i.e.* the universals that pass the test on line 9 of Algorithm 1, for partial 3-MOLS on different levels of the search tree.

It was found that that this average number of feasible candidate universals, which corresponds to the number of children of a node representing any partial 3-MOLS on level $i.\ell$ for $\ell \in \mathbb{Z}_{k-1}$, depends sensitively on the cycle structure of $u_0(1)$, but remains largely constant within a given section of the tree. Evidence of this may be seen for the 45 active nodes on level 0 of the enumeration tree for 3-MOLS of order 8 in Figure 3 for the two sets of universals $u_1(j)$ and $u_2(j)$ with $j \in \mathbb{Z}_k$. Notice in the figure, that the average number of feasible candidate solutions decreases with every additional universal in $\mathcal{P}$ as it becomes harder to preserve orthogonality. This regularity in the number of children of a node of the search tree, as well as its sensitive dependence on the cycle structure of $u_1(0)$ was also observed in the search trees for 3-MOLS of orders 7, 9 and 10.

These properties make it possible to estimate the average number of children of any partial 3-MOLS by only examining a very small random selection of partial 3-MOLS that are on the

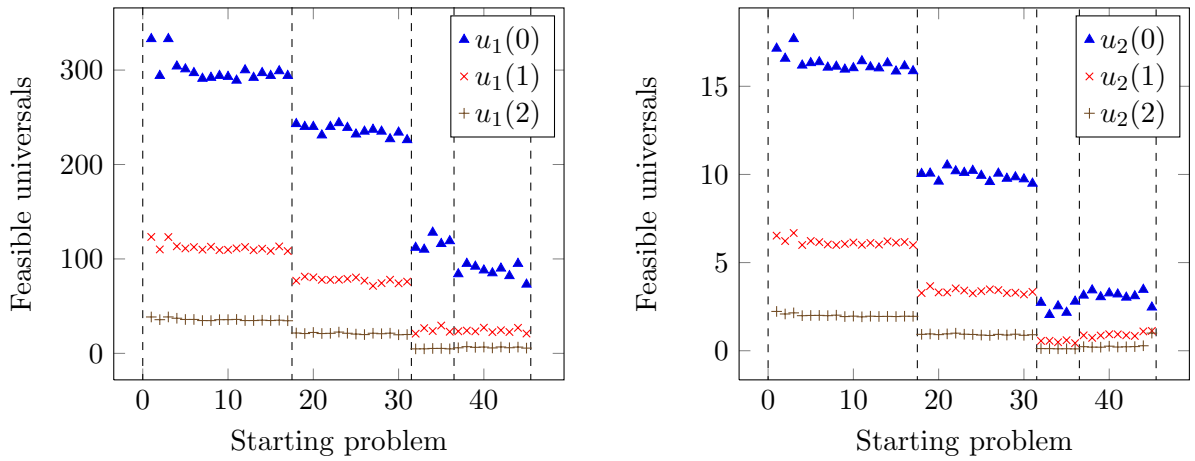


Figure 3: The average number of feasible candidate universals $u_i(j)$ found for $i = 1, 2$ and $j \in \mathbb{Z}_k$ in the enumeration of 3-MOLS of order 8 for each of the 45 partial 3-MOLS which pass the `isSmallest` test on level 0 of the search tree. The dashed lines indicate in which section the starting position resides, *i.e.* whether the permutation $u_0(1)$ in the initial partial 3-MOLS has the cycle structure $z_1 z_2^2 z_3$, $z_1 z_2 z_5$, $z_1 z_3 z_4$ or $z_1 z_7$, in that order.

	Order 8		Order 9	Order 10
	Actual	Estimated	Estimated	Estimated
Level 1	2.61×10^7	2.60×10^7	5.79×10^{10}	2.41×10^{14}
Level 2	4.34×10^9	3.74×10^9	3.39×10^{15}	9.67×10^{21}
Level 3	9.96×10^8	9.31×10^8	2.15×10^{16}	—

Table 3: A comparison of the actual and estimated total number of nodes on levels $0, 1, 2$ and 3 of the search tree for 3-MOLS of order 8, together with similar estimates for orders 9 and 10.

n	6	7	8
Level 0	0.15	0.07	0.03
Level 1	0.55	0.48	0.57
Level 2	0	0.54	0.51

Table 4: The proportions of nodes which pass the `isSmallest` test on levels $0, 1$ and 2 for 3-MOLS of orders 6, 7 and 8.

same level and in the same section of the tree. This process was repeated on every level of the tree in order to estimate the total number of nodes in the search tree for 3-MOLS of orders 8, 9 and 10. This estimate proved to be fairly accurate for order 8, as may be seen in Table 3.

In order to estimate the number of active nodes on levels 1 and 2 of the search tree, the pruning effect of the `isSmallest` test must be applied to these estimated total numbers of nodes on every level of the tree. Let p_i denote the percentage of partial 3-MOLS which pass the `isSmallest` test on level i. The values of p_0, p_1 and p_2 for orders $6, 7$ and 8 may be seen in Table 4. Notice that less than 10% of the nodes on level 0 are active, and that this value is approximately 50% for levels 1 and 2. Based on this evidence, the numbers of active nodes on levels 1 and 2 of the search trees for orders 9 and 10 were estimated for three values of $p = p_1 = p_2$, specifically $p = 0.5$ together with expected over and under estimate values, $p = 0.4$ and $p = 0.6$. Note that the pruning effect is carried forward through the tree, *i.e.* if $p = 0.5$, then 50% of the nodes on level 1 are considered inactive, which implies that half the nodes on level 2 would not have been reached at all so that only 25% of the total number of nodes on level 2 are considered active. For order 9 the number of active nodes of level 1 (*i.e.* the number of partial 3-MOLS with all 0 and 1 universals filled in which pass the `isSmallest` test) is estimated to be between 2.32×10^{10} and 3.47×10^{10}, depending on the value of p, and for order 10 this number grows to approximately 1.21×10^{14}. The remainder of the estimated numbers of active nodes may be found in Table 5.

To gather insight into the potential total runtime of the enumeration algorithm for 3-MOLS of orders 9 and 10, a representative sample of active nodes on level 1 of the respective search trees was used as starting points for Algorithm 1, after which the number of active nodes was multiplied by the weighted average time to completion. To enable comparison between computing systems of different speeds the estimated time to completion is expressed in GHz-days, the number of days that a single 1Ghz processor would take to complete the computation. It is expected that a complete enumeration of 3-MOLS of order 9 would take approximately 5.64×10^8 GHz-days, while for order 10 this is expected to take approximately 1.42×10^{18} GHz-days (these estimates may also be found in Table 5).

5 Conclusion

The serialized estimated enumeration time on a single 3.2 GHz core of 465 219 years for 3-MOLS of order 9, and 1.17×10^{14} years for order 10 is currently beyond the capabilities of most research computing clusters. For example, the high performance cluster, *Rhasatsha*, at Stellenbosch University currently consists of a hundred and thirty six 2.83 Ghz cores, thirty two 2.4 Ghz cores and three hundred and seventy six 2.1 GHz cores for a daily maximum throughput of approximately 1 250 GHz-days. The performance of this cluster is dwarfed by, for example, the *Great Internet Mersenne Prime Search* (GIMPS), a distributed computing project which makes use of volunteers' computing power to find extremely large prime numbers

	Order 9			Order 10		
p	0.4	0.5	0.6	0.4	0.5	0.6
Level 1	2.32×10^{10}	2.89×10^{10}	3.47×10^{10}	9.65×10^{13}	1.21×10^{14}	1.44×10^{14}
Level 2	5.43×10^{14}	8.48×10^{14}	1.22×10^{15}	1.55×10^{21}	2.42×10^{21}	3.48×10^{21}
Level 3	1.37×10^{15}	2.68×10^{15}	4.64×10^{15}	—	—	—
Time	4.51×10^{8}	5.64×10^{8}	6.77×10^{8}	9.11×10^{17}	1.42×10^{18}	2.05×10^{18}

Table 5: The estimated total number of active nodes on different levels of the search tree for the enumeration of 3-MOLS of orders 9 and 10, as well as the estimated time (measured in GHZ-days) that the enumeration would take.

[17] and Seti@Home, a distributed project examining large datasets for signs of extraterrestrial intelligence [2]. GIMPS has an average daily throughput of approximately 100 000 GHz-days [17], while SETI@Home averages 362 000 GHz-days daily [2]. If the enumeration of k-MOLS were to take place with the computing power that is available to these distributed projects, the enumeration of 3-MOLS of order 9 would take approximately 15.5 years (at 100 000 GHz-days daily) and it would be possible to answer the celebrated question of the existence of 3-MOLS of order 10 in approximately 3.9×10^{10} years. These estimates ignore the fact that the computing power of desktop computers, which are indispensable to distributed computing projects, have been estimated to double every 1.5 years over the last five decades [14, 18]. Assuming that this rate of growth continues, a distributed computing project of the scale of GIMPS may enumerate the 3-MOLS of order 9 in as little as 4.5 years, and the 3-MOLS of order 10 in approximately 51 years.

The enumeration of 3-MOLS of orders 9 and 10 therefore seems to be feasible as part of a long-term distributed, volunteer computing project. Moreover, the enumeration attempt would benefit greatly from a significant technical breakthrough in computing power or an important theoretical breakthrough (such as the design of a very effective pruning rule for the search tree or a speed-up of the `isSmallest` test). To resolve the question of the existence of 3-MOLS of order 10, however, it is only necessary to find a single 3-MOLS, making the estimated enumeration time a worst-case scenario that will only be reached if no 3-MOLS of order 10 exists.

References

[1] Abdel-Ghaffar KAS, 1996. *On the number of mutually orthogonal partial latin squares*, Ars Combinatorica **42**, pp. 259–286.

[2] Anderson DP, 2013. *SETI@Home*, [Online], [Cited 2013, June 4th], Available from: `http://setiathome.berkeley.edu/`.

[3] Bachet CG, 1624. *Problèmes plaisants & délectables: Qui se font par les nombres*, Pierre Rigaud & Associates, Lyons.

[4] Bose RC, Shrikhande SS & Parker ET, 1960. *Further results on the construction of mutually orthogonal Latin squares and the falsity of Euler's conjecture*, Canadian Journal of Mathematics **12**, pp. 189–203.

[5] Burger AP, Kidd MP & van Vuuren JH, 2010. *Enumerasie van self-ortogonale Latynse vierkante van orde 10*, Litnet Akademies (Natuurwetenskappe) **1(1)**, pp. 1–22.

[6] Colbourn CJ & Dinitz JH, 2006. *Handbook of combinatorial designs*, Chapman & Hall/CRC, Boca Raton (FL).

[7] Dukes P & Howard L, 2012. *Group divisible designs in MOLS of order ten*, Designs, Codes and Cryptography, pp. 1–9, [Online], [Cited 2013, June 4th], Available from: `http://link.springer.com/article/10.1007/s10623-012-9729-8`

[8] Euler, L, 1782. *Recherches sur une nouvelle espéce de quarrés magiques*, Verhandelingen uitgegeven door het zeeuwsch Genootschap der Wetenschappen te Vlissingen **9**, pp. 85–239.

[9] Fisher RA, 1925. *Statistical methods for research workers*, Oliver & Boyd, Edinburgh.

[10] Fisher RA, 1935. *The design of experiments*, Oliver & Boyd, Edinburgh.

[11] Keedwell AD, 2000. *Designing tournaments with the aid of Latin squares: A presentation of old and new results*, Utilitas Mathematica **58**, pp. 65–86.

[12] Kidd MP, 2012. *On the existence and enumeration of sets of two or three mutually orthogonal Latin squares with application to sports tournament scheduling*, PhD Dissertation, Stellenbosch University, Stellenbosch.

[13] Kidd MP, 2010. *A tabu-search for minimising the carry-over effects value of a round-robin tournament*, ORiON **26(2)**, pp. 125–141.

[14] Koomey JG, Berard S, Sanchez M, Wong H, 2011. *Implications of historical trends in the electrical efficiency of computing*, IEEE Annals of the History of Computing **33(3)**, pp. 46–54.

[15] Lam CWH, Thiel L & Swiercz S, 1989. *The non-existence of finite projective planes of order 10*, Canadian Journal of Mathematics **41(6)**, pp. 1117–1123.

[16] Laywine CF & Mullen GL, 1998. *Discrete mathematics using Latin squares*, Wiley, New York (NY).

[17] Mersenne Research, Inc., 2013. *Great Internet Mersenne Prime Search*, [Online], [Cited 2013, June 4$^{\text{th}}$], Available from: `http://www.mersenne.org/`.

[18] Nordhaus WD, 2007. *Two centuries of productivity growth in computing*, Journal of Economic History **67(1)**, pp. 128–159.

[19] Owens PJ & Preece DA, 1995. *Complete sets of pairwise orthogonal Latin squares of order 9*, Journal of Combinatorial Mathematics and Combinatorial Computing **18**, pp. 83–96.

[20] Robinson DF, 1981. *Constructing an annual round-robin tournament played on neutral grounds*, Mathematics Chronicle **10**, pp. 73–82.

Proceedings of the 2013 ORSSA Annual Conference
pp. 50–60
www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf

ORSSA Proceedings
ISBN
xxx-xxxxx-xxxxx-x

Influencing the Future: Open Access Horizon Scanning in Africa?

Isabel Meyer[1] Mario Marais[2]

Abstract

Horizon Scanning is a Futures methodology that aims to identify weak signals in the environment, and use these to inform strategic planning. Horizon Scanning is a people-driven process, which may or may not include the use of expert opinion. The outcome is inevitably influenced by the world view and priorities of the scanners. A number or organizations have been performing Horizon Scanning in Africa over the recent past, and the extent to which such work can identify trends and influence strategic thinking in a developing context, is of interest.

Successful, large scale Horizon Scanning projects have been undertaken in the developed world, and have in most cases been driven by specific organisational or government goals and funding. With respect to poverty and development, the question is posed whether or not it is possible to do Horizon Scanning in an "open access" fashion, i.e. to produce information that is independent of specific goals, available to all for free, and of sufficient value to result in the broad uptake of Futures thinking and the ubiquitous use of Futures products to influence decision making.

By its very nature, an assessment of the quality of any Futures work is difficult. This paper reviews some of the recent African Horizon Scans, at a process as well as a content level. It considers the strategic use of Horizon Scanning information in developing contexts, and the potential thereof to influence decision making for poverty reduction and development. The viability of open access Horizon Scanning in developing countries is also considered.

Key words: Horizon Scanning, Futures methodologies, Strategic planning, Open access

[1] Independent Management Consultant, South Africa, PO Box 331, Wapadrand 0050, email: isabelmeyer@mweb.co.za

[2] Corresponding author: CSIR, South Africa, P O Box 395, Pretoria, 001, email:mmarais@csir.co.za

1 Introduction

Horizon Scanning (HS) is a Futures methodology that is aimed at identifying trends and influencing strategic thinking. HS has been defined by Könnölä et al. [10] as "a creative process of collective sense-making by way of collecting and synthesising observations that hold the potential for the formulation of pertinent future developments and the derivation of actionable implications on decision–making" (p. 223). When decision-makers are faced with the future, they are dealing with a problem that Ackoff calls [1] "dynamic situations that consist of complex systems of changing problems that interact with each other", also defined as messes (p.99). The challenge, as framed by Ackoff in 1979, is for Operational Research (OR) to move from dealing with isolated problems to systems of messes and to participate in interdisciplinary approaches to treating messes [1]. The call for OR to deal with the complexities of the "real world" has endured, as a recent address by Royston shows [11]. From the perspective of dealing with messes, HS and other Futures methodologies is the concern of OR practice. A further positioning of this problem is in terms of CAPSTONE OR, which classifies OR (amongst others) as dealing with complex, dynamic, fuzzy and huge systems to support strategic scenario development [5]. According to Habeggar [6], the concept of HS is ill-defined, and refers in the narrow sense to a *policy tool* that systematically gathers a broad range of information about emerging issues and trends in an organization's political, social, technological or economic environment. HS has the potential to engage a diversity of role players in strategic conversations about the future, while at the same time making a difference in a developing context. This requires that an open access model of HS be considered.

A number of large HS projects have been executed in the international domain, and have demonstrated the potential to have an impact on a large variety of aspects. However, projects are almost invariably executed within a specific funding environment, and with specific policy goals. Recent HS and trend monitoring activities in Africa include the work that has been done since 2009 as part of a global *Searchlight* project of the Rockefeller Foundation (RF) [13]. This work comprised four different African Horizon Scans. The Searchlight scans differ from conventional HS in the sense that it is not guided by specific policy goals. The scanning agencies were left to define their own scanning agendas. The work is considered to be unique, since the output of the scans was used by the RF to guide the allocation of funds for development initiatives.

The question is posed as to whether or not it is possible to do HS in an exploratory fashion [2], in order to accommodate a variety of agendas, and to provide "open access" to the outcome. Information would have to be produced that is independent of specific goals, that is accessible at zero cost to a wide audience, and that results in a broad uptake of Futures thinking and its ubiquitous use for decision making.

We use "open access" in this context to refer to the variety (or ecosystem) of open access products, as well as to the characteristics of open access. This paper is concerned with the above question, based upon an assessment of the content that was produced by the four African scans. It introduces HS as a Futures methodology, outlines the research methodology and discusses the content assessment of the African scans. Finally, the viability of and strategies for exploratory open access HS are discussed.

2 Horizon Scanning as a Futures Methodology

The role of HS within the generic life cycle of a portfolio of Futures projects can be described as follows: Define Futures question to answer → Conduct HS → Disseminate results to inform portfolio of Futures projects → Generate outputs → Initiate actions to influence future (policy, strategy, projects, etc.). This presumes that there are specific questions that need to be answered when using HS as part of a portfolio of Futures projects. In many cases, HS is done as a standalone project.

A study of international large-scale cross-cutting Futures initiatives revealed that in order have impact, HS needs to be embedded into a comprehensive Foresight process that builds networks across professional communities, enables broad-based social learning processes and feeds the results into policy processes [6, 7]. Finland has focused on embedding Futures thinking in all sectors of its society [17].

The potential impact of HS projects needs to be read in the context of its role in the entire Foresight process. Habegger [7] outlines this role in Figure 1 below:

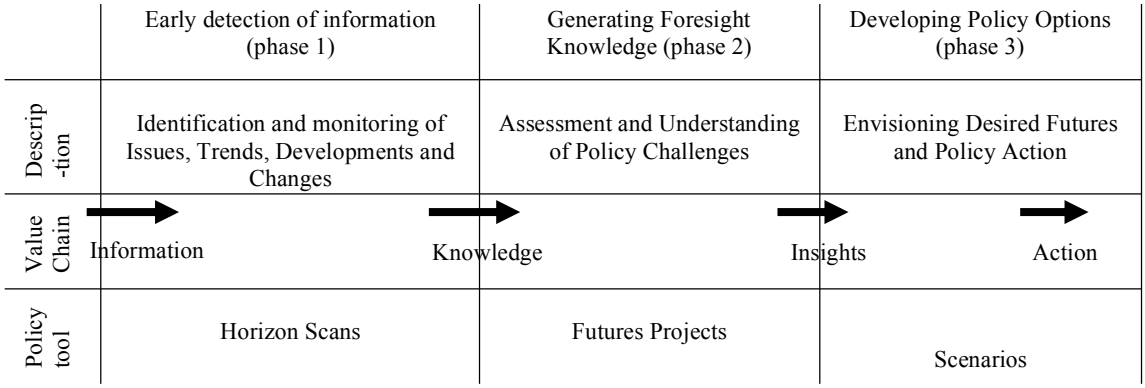


Figure 1: Three phases of a Foresight Process [7]

It is clear that HS is the first step in a set of processes that ends with envisioning of possible or desirable futures (scenarios) and taking action in order to "create" the desired future(s). In this case the focus was on desirable futures. The RF initiated the global set of HS projects to feed its Global Searchlight Project. The aim was to identify new policies, technologies and strategies for meaningful change in the lives of poor, vulnerable populations. The output from HS would thus be taken through the entire chain from information to action.

The RF's aim to use HS outputs is fairly unique, as was highlighted at a 2013 UK Futures Symposium on the difficulty of stimulating the use of HS in decision-making [3, 9]. The RF was open to innovation and left it to the contractors to select their own HS approaches. In Africa there were four grantees: Southern Africa was covered by the SA Node of the Millennium Project, West Africa by the Africa Centre for Economic Transformation (ACET) and the Center for Democracy and Development (CDD), and the Greater Horn and Eastern Africa by the Society for International Development (SID).

3 Research methodology

This study evaluated the African Scans in terms of process as well as content, and assessed the potential future impact of HS in the various regions. To this end, a framework was designed, based upon the assumption that some kind of impact is desirable (see Figure 2). HS can also be done without defined impacts in mind. It could provide freely available base data, or be used to create general awareness of Futures thinking.

The view of Habbeger [6] was adopted, who describes HS as having two distinct functions, namely providing information, and assisting policy development. In line with these two functions, the HS process is broadly defined as having three steps concerned with information or content generation (namely signal selection, signal processing and content generation), and a fourth step concerned with policy development (namely influencing impact). These four steps are mapped to the three major components of the evaluation framework, namely: Evaluate process, Assess content, and Evaluate strategy.

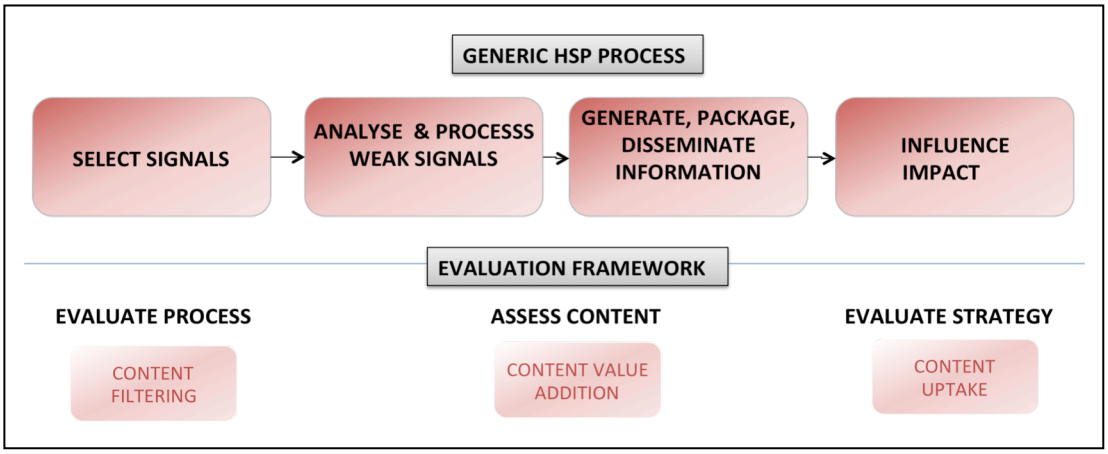


Figure 2: Generic Horizon Scanning mapped to the assessment framework

The four steps of the framework are described in more detail below.

Step 1: select signals (content filtering)

The different types of signals (weak signals, wild cards, emerging issues, trends) and the variety of information sources that were consulted, was considered [2, 4, 10]. Selection processes, as well as attributes of the information sources used, were evaluated.

Steps 2 and 3: Analyse & process weak signals; Generate, package, disseminate information (content value addition)

Content produced by the HS process provides the lever for influencing policy, and needs to be relevant, appropriate, and potentially impactful. The following content evaluation framework

was defined: Firstly, **a subset to evaluate** was identified. A qualitative analysis of the data was done, loosely based on the principles of grounded theory [8]. For the content that was selected, **signals were identified** [2, 4, 15]. The application of an ontology of concepts is useful in discovering data from large datasets, that comprise a diversity of perspectives. A PESTLE analysis provides a comprehensive view on the macro-environment, and is similar to categories used in earlier analyses of Horison Scanning outputs [14]. The above was therefore **categorised** according to PESTLE (Political, Economic, Sociological, Technological, Legal and Environmental), and was then evaluated for **coverage** (of categories, and of geography). The content was also evaluated for **quality**, by considering client feedback, and by comparing it with international HS [15]. The **value addition** of content was evaluated in terms of the extent of analysis and the development of implications.

Step 4: Influence impact (content uptake)

This step was concerned with the extent to which policy makers included outputs in decision making. It depends on the extent to which HS output relates to the objectives of the specific Futures project.

4 The content of the African Horizon Scans

This section presents various views on the content that was produced by the HS projects. Newsletters from the last year of production (i.e. 2012) were analysed for all four HS projects. The SA HS project used Undirected viewing, which consists of reading / examining a wide variety of publications and information sources for the purpose of being informed. It also used Directed viewing, which consists of responding to this information in terms of assessing its relevance to a specific organization. The process was aimed at sensing rather than making sense from information [9, 12]. Undirected viewing relates to exploratory scanning. The analysis outlined below provides a picture of the focus that emerged from the sensing process.

The purpose of the four regional scans was not the same, and information was presented differently by each scan. This was because the RF allowed grantees to select their own approaches. A direct comparison of content of the various scans is therefore not appropriate. However, a comparative analysis between the scans assisted in answering the question as to the potential future use of HS in developing contexts.

The *Southern African (SA) scan* reports on different topics, referenced per item, with an interpretation added to selected individual items [12]. The *ACET (East African - EA)* as well as the *SID (West African - WA) scans* produced thematic newsletters with all items in the newsletter related to the theme. *ACET* includes an introductory editorial, while the *SID* newsletter starts with an introduction, and concludes with insights, foresights, and intervention opportunities for the RF. The *CDD scan (WA*) delivers thematic reports, with some unrelated items. It includes an editorial, and is concluded with an expert interview.

4.1 Comparative content analysis

The content produced by the different Horizon Scans was analysed in terms of the diversity of sources that was used, the geographic balance thereof, and the various categories of information that was covered.

4.1.1 Diversity of sources

The SA scanning process utilised a push system of information, and used many different types of sources such as RSS feeds, Google trend alerts, Twitter, following a selected individual's work, as well as data from existing scanning providers (e.g. the Institute for Futures Research at the University of Stellenbosch) [9]. The other regional scans (excluding SA) utilised information differently. Organisational reports and conference proceedings were used in between 23% and 38% of cases compared to 7% for SA. The CDD scan had a relatively high utilisation (10%) of expert interviews. Aggregate sources of information such as news sites dominated for all scans, while scientific journals were not significantly utilised. The SA scan was non-thematic, and made more repetitive use of sources as compared to the thematic ACET scan. The use of expert opinion is highest in the thematic CDD scan.

4.1.2 Geographical balance of sources

For each HS project, the percentage of reports that relates to a specific geography was calculated, as indicated in Table 1.

Table 1: Percentage of reports pertaining to a specific geography, by each scan

	Geographic regions reported on by the scan			
Scan	**Southern Africa**	**East Africa**	**West Africa**	**All Africa**
SA (South Africa)	65	5	4	26
CDD (West Africa)	3	7	90	1
ACET (West Africa)	4	3	93	0
SID (East Africa)	1	94	1	4

This distribution reflects the fact that regions were requested to provide scanning information for their region. The notable exception is the SA scan's greater emphasis on aspects relating to the entire African continent. This contextualisation of information in a broader context could provide policy makers with more information within which to make policy decisions.

4.1.3 Coverage

The coverage of topics were analysed in three ways: a PESTLE categorisation of topics, a bottom-up categorisation based on a grounded-theory approach and a categorisation according to the different types of Futures concepts that were covered.

PESTLE Categorisation - The SA scan maintained a relatively balanced view across topics, with the exception of legal aspects, which are less reported on. In the other African scans a relatively low representation of political, legal and environmental issues as well as a relatively high representation of reporting on economic issues in the SID and ACET scans is noticeable. No reporting is done on pure technological issues in the SID scan (possibly related to the thematic nature of the scan).

Bottom-up categorisation (grounded theory) – Done for the SA scan only. Ecosystems, Connectivity and Health Care contain the most topics (14 – 18), possibly reflecting the dominant development discourses. The next level (8-10 topics) includes Agriculture, Education, Donors & development funding, Governance, Personal finance and Innovation. Skills and Employment (2 topics) is low, relative to the government's unemployment focus. Population dynamics seems underrepresented, given the huge urbanization trend in Africa. However, Rural-urban migration is reflected. Information and Communications Technology (ICT) is represented in a few categories. The categories that have not been covered in the various scans are of interest. Basic services (e.g. Water and Sanitation) and Entrepreneurial development seems to be under-represented. Connectivity, Information and Knowledge are mostly about ICT for Development (also seen regularly in the popular media).

Futures concepts - Weak signals are expected to dominate HS, and was high in all of the scans except ACET (14%) - likely due its thematic nature. The SA scan's high percentage (56%) of weak signals is consistent with its Undirected scanning approach (see Figure 3 below).

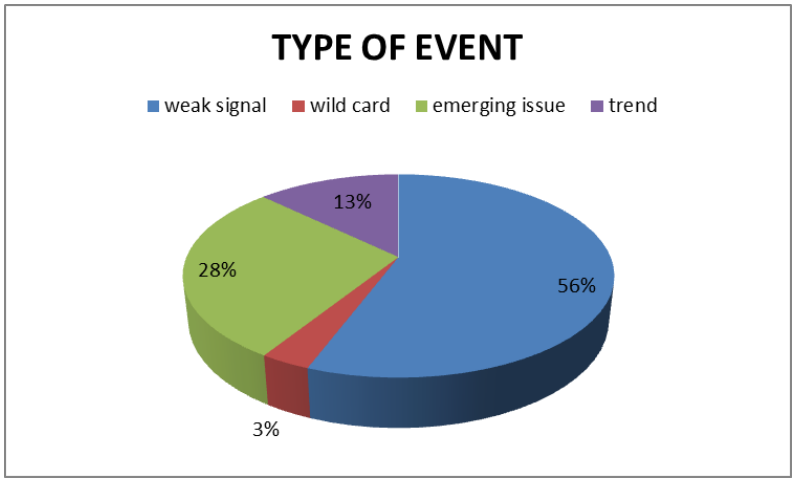


Figure 3: Percentage of each futures concept that was generated by the Southern African scan

The high percentage of trends in the ACET and SID scans is probably due to their thematic nature. The low percentages of trends of the CDD (14%) and SA (13%) scans are possibly due to their mixed approaches: the inclusion of themed and other topics for CDD, and combined Undirected and Directed scanning for the SA scan. One would have expected a higher trend ratio in CDD, given the use of experts. There is generally a low prevalence of wild cards in all of the scans, which can be expected since wild cards signify (scarce) disruptive events. The thematic nature of the WA scans probably explains its absence of wild cards. Emerging issues are around 30% for all of the scans except for the theme-driven ACET scan.

4.1.4 Summary of findings

Legal aspects are not well represented across all scans. Sources of the SA scan are dominated by news aggregation sites, and are low on expert opinion. The CDD scan had a relatively high utilisation of experts, probably due to its thematic nature. Grey literature (e.g. organisational reports, working papers) was low in the SA scan, but fairly well represented in the other scans.

The SA scan reflects a broader awareness of Africa-wide issues. It may lead to shared learning, as well as improved quality of policy decisions. In the SA scan, a wide range of concepts related to poverty are covered, but key aspects such as basic services are not included, and housing is underrepresented. Ecosystems and natural resources are surprisingly well represented, as is ICT. An appropriate framework, such as the 15 Global Challenges of The Millennium Project, could be used to guide HS [16].

Value addition by the WA and EA scans included editorials, expert interviews and Futures information such as insights and focus areas. The SA scan added interpretive information as the process evolved. The diversity of approaches and results probably reflect the interests and concerns of the scanning organisations in response to the developmental contexts in which they find themselves. The focus of the EA and WA scans on local sources may reflect that local influences on development are considered to be more important. The SID (EA) scan was most explicit in terms of extrapolating from the scanning information, interpreting information in a Futures context and directing attention towards interventions.

Based upon this research, the following general recommendations were developed: Fit process to purpose, fit the balance of data sources to the purpose (diversity of types of sources, including a considered and balanced geographical focus) and match coverage of topics with key areas of poverty reduction.

5 Strategies for Open Access Horizon Scanning

Achieving open access HS in developing contexts has three implications. Firstly, the results cover topics relevant to a diversity of development actors. Secondly, results are freely available, and thirdly, a strategy is in place to create awareness and facilitate adoption of Futures thinking as a part of decision making.

In order to be inclusive and accommodate a diversity of role players, it means that participation in the scanning process as well as in the use of results thereof is obtained. The bottom-up exploratory approach referred to by Amanatidou *et al.* [2] implies that results from scanning communities, each with a specific area of interest (e.g. health), are analysed in order to cover all relevant domains (e.g. healthcare).

Both issue-based and exploratory scanning approaches are required in order to have broad reach and provide open access. Economic viability and the need to stimulate interest may require customised products that deliver value for paying clients. A diversity of income streams is required to support the production of free, open access scanning information in order to lower the barriers to use and adoption.

Four different approaches to Open Access HS are proposed. Firstly, undirected scanning, not informed by specific objectives and without value addition in terms of interpretation of information; secondly, undirected scanning, not informed by specific objectives but with interpretation of information; thirdly directed scanning (modular), where scanning is customised for different market segments; and lastly directed scanning, commissioned by specific decision makers but where results are made freely available.

Any environment will have a diversity of actors (government, NGOs, business) with value chains that have different characteristics and that require different HS strategies as part the full

Foresight Value Chain (information to knowledge, insight and action). Execution of the full Value Chain is an intensive, lengthy and costly exercise and quick impact is difficult to achieve. In order to affect a broad uptake, a different strategy is required. The extent of product customisation is likely to affect its value for the client.

These parameters provide a framework within which market segments can be described:

FOCUS	LIMITED	LARGE
SPECIFIC	**Customised products for limited client base Limited diversification**	**Customised products for individual customers**
BROAD	**Generic products or Training**	**More specialised products for greater diversity of groups**

CAPACITY

Figure 4: Customisation of products relative to capacity and focus of the scanning agency

The above diagram describes the nature of the product that is appropriate for different segments, and that is achievable given the available capacity of the scanning agency or network. Given the limited capacity and broad focus of the African agencies, these scans fall within the bottom left quadrant. The content of the EA and WA scans do show a slightly narrower focus due the presence of themes and expert opinion.

Given the objective of Open Access HS of reaching a wide audience, and assuming funding constraints and therefore limited resources, five different dissemination and adoption strategies could be pursued. Firstly, broadcasting (low cost, wide audience, low value for client, unpredictable adoption). Secondly, train a wide diversity of clients (low cost, wide audience, higher value for client, targeted and more predictable adoption). Thirdly, customise cheaply (medium cost, narrow but diverse audience, higher value for client, higher likelihood of adoption). Fourthly, select key clients in a limited number of sectors (medium to high cost, narrow audience, high value for client, influential early adopters); and lastly, build the scanning agency or network's capacity and profile via a national initiative driven by a national actor. It follows that Open Access Horizon Scanning is possible, provided that an implementation strategy is followed that balances the trade-off between direct broad uptake and generating indirect influence.

6 Conclusion

By its very nature, an assessment of the quality of any Futures work is difficult. This paper reviewed some of the recent African Horizon Scans, at a process as well as a content level. With respect to poverty and development, the question is posed whether it is possible to do HS in an "Open Access" fashion, i.e. to produce information that covers a wide range of interests, and

that are at the same time available for free in order to stimulate the broad uptake and ubiquitous use of Futures thinking and Futures products.

Given the objective of reaching different types of audiences, and assuming limited resources, a number of strategies could be pursued to ensure the viability of Open Access Horizon Scanning.

In dealing with a wide range of clients and differing scanning interests, particular effort needs to be made to clearly define the objective of the scan(s), the target audience and the extent to which value is added to the output. Enabling a network of scanners that know their target audience well may be the way to deal with this complexity. In the case of Finland a concerted strategy is in place to involve a wide range of actors in all sectors of society in an ongoing national conversation about the future. This could be the "holy grail" of Futures thinking, especially in the many deeply unequal developmental economies in Africa where many voices are not heard.

Acknowledgements The SA Node of the Millennium Project for sponsoring this research, and thanks to Tanja Hichert, Bernd Oellermann, Geci Karuri-Sebina, Rasigan Maharaj and Bob Day for excellent cooperation. Many thanks to Olli Hietanen of the Finland Futures Research Centre who provided valuable insights as well as access to information resources.

References

[1] Ackoff, R.L., 1979. The Future of Operational Research is Past. *The Journal of the Operational Research Society* 30: 93-104.

[2] Amanatidou, E., Butter, M., Carabias. T.K., Könnölä, T., Leis, M., Saritas, O., Schaper-Rinkel, P., and van Rij, V., 2012. On concepts and methods in horizon scanning: Lessons from initiating policy dialogues on emerging issues. *Science and Public Policy* 39:208-221.

[3] Defence Science and Technology Laboratory (dstl)., 2013. Bridging the gap: How can horizon scanning and strategic Futures be used to inform policy? Third Horizon Scanning and Futures Symposium, held on 15 and 16 January 2013 at the Defence Academy in Shrivenham, UK. https://www.dstl.gov.uk/bridgingthegapsymposium. [accessed June 2013].

[4] European Union iKnow Project. n.d. *What is a weak signal?* Available online at: http://wiwe.iknowFutures.eu/what-is-a-weak-signal/. [accessed June 2013].

[5] Fortuin, L., Van Beek, P. and Van Wassenhove, L., 1996. *OR at work:Practical experiences of Operational Research.* London: Taylor & Francis.

[6] Habegger, B., 2009. *Horizon Scanning in Government: Concepts, Country Experiences, and Models for Switzerland.* Centre for Security Studies, ETH, Zurich. Available online at: http://www.isn.ethz.ch. [accessed December 2012].

[7] Habegger, B., 2010. Strategic Foresight in public policy: Reviewing the experiences of the UK, Singapore, and the Netherlands. *Futures* 42:49-58.

[8] Heath, H., and Cowley, S., 2004. Developing a Grounded Theory Approach: a Comparison of Glaser and Strauss. *International Journal of Nursing Studies* 41:141-150.

[9] Hichert, T., 2013. Personal Communication. (11 Jan 2013).

[10] Könnölä, T., Salo, A., Cagnin, C., Vicente Carabias, V., and Vilkkumaa, E., 2012. Facing the future: Scanning, synthesizing and sense-making in horizon scanning. *Science and Public Policy* 39: 222-231.

[11] Royston, G., 2013. Operational Research for the Real World: big questions from a small island. *The Journal of the Operational Research Society* 64: 793-804.

[12] SA Millennium Node, n. d. Horizon Scanning Proposal to the Rockefeller Foundation.

[13] SA Millennium Node., 2011. Horizon Scanning Project Annual Report, Jan 2011.

[14] Sucharita Gopal., 2012. Global synthesis of Searchlight reports using knowledge discovery and visualization. *Foresight* 14(6): 468 – 488.

[15] Talwar, R., and Schultz, W., 2005. *Baseline Scanning Project: Executive Summary.* DEFRA Strategy and Sustainable Development Directorate, DEFRA.

[16] The Millennium Project., 2013. Website.
http://www.millenniumproject.org/millennium/challenges.html. [accessed June 2013].

[17] Tiihonen, P., 2011. *Revamping the work of the Committee for the Future Publication of the Committee for the Future* 7/2011. Available online at:
http://www.eduskunta.fi/triphome/bin/thw.cgi/trip?${APPL}=erekj&${BASE}=erekj&${THWIDS}=0.5/1361734445_46666&${TRIPPIFE}=PDF.pdf. [accessed February 2013].

Proceedings of the 2013 ORSSA Annual Conference
pp. 61–70
www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf

ORSSA Proceedings
ISBN
xxx-xxxxx-xxxxx-x

Investigation of a Hybrid Renewable Energy System to meet South Africa's Baseload Energy Demand using a Multi-objective PBIL Optimisation Algorithm

Nina Uys[1] Tanya Lane-Visser[2]

Abstract

South Africa's current baseload energy demand is met by electricity produced from coal and nuclear fuel. The demand for electricity is growing and the concern is that this dependence on fossil fuels will have an irreversible adverse environmental impact. Introducing renewable energy sources into the energy system can address this issue. However, renewable energy technologies are intermittent in nature because of their dependence on the climate and their limited storage capability. It is also due to this fluctuating supply that renewable energy technologies have traditionally only been used for peaking, and not baseline, power demands. Combining various technologies in an integrated system though, can possibly result in stable supply. In this study, the multi-objective optimisation of an integrated, large-scale hybrid renewable energy system replacing some of the traditional fossil fuel baseload power stations in an economically feasible way is investigated. The aim is thus to minimise the dependency on traditional fossil-fueled baseload stations at minimal cost. Using the data from national feasibility studies and climate data, the electricity supply potential of a combination of wind, solar photovoltaic, concentrated solar power, pumped storage and hydro power sites is modelled and measured against the national mid-winter demand curve, when national electricity demand is at its highest. To achieve optimisation and conversion, the Population Based Incremental Learning algorithm is used. The Pareto optimal solution set shows the remaining fossil-fuel requirement for a given expenditure. This model can be expanded to include factors such as grid stability and seasonal climate fluctuations in order to aid national renewable energy policy making.

[1]Nina Uys, Department of Industrial Engineering, University of Stellenbosch, Private Bag X1, Matieland, 7602, email: ninauys@gmail.com
[2]Tanya Lane-Visser, Department of Industrial Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602, e-mail: tanyav@sun.ac.za

Key words: PBIL, Renewable Energy, Hybrid Systems, Pareto Front.

1 Introduction

South Africa's current baseload energy demand is met by electricity produced from coal and nuclear fuel. The demand for electricity is growing and the concern is that the domestic supply of these fuels will be depleted before the turn of the century and that this dependence on fossil fuels will have an irreversible adverse environmental impact [1].

Renewable energy technologies have the potential to address these concerns. However, the technologies are mostly intermittent in nature because of their dependence on the climate and their limited storage capability [2]. It is due to this fluctuating supply that renewable energy technologies have traditionally only been used for peaking, and not baseline, electricity demands.

Erdinc & Uzunoglo [3] argue that a combination of various renewable energy sources, thus a hybrid energy system, can meet energy demands in a reliable, economical and environmentally friendly manner. Kempton [2] supports this notion that different energy solutions integrated into the energy system can help balance the daily demand and supply of electricity.

In this study, the possibility of an integrated, large-scale hybrid renewable energy system replacing some of the traditional fossil fuel baseload power stations in South Africa is investigated.

This paper is organised as follows: firstly, the literature on hybrid systems and South Africa's renewable energy efforts are described. Next, the study purpose and scope are stated. Then, the population based incremental learning (PBIL) algorithm and multi-objective optimisation methodology is explained and the Pareto optimal results from the model are shown. Lastly, future research recommendations are made and conclusions are discussed.

2 Literature Study

2.1 Hybrid Energy Systems Optimisation

As mentioned, there exists a mismatch between electricity demand and the available supply of electricity from renewable energy sources. For example, South Africa's energy demand peaks in the evening between 18:00 and 21:00 [4]. A solar photovoltaic (PV) plant supplying power to the grid would however have a maximum output in the middle of the day. In order to address this mismatch, careful analysis has to be done in choosing the various renewable energy solutions in order to match energy demand cycles [5].

Supply availability is however not the only consideration. In their literature study on optimisation studies surrounding renewable energies, Baños et al [6] also identified safety, capital and maintenance cost, pollutant emissions from manufacturing processes, availability of renewable sources, efficiency of energy conversion, land requirements, social impacts and water consumption as factors for consideration.

Considering the large number of variables and the applicable mathematical models, traditional linear programming may not be able to account for all of these factors. Due to this complexity, many authors propose using meta-heuristics to solve the problem of designing hybrid energy solutions [7].

2.2 Renewable Energy Technologies

In order to aid in the management of electricity peaking energy demands and to address the country's fossil-fuel dependency, the South African Department of Energy (DoE) initiated the Independent Power Producer (IPP) Procurement Program [8]. This initiative allows for private bidders to contribute electricity generated through renewable technologies to the national energy grid. The grid is owned and operated by the state-owned utility, and the utility would buy any supplied electricity at the bid rate. Short descriptions of the qualifying technologies are provided below:

- Solar photovoltaics (PV) are made from semi-conductor materials that due to a photo-electric effect release electrons when exposed to solar radiation [9]. This effect is most effective when solar radiation is at its highest around midday, and the supply curve is roughly parabolic peaking around noon on clear days [10]. The CSIR, Eskom and the Department of Minerals and Energy have mapped the annual solar radiation based on ground measurements for the country [10].
- Concentrated Solar Power (CSP) has various degrees of commercial availability. Eskom is currently constructing a CSP plant in Upington that will use central receiver technology. Large mirrors, or heliostats, focus thermal energy on a central receiver, where a molten salt mixture is used to heat water to steam that drive a conventional steam turbine. Due to the thermal storage capability of the molten salt, this renewable technology has the ability to generate electricity when its primary resource, *i.e.* the sun, is unavailable. This technology is however not as commercially established as PV, and proof of concept plants are still being constructed world-wide. [11]
- The energy extracted from wind via wind turbines is a function of the blade area, air density and the cubed value of the wind speed. Wind turbines are classified according to their wind speed cut-in and cut-out rates, *i.e.* the wind speeds that they can accommodate [12]. The South African National Energy Association (SANEA) initiated a project to map the wind resource atlas of South Africa in order to exploit wind energy in the country. This map shows the estimated mean annual wind speeds for various parts of the country. [13]
- The Department of Minerals and Energy released a study on the supply potential for hydropower installations in South Africa [14]. Hydropower is the use of the kinetic energy of flowing water driving a water turbine. Conventional hydropower can be sourced from rivers, dams or canals and is a renewable energy source. Pumped storage on the other hand, requires the potential energy created from the difference in height between two dams. Water is pumped from the bottom dam to the top dam in times of energy oversupply, and then released during peak demand. This can be considered a renewable energy source if renewable energy is used to pump the water. The typical efficiencies of these plants can be sourced from currently installed turbines [15] [16].

3 Study Purpose and Scope

The purpose of this study is to create a model that determines what renewable energy combination can best minimise the required traditional fossil-fuelled baseload supply for a given operational expenditure. The model will be applicable to South Africa's unique electricity demand and supply environment. For example, electricity demand dips during midday and sharply inclines in the evening, compared to the United States and Europe where demand increases steadily during the day, peaks around 16:00, and then declines until early morning. South Africa also has unique long distance high voltage transmission requirements. This does not allow similar international studies to be utilised in South Africa's electricity environment.

The model will compare the supply of a given solution to the energy demand curve for a week in mid-winter, when overall energy demand is at its highest and the incline in early evening electricity demand is the steepest. The focus was on creating a model that is flexible and can be adjusted to the rapidly maturing renewable energy efficiencies and capacities as well as more reliable climate assessments for individual technology site installations.

There exists potential complexity in hybrid energy solution optimisation and this initial study focuses only on the cost and baseload minimisation. When only these two factors are considered, it is possible to use linear programming to find the optimal solution set. However, because it is the intention that this model be used as a sound base for future work that will include other relevant factors that will add to the problem's complexity, a meta-heuristic will be used to solve the current problem. This will allow for future expansion that linear programming will not necessarily allow. Future research and expansion suggestions are discussed in the last section.

4 Multi-objective Modelling of the Hybrid Renewable Energy System using PBIL

4.1 The Population Based Incremental Learning (PBIL) Algorithm

The PBIL was first developed by Shameet Baluja [17] and is an abstraction from the basic Genetic Algorithm (GA). It incorporates competitive learning by using a probability vector when generating solutions. It has shown improvements on speed and accuracy compared to GA when used on static function and discrete space optimisation problems. The current optimisation problem conforms to these two attributes, and it is thus used to optimise the hybrid renewable energy system.

4.2 Modelling the Renewable Energy Technologies

The sizing of individual stations and the cost per megawatt-hour were based on the DoE IPP procurement programme documentation. In this documentation, the maximum site size for a bid per technology is specified. The buy-in tariffs from earlier bidding rounds are also provided [1]. The potential supply data is sourced from various feasibility studies and climate assessments (as discussed in the literature study).

Table 1 shows the cost per megawatt-hour was used for the calculations [1]:

Table 1: Technology Cost per Megawatt-hour

Technology	**Cost per Megawatt-hour**
Solar PV	R 2 850.00
Concentrated Solar Power	R 2 850.00
Wind Energy	R 1 150.00
Pump Storage	R 2 000.00
Hydropower	R 1 030.00

4.3 Assumptions

It is not within the scope of this study to consider existing renewable energy installations in South Africa. This is because the current installed capacity (if pumped storage is excluded) is negligible if compared to the energy demand. This study also does not consider electricity imports from neighbouring countries. Sufficient land availability is assumed for wind and solar PV sites. The supply potential for hydro stations is restraint based on the study from the Department of Minerals and Energy [14].

4.4 The Optimisation Model

A solution comprised of changing the variable i, the number of installations of technology i. Each technology had an associated fixed daily supply curve, based on the resource assessments from the literature. The number of installations was encoded in a binary format to allow for the use of the probability vector.

The fitness of a solution was subject to two factors that were to be minimised in this algorithm. The first variable Cost (C) was calculated by

$$C = \sum_{i=1}^{n} c_i * supplied\ megawatthours_i \tag{1}$$

where c_i is the cost per megawatthour of technology i (as specified by the DoE) and n is the number of different technologies. This is thus the total expenditure for the week for the utility buying the supplied power from the provider.

The second variable, Baseload Requirement (BR) is the traditional or non-renewable required baseload for a given solution. It was minimised for each solution by selectively using the hydro and pumped storage power during the week, the only technologies that are not climate dependent and can be utilised at will. This selective use was optimised by determining for which time intervals the deficit between demand and supply was at its highest, and utilising the

technologies only during these time intervals. The required traditional baseload reliance was thus minimised in this way.

The model was subject to the inherent technology constraints, *i.e.* resource constraints and plant size constraints as discussed in the literature study.

A solution set, or population, was sorted using a non-dominated sorting algorithm to find the Pareto front for the population. This is achieved by removing all dominated solutions from the population.

The probability factor was initiated with a value of 0.5 for each bit. The probability vector was updated according to the formula

$$P_i = P_i(1 - LR) - f_i LR \quad (2)$$

where LR is the learning rate and f_i the fraction of the Pareto dominated front of the population that were true, or equal to 1, for that bit. The learning rate was fixed at 0.15, as it provided consistent model results.

5 Results

The model converged consistently after approximately four hundred iterations. The solution set favoured the use hydro and pumped storage technologies. This was expected as the technologies are not significantly more expensive than solar and wind, and can be utilised at will when the deficit between supply and demand is at a maximum. The use of these technologies is however capped due to the unavailability of water abundant and suitable topographical sites in the country.

An example of the supply and demand curves for a solution for the evaluated week is shown in **Figure 1Error! Reference source not found.**. It can be seen that, as discussed, the supply of individual technologies and the demand curve is a mismatch. However, when the different technologies are combined in an optimal way, the summation of the supply better matches the demand curve. This indicates that the objective of minimising the baseload requirement for a given solution was achieved.

It can be seen in the graph that at times there is, at times, a mismatch between the total supply curve shape and that of the demand curve. This is expected because only five technology types were used, all with inherent constraints. As more renewable technologies mature and become commercially viable, the hybrid solution will be better optimised to match the demand curve.

Figure 2 shows the Pareto optimal solution set. The relationship between renewable energy expenditure per week and required baseload installed capacity is approximately linear. As discussed in the sections above, the scope of the current study was restricted to linear optimisation functions and this linear result set was expected. As expected, the cost of a given solution is increased as the reliance on renewable energies is increased. This solution set can be used by policy makers. For example, if a weekly expenditure of approximately R420 000 on electricity produced by renewable sources is accepted, then only 22 000 MW of installed capacity from traditional fossil-fuelled sources are required (as opposed to the current 40 000 MW).

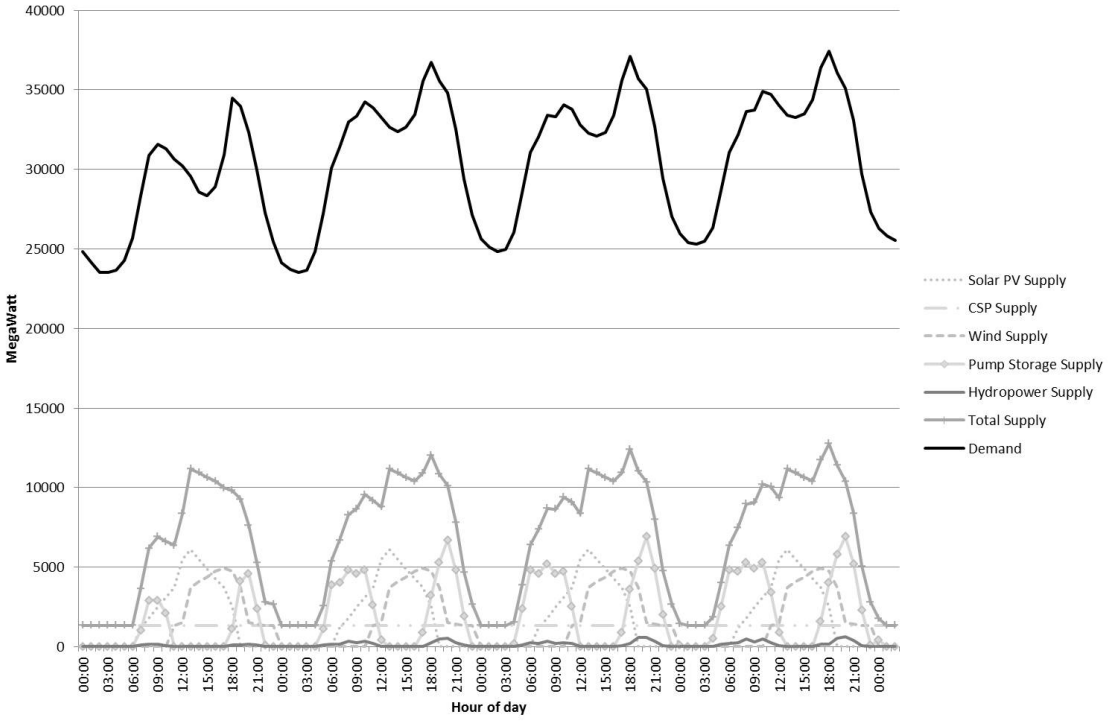


Figure 1: Example of a Hybrid Energy Solution Supply Compared to Demand

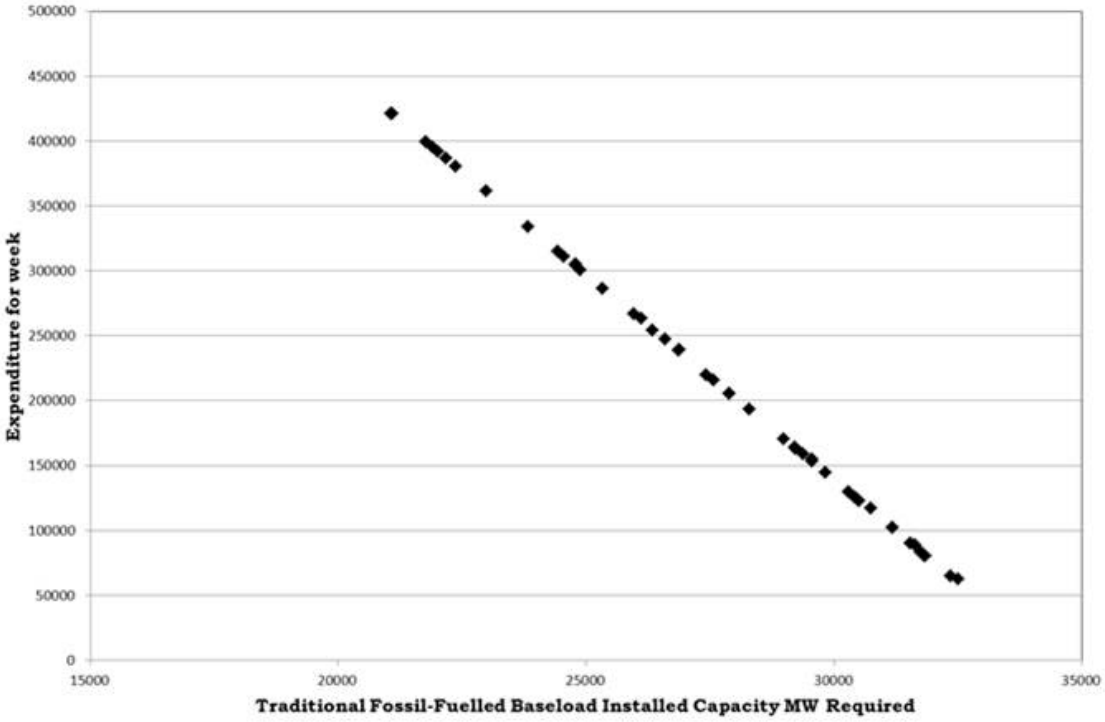


Figure 2: Pareto Optimal Solution Set

It can be seen that with the current technologies it is not possible to meet the country's entire electricity demand using only renewable energies. The reason for this is that the pumped storage schemes require water to be pumped to the top reservoir nightly. The other renewable technologies will not be able to provide this electricity when required.

6 Conclusions

It is concluded that the study achieved the goal of creating a model that shows what renewable energy combination can best minimise the required traditional baseload supply for a given operational expenditure in South Africa. The PBIL algorithm proved successful in providing consistent and converging results.

7 Future Research

The current model only accounts for fluctuations in climate during any given day and not for seasonal climate fluctuations that impact renewable energy output. It is recommended that the model next be expanded to allow for this. In future, Monte Carlo simulation can also be used to further optimise the solution set by considering climate behaviour probability distributions.

As mentioned in the literature study, a hybrid energy solution also needs to include many other factors which can be incorporated in the model in future, as well as new technologies as they mature and become commercially viable.

Bibliography

[1] J. Goldie-Scot, "South Africa decides to give PV and wind a tender embrace," Bloomberg New Energy Finance, 2011.

[2] W. Kempton, "Chapter 5: Analyis Large-Scale Integration of Renewable Energy," in *Renewable Energy Systems*, Boston, Academic Press, 2010.

[3] O. Erdinc and M. Uzunoglu, "Optimum design of hybrid renewable energy systems: Overview of different approaches," *Renewable and Sustainable Energy Reviews,* 2011.

[4] Eskom SOC Holdings Limited, "Adequacy Reports - Archive," 2013. [Online]. Available: http://www.eskom.co.za/c/article/1035/adequacy-report-week-26-2012//.

[5] S. Klychev, M. Mukhammadiev, R. Zakhidov and K. Potaenko, "Technical and Economic Conditions for Creation of Combined Solar-Wind Power Plants," *Applied Solar Energy, ProQuest,* pp. 43(4), 214-217, 2007.

[6] R. Baños, F. Manzano-Agugliaro, F. Montoya, C. Gil, A. Alcayde and J. Gómez, "Optimisation Methods Applied to Renewable and Sustainable Energy: A Review," *Renewable and Sustainable Energy Reviews, Elsevier,* 2010.

[7] R. Dufo-Lopez and J. Bernal-Agustin, "Multi-Objective Design of PV-wind-diesel-hydrogen-battery Systems," *Renewable Energy, Elsevier,* 2008.

[8] South African Department of Energy, "Renewable Energy Independent Power Producer Procurement Programme," 2012. [Online]. Available: http://www.ipprenewables.co.za/.

[9] South African Photovoltaic Industry Association, "Solar PV," 2013. [Online]. Available: http://www.sapvia.co.za/solar-pv.

[10] T. Fluri, "Solar Resource Mapping in South Africa," 2009. [Online]. Available: http://siteresources.worldbank.org/EXTENERGY2/Resources/CSP_Incentives_DP_ 24.pdf.

[11] Eskom SOC Holdings Limited, "Concentrating Solar Power: A Brief Overview of the Technology," 2013. [Online]. Available: http://www.eskom.co.za/c/article/1015/brief-overview-of-the-technology/.

[12] South African Wind Energy Program, "Wind Farm Calculations - SAWEP Workshop," in *WASA*, 2010.

[13] S. Szewczuk and E. Prinsloo, "Wind Atlas for South Africa (WASA): Project overview and current status," in *Science real and relevant conference 2010*, 2010.

[14] B. Barta, "Capacity Building in Energy Efficiency and Renewable Energy: Baseline Study - Hydro Power in South Africa," Department of Minerals and Energy, 2002.

[15] Eskom Generation, "Drakensberg Pumped Storage Technical Information," October 2005. [Online]. Available: http://www.eskom.co.za/content/DrakensbergTechnicalinfo.pdf.

[16] Eskom Generation, "Palmiet Pumped Storage Technical Information," October 2005. [Online]. Available: http://www.eskom.co.za/content/PalmietPumpedStorageTechnicalinfo.pdf.

[17] S. Baluja, "Population-Based Incremental Learning. A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning," School of Computer Science, Carnegie Mellon University, 1994.

[18] O. Elma and S. US, "A comparative sizing analysis of a renewable energy supplied stand-alone house considering both demand side and source side dynamics," *Applied Energy,*

Elsevier, 2012.

[19] N. Kulichenko and J. Wirth, "Regulatory and Financial Incentives for Scaling Up Concentrating Solar Power in Developing Countries," 2011. [Online]. Available: http://siteresources.worldbank.org/EXTENERGY2/Resources/CSP_Incentives_DP_ 24.pdf.

Proceedings of the 2013 ORSSA Annual Conference
pp. 71–81

www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf

ORSSA Proceedings
ISBN
xxx-xxxxx-xxxxx-x

A model to optimise the linked sawing and ripping decisions in the South African pine wood industry

Berndt Lindner[1] Tanya Visser[2] Brand Wessels[3]

Abstract

This paper investigates how to model and link two important cutting operations in the South African pine wood industry. The model is proposed to help answer the question: given the current log supply - what should some of the decisions, specifically machine settings, be for the two cutting centres to try and meet market demand at a minimal cost. The first decision is how to saw up the log supply into different thicknesses by choosing specific sawing patterns; the second is to decide on a rip saw's settings (priority values) which determines how the products from operation one are cut into products of a certain demanded thickness and width. The objective function to minimise includes the raw material waste cost and an over/under production cost. The over production cost is estimated to represent the stock keeping costs. The under production cost is estimated as the buy-in cost of purchasing the raw material from another wood supplier. The modelling tools used in this paper are the Population Based Incremental Learning (PBIL) algorithm and Mixed Integer Programming (MIP) and the modelling software used are Simsaw and AIMMS. The model performs well against current decision software in South Africa, namely the Sawmill Production Planning System (SPPS) package, which combines simulation (Simsaw) and Mixed Integer Programming techniques to maximise profit. The model adds further value in modelling and determining the ripping priority settings linked to the decided upon sawing patterns.

Key words: Lumbering Industry, Cutting Stock, Linear Programming, Metaheuristics, PBIL Algorithm

[1] Berndt Lindner: Department of Industrial Engineering, Stellenbosch University, South Africa, PO Box 3112, Matieland 7602, email: 15150526@sun.ac.za
[2] Department of Industrial Engineering, Stellenbosch University, South Africa, Private Bag X1, Matieland, 7602, email: tanyav@sun.ac.za
[3] Department of Forest and Wood Science, Stellenbosch University, South Africa, Private Bag X1, Matieland 7602, email: cbw@sun.ac.za

1 Introduction

Wood, as a raw material, is known to be the single biggest cost factor for wood product manufacturers [5, 3, 13]. The wood product manufacturing industry consists of many dependent cutting operations along its production chain, as illustrated in **Figure 1**. Recovery of this raw material is heavily influenced by the cutting decisions made. Converting logs into wood products can be viewed as a two-stage process [9]. Logs supplied by the forestry industry are first sawn (operation one in **Figure 1**) into slabs of wood, known as flitches. This operation is commonly referred to as the primary breakdown [9].

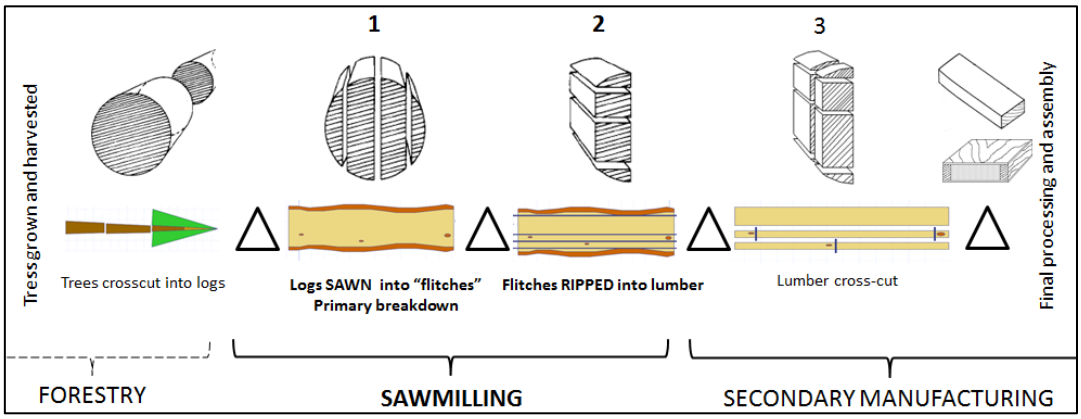


Figure 1: Illustration of important cutting operations in the wood industry. Operations one and two are the focus areas of this paper. Δ represents buffer storage of products.

The term flitch generally refers to a piece of timber sawn from a log, which has two sawn faces and two edges containing bark. These flitches are then further processed to produce ripped pieces of wood (flitches that have been cut along their width as illustrated by operation two in **Figure 1**).

Both of these operations have the potential to waste a significant amount of wood, because of process constraints such as fixed saw blade thicknesses and the cutting decisions. While approximately only 46% of the log volume is converted into sellable products for South African sawmills [13], it is estimated that a one per cent increase in volume recovery will result in additional profit of about R2.2 million annually [18]. The important question is what factors prevent this increase? With certain product and process constraints a given (such as saw blade thicknesses), the most challenging problem is the complexity inherent in matching the naturally varying log supply to the end demand for wood products. Sawmills have traditionally been production driven, operating on a push system, but there has been a gradual move for sawmills to become more market driven and operate as pull systems [4]. This challenge can be addressed by the optimal selection of cutting decisions.

Finding the optimal cutting decisions is subject to two main and often opposing goals: maximising recovery (cutting the greatest amount of products out of the given raw materials with minimal waste) whilst meeting the market's product demand.

2 Literature on sawing operations

A typical machine used for sawing logs into flitches (operation one) in South Africa includes the framesaw, amongst others [11]. A framesaw cuts each log into its different sized flitches depending on the spacing between its various blades, referred to as sawing patterns (**Figure 1**).

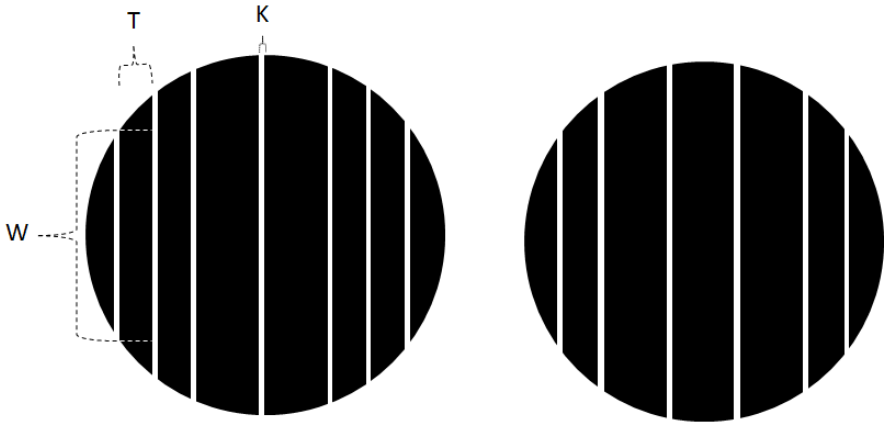


Figure 2: Two possible framesaw sawing patterns, each with different volume and product outputs. T indicates the thickness of the flitch; W indicates the clean width of the flitch, i.e. having no bark; K indicates the kerf or thickness of the blade. Chips and sawdust produced by the blades are commonly referred to as "waste" (since its demand and selling price is extremely low in comparison to solid wood).

Because of the large production volume requirements, a medium-sized South African sawmill has an annual log intake of 100 000 m^3 [18], each log cannot be cut according to a different (specifically optimal) sawing pattern, as setting up each framesaw pattern loses valuable production time. What typically occurs is that logs are sorted into log classes, having certain diameters, and each log class has a set sawing pattern assigned to it. Batches of logs, belonging to a certain log class diameter, are then sawed sequentially. The batched output of flitches are then dried and sorted only according to their thicknesses (with mixed and varying widths) before being processed by the second operation.

Many authors have used Operations Research (OR) techniques for illustrated cutting decisions in the wood industry. Notably *Lumber production optimization* [7] and *Secondary log breakdown optimization with dynamic programming.* Todoroki & Ronnqvist [10] went further and used linked dynamic programming to combine primary and secondary log breakdown (operations one, two and three in **Figure 1**). Various other techniques have also been adapted to the industry and most sawmills today use optimization techniques to maximize yields from logs [8].

In South Africa specifically, software packages are available for simulating and optimising sawing processes, namely Simsaw and SPPS. Simsaw 6 is a sawing simulation tool which predicts the sawn product (board) recovery from logs, given certain inputs [14]. User inputs include the possible sawing patterns and some machine settings. The Sawmill Production Planning System (SPPS) uses simulation data from Simsaw and linear and mixed integer programming techniques to achieve its objective of maximising the total profit, subject to constraints set by the user [12].

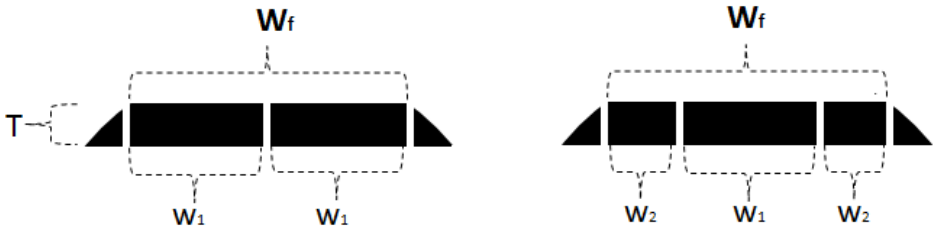


Figure 3: Different ripping decisions present at operation two. Unlike the frame saw at operation one, these saws, called optimising gang rip saws, can shift the spaces between their blades for every piece of incoming material (flitches). T indicates the thickness of the flitch, W_f indicates the clean width of the flitch, note it's the same in both cases; W_1 is product one's width; W_2 indicates product two's width.

For the ripping process (operation two in **Figure 1**) the industry has changed from using extensive manual decision sawing systems to machines capable of automated decision making in an effort to increase yield, amongst other objectives [17]. These machines are referred to as gang saws. There are many types of gang saws but the ones having multiple moving blade saws produce the greatest recovery, assuming they are equipped with an accurate scanning system [17].

Most of the optimising rip (operation two) and chop (operation three) saws use priority rule calculations to determine which products to rip. There are two basic priority value set up modes: static and dynamic. The static priority mode uses only one value for each product during the ripping process, unless changed manually by the operator [16, 6]. Static values are the conventional value setup system and are split up into two criteria: value and yield. The yield method maximizes the part yield, based on the surface area without considering the demand requirements and does thus not employ any value system. Such yield-based methods tend to rip products that best fit into the cutting decision, making it difficult to meet specific market demanded products. This is, however, how Simsaw and subsequently SPPS models ripping decisions are made. Value methods, on the other hand, maximize total part value from each strip, based on the part values assigned to each part size. This method is more commonly used in rough mills when the goal of the cutting is to satisfy demand [17].

3 Problem and Methodology

Most of the results obtained from the models and software packages mentioned are difficult to interpret and implement with the industry's independent machine centres. Specifically determining the priority values at operation two in conjunction with the decisions, or cutting patterns, chosen at operation one within the industry's operational method of using batches in the production of logs and flitches. This paper develops a model to find the best way of sawing batches of logs. This is done by determining the settings at operation one and two that is optimal over the entire batch. Because of these criteria it was felt that metaheuristics would be more suitable than other analytical techniques.

3.1 Methodology

There have not been many metaheuristics applied specifically to the problem addressed in this paper to the best of the authors' knowledge. However there have been related studies for using

metaheuristics on other cutting decisions in South Africa. Van Zyl [18] investigated the use of *metaheuristics in the optimisation of log position during sawmill processing* in South Africa. The metaheuristics investigated include particle swarm optimisation, simulated annealing, the genetic algorithm, and the Population Based Incremental Learning (PBIL), amongst others. In addition, Wessels et al. [15] developed a search (tentacle) algorithm to also determine the optimal log position before being sawn (a typical decision part of operation one) and compared it to the other algorithms mentioned above. Wessels et al. [15] found the tentacle and PBIL algorithm performed the best, but unfortunately not significantly better than randomly sampling within the search space. Based on their findings, though, it was felt that the PBIL algorithm is suitable for use in this paper.

The PBIL algorithm was developed by Shameet Baluja [1] and was based upon the genetic algorithm. It uses an updated probability vector as a basis for generating a population of solutions. The probability vector is updated by it learning or striving toward the fittest solutions in the population. There are four parameters that can be adjusted in the PBIL algorithm and the typical values suggested by some authors [1, 18, 15, 2] are: population size (10-100 solution vectors); learning rate (0.1-0.4); mutation shift (0.05 or 0.1); and mutation probability (0.02-0.1).

4 Formulation of the problem

4.1 Sawing flitches (operation one)

The inputs for this operation are a number of logs supplied by foresters falling within specified log diameter classes. Logs are trees that have been cross-cut into appropriate lengths, usually specified by the sawmill. A parameter of this operation is the possible sawing patterns pre-determined by management. Usually those yielding the highest recovery rates and an appropriate mixture of products demanded are possible selections based on experience. The decision variables represent which sawing pattern will be assigned to which of these log class, equation (1). The outputs from this operation are a number of edged boards, called flitches (4). For this paper the flitches are classified by their clean widths - clean meaning there is no bark or wane and its edges are straight (**Figure 2**). These clear flitches have a determined thickness and width (2). Instead of determining each log's flitch output mathematically and, so to speak, re-invent the wheel, existing simulation software (Simsaw 6) is used. With Simsaw the user can generate logs and specify possible sawing patterns and it simulates the board output for each log class and sawing pattern. For this paper the logs were distributed normally within 95% limits of specified diameters (e.g. a lower limit of 19 and upper limit 20.9 cm). These logs were then simulated through user defined sawing patterns and the output boards, namely the number of flitches (3), were used as a parameter in the model.

$$y_{dc} = \begin{cases} 1 \; Log\ class\ d\ is\ assigned\ to\ sawing\ pattern\ c.\ \ c = 1,2\ ,3,\dots,j \\ 0 \; otherwise \end{cases} \qquad (1)$$

$$F_{tf} = \text{Clean width measurement of flitch with thickness t and clean width f}$$
$$t = 1,2,3,\dots,k; f = 1,2,3,\dots,m. \quad (2)$$

$$SF_{dctf} = \text{Simulated number of flitches, having thickness t and clean width f produced if log class d is assigned to possibe cutting pattern c} \quad (3)$$

$$NF_{tf} = \sum_{d=1}^{i}\sum_{c=1}^{j} Y_{dc} DF_{dctf} \quad (4)$$

Equation 4 then calculates the number of flitches having thickness (*t)* and clean width (*f)* that becomes the input material for the second operation.

$$\sum_{c=1}^{j} y_{dc} = 1 \quad \forall\, d \quad (5)$$

Constraint 5 is required to ensure that all log classes *(d)* are assigned to only one sawing pattern *(c)*. This constraint was enforced into the model by cleverly manipulating the PBIL's probability vector.

4.2 Ripping the flitches (operation two)

The second operation was formulated as a Mixed Integer Problem (MIP) as follows. Dynamic programming could also be used to model this problem [9].

$$P_w = \text{the width of product } w \qquad w = 1,2,3,\dots,n \quad (6)$$

$$R_{tw} = \text{priority value of products ripped with thickness t and width } w \quad (7)$$

$$x_{tfw} = \text{number of } w \text{ products cut from flitch having thickness t and width f} \quad (8)$$

$$x_{tfw} \text{ is integer} \quad (9)$$

$$K = Kerf,\ \text{thickness of blade} \quad (10)$$

Then the objective function for operation 2 is modelled as:

$$\max z = \sum_{t=1}^{k}\sum_{f=1}^{m}\sum_{w=1}^{n} P_w R_{tw}\, x_{tfw} \quad (11)$$

Subject to the following two constraints:

$$\sum_{w=1}^{n} x_{tfw} \le 3 \quad \forall\, (t,f) \quad (12)$$

$$\sum_{w=1}^{n} x_{tfw}\, P_{tw} + \left(\sum_{w=1}^{n} x_{tfw} - 1\right) K \le F_{tf} \quad \forall\, (t,f) \quad (13)$$

Constraint (12) serves to ensure that a maximum of three products are ripped per flitch. A value of three is used since this the actual setting for the machine used by the case company. Please note this value can change depending on a machine's design and settings. Constraint (13)

is to ensure that the sum of all the products cut out ($\sum_{w=1}^{n} P_w x_{tfw}$) and the sum of the blade's thicknesses ($\sum_{w=1}^{n} x_{tfw} - 1$) fit into the clean width of the flitch (F_{tf}).

The optimal ripping pattern for each flitch is found and after all the flitches have been virtually ripped, the final supply of products having thickness t and width w is determined as:

$$S_{tw} = \sum_{f=1}^{k} x_{tfw} NF_{tf} \tag{14}$$

4.3 Operation one and two

As can be seen the output from operation one, namely the number of flitches having thickness t and clean width f (4) which depend depends on y_{dc} (1) becomes the input for operation two and depending on decided upon R_{tw} (7), x_{tfw} products will be ripped from a flitch having a clean width f. This is then multiplied by the number of flitches having that clean width f (NF_{tf}) to determine the total supply of products having thickness t and width w (14). So for this paper's problem the linked decisions are the variables in equations (1) and (7). These two variables are represented together as a single binary code and the probability vector in the PBIL has the same amount of bits from which a population of solution vectors are created and tested against a fitness function.

The fitness function for each solution is determined by the total cost incurred for that solution. This cost is a summation of the estimated raw material waste cost and the cost of over or under producing. The volume (m^3) of raw material wasted is determined by subtracting the total products produced (14) from the total supply of logs. This is then multiplied by the raw material cost (R/m^3). Overproduction incurs handling and damage costs amongst others [6]. The over production cost is determined, if the total supply (14) exceeds the specified demand, by subtracting these two volumes and multiplying this with the estimated cost of storing all the products overproduced (R/m^3). Underproduction can delay assembly or delivery of the final product. Some companies incur an extra cost to fill their orders by purchasing the under-produced parts from alternative sources [6]. The under production cost is similarly determined by the cost of buying in the volumes short from another supplier. SPPS has a similar cost factor included in its profit function, but it does not have a cost factor for overproducing.

Except for Simsaw 6 determining the variable in equation (4) all equations were modelled in AIMMS 3.12. The MIP problem of operation two was solved (CPLEX 12.4) within the PBIL's execution code.

5 Model application – a case study

Data for the model was discussed and captured with management of a typical pine wood manufacturing company (**Table 1**-3). The company is a dominant player in the South African market.

Table 1: Logs supplied according to case company's Enterprise Resource Planner (ERP).

Supply			**(March 2013)**
Log class	Diameter [cm]	Number	Volume [m^3]
1	19-20.9	1284	127
2	21	2197	262
3	23	2913	412
4	25	2927	485
5	27	3129	599
6	29	2915	639
7	31	2510	625
8	33	1680	471
9	35	1113	350
10	37	611	213
11	39	332	128
12	41	440	187
		22051	4499

Table 2: Average expected demand for the 2012 year determined by management at case company [m^3].

Demand				**(Monthly Average 2012)**				
w / t	15	32	36	42	47	71	96	148
25				100	100	251	414	235
38	120	200	16	140	80	250	504	40

Table 3: Values used by the model, cost were estimated by company's management accountants

Model's parameters		**(May 2013)**
Raw material cost estimate	2174	R/m^3
Over production cost estimate Warehousing and related	138	R/m^3
Under production cost estimate Buy in cost calculate to SA's lumber price index	2367	R/m^3
Kerf of blades at operation 1	5	mm
Kerf of blades ate operation 2	4.7	mm
Number of possible sawing patterns at operation 1	4	

6 Results and Discussion

Firstly, the PBIL's parameter settings mentioned in section 3.1 were tested in the model and evaluated under different parameter combinations. The best combination of parameters found (Figure 4) is: learning rate = 30%; mutation shift = 5% and mutation probability = 5%. This best PBIL parameter combination was again run, for 500 generations, and the final answer was compared to SPPS solver's optimal solution and also to the best solution found by the model randomly searching within the search space (Table 4).

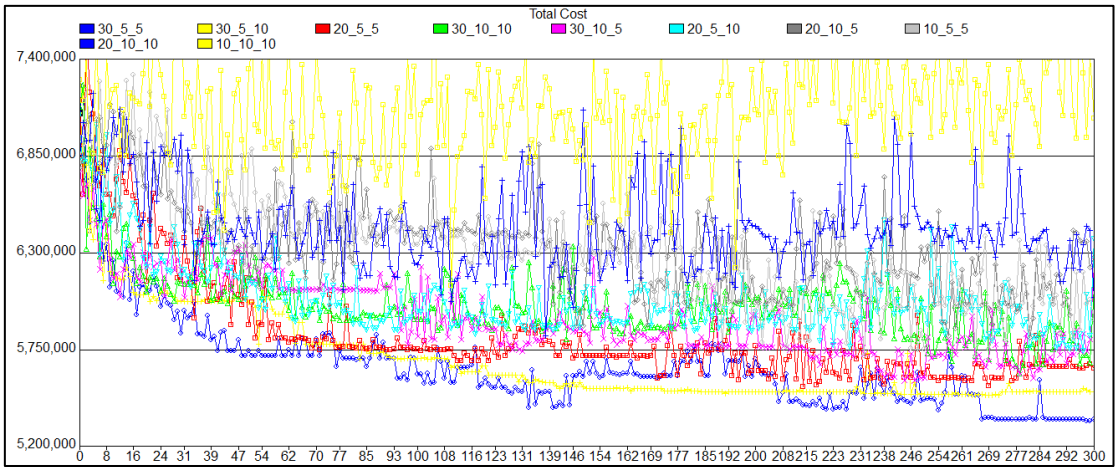


Figure 4: The model's best (min) cost function found, y-axis, within each generation (300), x-axis, containing 30 solution vectors (population size in PBIL). This is under the different combinations of the PBIL's parameters proposed. 30_10_5 indicates the combination with learning rate = 30%, mutation shift = 10% and mutation probability = 5%. The different combinations are ordered from best to worst in the legend.

The model was compared to SPPS (which also utilises Simsaw), since it is the most similar modelling tool or technique known in South Africa, however it differs somewhat in its problem formulation.

Table 4: Results found by this paper's model, the model randomly searching (not learning), and SPPS's solution.

	Model's best PBIL-30_5_5	Model's random search PBIL-0_0_0	SPPS solver
Total cost function	R5 303 046	R6 065 040	R6 445 488
Raw material waste cost	R4 372 138	R4 422 040	R4 574 444
Under production cost	R874 626	R1 550 483	R1 775 162
Over production cost	R56 281	R92 517	R95 882

The model outperforms the SPPS solver in the specific metrics above and data used. This does not mean the model outperforms the SPPS software in general, especially the raw material cost, since both have slightly different optimising objectives. SPPS maximises profit instead of just the cost function used in this paper, it can also take into account other constraints, such as production and kiln drying capacities. More results and comparisons would further substantiate the model's performance. However the model adds great value by taking into account over production costs, since only an under production cost is used in SPPS's objective function. Most importantly the model differentiates itself in that it determines the ripping priority settings (which SPPS does not) at operation two as well as well as the linked sawing pattern decisions at operation one.

6.1 Conclusions and recommendations

The model was able to link the sawing and ripping decisions desired and the simulation and optimisation techniques used helped the model outperform the compared SPPS solver's solutions for the specific metrics and data used. More studies are needed to further validate and substantiate the model.

Further improvements to the model would be: expanding it to include the third operation (cross-cutting) and its machine priority settings; more evaluation of the number and possible sawing patterns; and formulating more accurate cost equations - there is for example a greater cost involved with over production than just the warehouse cost used considered in this paper. Finally, the model and its results are from deterministic inputs, techniques such as Monte-Carlo simulation could be useful for the naturally varying log supply and fluctuating market demands. This would make the model and its results stochastic, adding more value to the decision maker.

References

[1] Baluja, S., 1994. Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning. School of Computer Science, Carnegie Mellon University.

[2] Bekker, J., 2012. Simulation of discrete event stochastic processes. MEng Course Notes. Quantitative Management 2. Departmenr of Industrial Engineering. Stellenbosch University.

[3] Buehlmann, U., Thomas, R.E. & Zuo, Xiaoqui, 2011. Cost minimization through optimized raw material quality composition. Robotics and Computer-Integrated Manufacturing, 27(4), pp.746–754.

[4] Kapp, S.B. et al., 1999. A feasibility study on the development of an integrated manufacturing planning and control system in the South African Sawmill Industry. South African Forestry Journal, (184(1)), pp.80–87.

[5] Lundahl, C., 2007. Optimized processes in sawmills. Licentiate Thesis. Lulea ° University of Technology. Division of Wood Technology, Skelleftea °, Sweden.

[6] Maness, T., Ye, L. & Ristea, C., 2009. An Adaptive Optimization Method for Chop Saw Systems. INFOR: Information Systems and Operational Research, 47(3), pp.203–213.

[7] Reinders, M. & Hendriks, T., 1989. Lumberproduction optimization. European journal of operational research, 42, pp.243–253.

[8] Todoroki, Christine & Rönnqvist, M., 2002. Dynamic Control of Timber Production at a Sawmill with Log Sawing Optimization. Scandinavian Journal of Forest Research 17, 79–89.

[9] Todoroki, CL & Ronnqvist, E., 1997. Secondary log breakdown optimization with dynamic programming. Journal of the Operational Research Society. 48(5), pp.471–478.

[10] Todoroki, CL & Rönnqvist, E., 1999. Combined primary and secondary log breakdown optimisation. Journal of the Operational Research Society, 50(3), pp.219–229.

[11] Wessels, CB, 2009. A model to determine the theoretical maximum feed speed of a frame saw. Southern Forests: a Journal of Forest Science, 71(1), pp.31–36.

[12] Wessels, CB et al., 2006. Integrating harvesting and sawmill operations using an optimized sawmill production planning system. Proceedings International Preceion Forestry Symposium, Stellenbosch, South Africa, IUFRO, ISBN 0-7972-1121-7, pp117. Vol. 118.

[13] Wessels, Cb, 2010. Sawmill efficiency measures, Lecture Notes Distributed in Wood Products Science 454. Stellenbosch University.

[14] Wessels, Cb, Dell, M.P. & Price, C.S., 2001. SIMSAW 6 User Manual, CSIR,Durban, South Africa.

[15] Wessels, Cb, De V du Plessis, J. & Smit, N., 2011. Cant sawing log positioning in a sawmill: searching for an optimal solution. Southern Forests: a Journal of Forest Science, 73(1), pp.15–22.

[16] Ye, L. & Maness, T.C., 2006. A real-time goal-seeking model for optimizing chop saw systems. Forest products journal, 56(11-12), 73-78, 56(9959).

[17] Zuo, X. 2003. Improving lumber cut-up manufacturing effi ciency using optimization methods. Raleigh, NC: North Carolina State University. 222 p. Ph.D. dissertation.

[18] Van Zyl, F., 2011. Determining the optimal log position during primary breakdown using internal wood scanning techniques and meta-heuristic algorithms. MScEng Thesis. , (March). Stellenbosch University.

Proceedings of the 2013 ORSSA Annual Conference
pp. 82–89
www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf

ORSSA Proceedings
ISBN
XXX-XXXXX-XXXX-X

Optimal Liquidity Execution Planning using Stochastic Programming and Robust Optimization

H. Raubenheimer* S.E. Terblanche†

Abstract

The new Basel III framework sets out higher and better quality capital, better risk coverage, the introduction of a leverage ratio as a backstop to the risk-based requirement, measures to promote the buildup of capital that can be drawn down in periods of stress, and the introduction of two global liquidity standards. These liquidity standards focus on testing the short- and long- term solvency of banks. Apart from managing a dedicated portfolio of high quality liquid assets and attaining short- and long-term liquidity ratios of above 100%, banks must also consider strategizing its response to liquidity crises in advance, including a plan for liquidity execution.

In this paper two optimization approaches for resource allocation in the liquidity execution process are considered. We propose a stochastic programming model and a robust optimization model incorporating uncertainty and worst cases in terms of "market depth" and liquidity stress. A formulation of the two models is provided which is computationally efficient and minimizes the cost of the liquidity execution. An illustrative example is provided to show how these models could be applied in practice in order to perform liquidity execution planning.

Key words: Liquidity Execution; Stochastic Programming; Robust Optimization

1 Introduction

Liquidity risk may refer to the risk of money markets not being able to supply funding to businesses when required (i.e. funding liquidity risk) or, more broadly, to the management of short term cashflow requirements. Alternatively it may refer to an insufficient capacity in the market to handle asset transactions at the time when the deal is required (i.e. market liquidity risk). In [4] funding liquidity risk in general is described as a consequential risk following a troubled market situation characterized by losses. And it is a risk faced by all financial institutions. For this reason banks hold liquidity capacity.

In [3] the recent credit crisis is attributed to the build-up of excessive on and off-balance sheet leverage in the banking sector, as well as the gradual erosion of the level and quality of the capital base. Furthermore, many banks were holding insufficient liquidity buffers. According to [3] the market lost confidence in the solvency and liquidity of many banking institutions during the crisis. This sentiment rapidly transmitted to the rest of the financial system and the real economy. This resulted in a massive contraction of liquidity and credit availability.

*Centre for Business Mathematics and Informatics, North-West University, Potchefstroom
†Centre for Business Mathematics and Informatics, North-West University, Potchefstroom

In order to promote a stronger and more robust banking sector, the new Basel III liquidity regulation ([2] and [3]) aims to reform and strengthen global capital and liquidity rules. The new regulations highlight the importance of managing a liquidity contingency buffer in order to improve the ability of the banking sector to absorb shocks arising from financial and economic stress. This should reduce the risk of spillover from the financial sector to the real economy. In [4] a description is provided of the regulations related to maintaining a high quality liquidity portfolio for the purposes of hedging against liquidity outflows under stress scenarios.

The new Basel III framework sets out higher and better-quality capital, better risk coverage, the introduction of a leverage ratio, measures to promote the buildup of capital that can be drawn down in periods of stress, and the introduction of two global liquidity standards. These liquidity standards, namely the Liquidity Coverage Ratio (LCR) and the Net Stable Funding Ratio (NSFR), focus on testing the short- and long- term solvency of banks. Liquid assets include level 1 and level 2 assets. Level 1 assets have a zero haircut and include cash, central bank reserves, and premium government and municipal bonds. Level 2 assets have a regulatory haircut of 15% and include amongst others high quality corporate and covered bonds. Apart from managing a dedicated portfolio of these high quality liquid assets and attaining short-term (LCR) and long-term (NSFR) liquidity ratios of above 100%, banks must also consider strategizing their response to liquidity crises in advance, including a plan for liquidity execution.

This paper investigates two optimization approaches for resource allocation in the liquidity execution process extending the work done by [4] related to planning for optimal liquidity execution. A stochastic programming model is proposed for solving the optimal liquidity execution problem by taking scenario data into account. In addition, a robust optimization model is presented which incorporates the uncertainty in instrument prices and worst cases in terms of liquidity stress. Section 2 discusses the optimal liquidity execution proposed by [4]. In Section 3 and 4 we discuss the formulation of the two proposed optimization models, which minimizes the cost of the liquidity execution. Section 5 focusses on the results from the analysis of the model parameters of both models, before conclusions are set out in Section 6.

2 Planning for optimal liquidity execution

In [4] three multi-stage minimal cost optimization models are proposed to quantify the decision regarding liquidity execution. These models provide an intuitive view of when and how funds should be deployed to meet liquidity needs while minimizing transaction costs. These models provide not only a feasible approach at the time of liquidity execution i.e., in a distress period, but also pffers build an a priori practical liquidity plan consistent with Basel III required contingency funding plans. Furthermore the models are useful for testing the sufficiency of a portfolio.

The first model in [4] deploys an inventory of liquidity assets and imposes only time constraints. The model accommodates multi-stage liquidity requirements where the liquidity gap and execution cost can vary across stages. The second model extends the first by assuming a linear price dependency on the volume of tradable assets, i.e. tiered execution costs. It is assumed that assets can be traded up to a certain level, called the "market depth" at a specific market bid-ask spread. To execute trades at a higher volume, the cost increases unfavorably. The third model accommodates collateralized borrowing, i.e. the financial firm can pledge an asset as collateral instead of selling it. In the subsequent sections we propose extensions of the second model to incorporate uncertainty. Furthermore, rather than using a market value approach, we formulate the problem in terms of share volumes. By doing this we are able to model price scenarios from real market data.

3 Optimal Liquidity as a Stochastic Programming Problem

In order to construct a scenario tree we simulate future market states. A scenario tree is a discrete approximation of the joint distribution of random factors (prices). To facilitate the mathematical formulation of the optimization problem, we represent the scenario tree in terms of states (nodes) $s_t^{v(t)}$, where time $t = 0, 1, 2, \ldots, T$ and $v(t) = 0, 1, 2, \ldots, N_t$ the numbers of the states at time t. The set of states at time t are denoted by $\mathcal{S}_t = \left\{ s_t^{v(t)} | v(t) = 0, 1, \ldots, N_t \right\}$, where $\mathcal{S}_0 = \left\{ s_0^0 \right\}$. The set of all states in the scenario tree is denoted by $\mathcal{S} = \cup_{t=0}^{T} \mathcal{S}_t$. Links $\varepsilon \in \mathcal{S} \times \mathcal{S}$, indicate the possible transitions between states. To enforce non-anticipativity, i.e.to prevent foresight of uncertain future events, we order the elements of ε in pairs $\left(s_t^{v(t)}, s_{t+1}^{v(t+1)} \right)$ where the dependence of the index $v(t)$ on t is explicitly indicated. The order of the states indicates that state $s_{t+1}^{v(t+1)}$ at time $t+1$ can be reached from state $s_t^{v(t)}$ at time t. $s_{t+1}^{v(t+1)}$ is the successor state and $s_t^{v(t)}$ the predecessor state. By using the superscript "+" to denote the successor states, and the superscript "-" to denote the predecessors, we have $s_t^{v(t)+} = s_{t+1}^{v(t+1)}$ and $s_{t+1}^{v(t+1)-} = s_t^{v(t)}$. Each state $s_t^{v(t)}$ has an associated probability p_t^s, for $s \in \mathcal{S}_t$, such that $\sum_{s \in \mathcal{S}_t} p_t^s = 1$. $s \in \mathcal{S}$ can be seen as a path through the scenario tree.

Let $\mathcal{K} = \{1, 2, \ldots, K\}$ be the index set for the K distress stages. Let $\mathcal{J}(k) = \{1, 2, \ldots, J_k\}$ denote the periods within a distress stage $k \in \mathcal{K}$. The liquidity gap expected during each distress stage $k \in \mathcal{K}$ is given by G_k. Let the set $\mathcal{I} = \{1, 2, \ldots, I\}$ denote the indices of the instruments available to deploy and let A_i be the market value or principal of each instrument $i \in \mathcal{I}$. Let l_{ij} be the execution limit of instrument i in a given execution period $j \in \mathcal{J}(k)$ in a given distress period $k \in \mathcal{K}$. In an attempt to capture market impact, a tiered approach is followed to represent transaction costs. That is, the expected transaction cost for executing a trade of average volume for instrument $i \in \mathcal{I}$ in distress period $k \in \mathcal{K}$ is given by c_{0ik}. However, if the transaction volume exceeds a preselected threshold v_{ik}, the transaction cost is given by c_{1ik} with $c_{1ik} > c_{0ik}$. The price at which an instrument $i \in \mathcal{I}$ is executed in an execution period $j \in \mathcal{J}(k)$, distress period $k \in \mathcal{K}$ and scenario $s \in \mathcal{S}$ is given by p_{ij}^s.

The decision variable $w_{ij}^s \geq 0$ is used to denote the volume of instrument $i \in \mathcal{I}$ that will be executed in a execution period $j \in \mathcal{J}(k)$, distress stage $k \in \mathcal{K}$ and scenario $s \in \mathcal{S}$. The auxiliary variable $y_{ij}^s \geq 0$ is used to represent the excess volume above the threshold v_{ik}. The objective of the problem is to minimize the tiered transaction costs for each distress stage $k \in \mathcal{K}$ and scenario $s \in \mathcal{S}$, which is given by the following:

$$f_k^s = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}(k)} (c_{1ik} - c_{0ik}) p_{ij}^s y_{ij}^s + c_{0ik} p_{ij}^s w_{ij}^s \quad \forall k \in \mathcal{K}, \quad \forall s \in \mathcal{S} \tag{1}$$

The resulting Optimal Liquidity Stochastic Programming Problem (OLSPP) is given by the following:

$$\min \quad \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}} \rho^s f_k^s \tag{2}$$

s.t.

$$w_{ij}^s \leq l_{ij} \qquad \forall i \in \mathcal{I}, \quad \forall j \in \mathcal{J}(k), \quad \forall k \in \mathcal{K} \tag{3}$$

$$y_{ij}^s \geq w^s - v_{ik} \qquad \forall i \in \mathcal{I}, \quad \forall j \in \mathcal{J}(k), \quad \forall k \in \mathcal{K}, \quad \forall s \in \mathcal{S} \tag{4}$$

$$\sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{J}(k)} w_{ij}^s \leq A_i \qquad \forall i \in \mathcal{I}, \quad \forall s \in \mathcal{S} \tag{5}$$

$$\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}(k)} p_{ij}^s w_{ij}^s - f_k^s \geq G_k \qquad \forall k \in \mathcal{K}, \quad \forall s \in \mathcal{S} \tag{6}$$

Constraint set (3) ensures that the execution volume w_{ij}^s remains below the predefined limit l_{ij} and constraint set (4) facilitates the tiered transaction costs by assigning a value to y_{ij}^s whenever the execution volume w_{ij}^s exceeds the predefined threshold v_{ik}. Constraint set (5) ensures that the execution volume for each asset $i \in \mathcal{I}$ over all the execution periods does not exceed the available volume A_i and the constraint set (6) specifies that the execution amount minus the costs should satisfy the liquidity gap G_k for each distress stage $k \in \mathcal{K}$.

4 Optimal Liquidity as a Robust Optimization Problem

The basic idea behind the robust formulation is the assumption that instrument prices are from an uncertainty set $\mathcal{P}$, which denotes the convex hull of all historic prices, instead of using scenario-dependent prices p_{ij}^s. That is, the variables $\varphi_{ij} \in \mathcal{P}$ are introduced as part of the optimization problem to represent the worst case prices for every historic period $j \in \mathcal{J}(k)$ in historic distress stage $k \in \mathcal{K}$. For testing purposes, the uncertainty set $\mathcal{P}$ is taken as the convex hull of the scenario generated prices p_{ij}^s. This will allow for a more direct comparison with the stochastic programming approach outlined above. One of the benefits of using a robust framework is that we can now drop the scenario indices from the variables that will be re-used from the OLSPP formulation. For instance, the variable $w_{ij} \geq 0$ without the scenario subscript will be used to denote the volume of instrument $i \in \mathcal{I}$ that will be executed in a execution period $j \in \mathcal{J}(k)$, distress period $k \in \mathcal{K}$. Similarly will we use the variable y_{ij} without the scenario subscript to denote the excess volume above the threshold v_{ik}.

In order to obtain a better understanding of the proposed robust formulation it would suffice to firstly consider how the constraint set (6) presented above as part of the OLSPP would be treated within a robust setting. The constraints (6) state that for all scenarios $s \in \mathcal{S}$, the amount generated by trading the assets $i \in \mathcal{I}$ over the periods $j \in \mathcal{J}(k)$, minus the trading costs, should satisfy the liquidity gap G_k for each distress stages $k \in \mathcal{K}$. In the robust framework, the same constraint is formulated such that the amount generated by trading the assets minus the trading costs should satisfy the liquidity gap provided that the worst case prices $\varphi_{ij} \in \mathcal{P}$ are considered. That is, the constraint is in part an optimization problem, which can be formulated as follows:

$$\min_{\varphi_{ij} \in \mathcal{P}} \left\{ \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}(k)} (\varphi_{ij} w_{ij} - (c_{1ik} - c_{0ik}) y_{ij} - c_{0ik} w_{ij}) \right\} \geq G_k \tag{7}$$

The optimization problem on the left-hand side of the constraint can be written as:

$$\min \quad \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}(k)} \varphi_{ij}((1 - c_{0ik}) w_{ij} - (c_{1ik} - c_{0ik}) y_{ij}) \tag{8}$$

$$\text{s.t.} \quad \varphi_{ij} - \sum_{s \in \mathcal{S}} \lambda^s p_{ij}^s = 0 \qquad \forall i \in \mathcal{I}, \ \ \forall j \in \mathcal{J}(k) \tag{9}$$

$$\sum_{s \in \mathcal{S}} \lambda^s = 1 \tag{10}$$

The constraints (9) and (10) express the worst case price φ_{ij} as a convex combination of the prices in the uncertainty set $\mathcal{P}$. It is clear that this is a non-linear optimization problem in the variables φ_{ij}, w_{ij} and z_{ij}. A dual approach is now followed (see [1]) to obtain a so-called linear robust counterpart. By associating the unrestricted dual variables $\pi_{ij}^{(1)}$ and $\gamma_k^{(1)}$ with the constraints (9) and (10) respectively, the following dual problem, referred to as subproblem one

(SP1), can be written as:

$$\max \quad \gamma_k^{(1)} \tag{11}$$

$$\text{s.t.} \quad \gamma_k^{(1)} - \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}(k)} p_{ij}^s \pi_{ij}^{(1)} \leq 0 \qquad \forall s \in \mathcal{S} \tag{12}$$

$$\pi_{ij}^{(1)} - (1 - c_{0ik}) w_{ij} + (c_{1ik} - c_{0ik}) y_{ij} \leq 0 \qquad \forall i \in \mathcal{I}, \quad \forall j \in \mathcal{J}(k) \tag{13}$$

A similar approach can now be followed to linearize the robust objective function given below:

$$\max \quad \theta \tag{14}$$

$$\text{s.t.} \quad \theta \leq \min_{\varphi_{ij}\in\mathcal{P}} \left\{ \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{J}(k)} ((c_{1ik} - c_{0ik}) y_{ij} \varphi_{ij} + c_{0ik} w_{ij} \varphi_{ij}) \right\} \tag{15}$$

The optimization problem on the right-hand side of the constraint (15) can be written as:

$$\min \quad \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{J}(k)} \varphi_{ij} ((c_{1ik} - c_{0ik}) y_{ij} + c_{0ik} w_{ij}) \tag{16}$$

$$\text{s.t.} \quad \varphi_{ij} - \sum_{s\in\mathcal{S}} \lambda^s p_{ij}^s = 0 \qquad \forall i \in \mathcal{I}, \quad \forall j \in \mathcal{J}(k) \tag{17}$$

$$\sum_{s\in\mathcal{S}} \lambda^s = 1 \tag{18}$$

The dual approach followed above with the liquidity constraint is repeated to obtain the linear robust counterpart for the objective function. By associating the unrestricted dual variables $\pi_{ij}^{(2)}$ and $\gamma^{(2)}$ with the constraints (17) and (18) respectively, the second dual subproblem (SP2) can be written as:

$$\max \quad \gamma^{(2)} \tag{19}$$

$$\text{s.t.} \quad \gamma^{(2)} - \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{J}(k)} p_{ij}^s \pi_{ij}^{(2)} \leq 0 \qquad \forall s \in \mathcal{S} \tag{20}$$

$$\pi_{ij}^{(2)} - c_{0ik} w_{ij} - (c_{1ik} - c_{0ik}) y_{ij} \leq 0 \qquad \forall i \in \mathcal{I}, \quad \forall j \in \mathcal{J}(k) \tag{21}$$

By combining the two subproblems SP1 and SP2 with the original problem constraints (3), (4) and (5) from OLSPP, the following Optimal Liquidity Robust Programming Problem (OLRPP) is obtained:

$$\max \quad \theta \tag{22}$$

$$\text{s.t.} \quad \theta \leq \gamma^2 \tag{23}$$

$$\gamma^{(2)} - \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{J}(k)} p_{ij}^s \pi_{ij}^{(2)} \leq 0 \qquad \forall s \in \mathcal{S} \tag{24}$$

$$\pi_{ij}^{(2)} - c_{0ik} w_{ij} - (c_{1ik} - c_{0ik}) y_{ij} \leq 0 \qquad \forall i \in \mathcal{I}, \quad \forall j \in \mathcal{J}(k) \tag{25}$$

$$\gamma_k^{(1)} \geq G_k \qquad \forall k \in \mathcal{K} \tag{26}$$

$$\gamma_k^{(1)} - \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}(k)} p_{ij}^s \pi_{ij}^{(1)} \leq 0 \qquad \forall s \in \mathcal{S} \tag{27}$$

$$\pi_{ij}^{(1)} - (1 - c_{0ik}) w_{ij} + (c_{1ik} - c_{0ik}) y_{ij} \leq 0 \qquad \forall i \in \mathcal{I}, \quad \forall j \in \mathcal{J}(k) \tag{28}$$

$$w_{ij} \leq l_{ij} \qquad \forall i \in \mathcal{I}, \quad \forall j \in \mathcal{J}(k), \quad \forall k \in \mathcal{K} \tag{29}$$

$$y_{ij} \geq w_{ij} - v_{ij} \qquad \forall i \in \mathcal{I}, \quad \forall j \in \mathcal{J}(k), \quad \forall k \in \mathcal{K} \tag{30}$$

$$\sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{J}(k)} w_{ij} \leq A_i \qquad \forall i \in \mathcal{I} \tag{31}$$

5 Computational Results

Table 1 displays the available liquidity sources with the number of assets and current prices. Bond 1 and equity 1 both carry a small cost while bond 2 carries a medium to large cost. Equity 2 carries a large cost and is not immediately available. Market indices were used to construct the scenarios. Return data on the indices observed during the last credit crisis were used. A moment-matching scenario generation approach, introduced by [5], was used to generate the input scenarios for the optimization problems. This scenario generation method is preferred as it allows the inclusion of worst case scenarios. Table 1 also displays the tradable limits, tiered cost and market depth. The liquidity gap was set to 1 200 000 units of currency for each stage. The numerical example consists of two stages, the first consisting of three days and the second of two days. Both the OLSPP and the OLRPP models were implemented and solved in SAS/OR using PROC OPTMODEL.

	Owned	Price	Stage 1						Stage 2				
			Cost		Market Depth	Available Liquidity			Cost Stage 2		Market Depth	Available Liquidity	
			C0	C1		j=1	j=2	j=3	C0	C1		j=1	j=2
Bond 1	2000	393	20	250	1020	1272	1272	1272	40	450	1020	891	891
Bond 2	1000	385	100	400	2080	779	779	779	200	1000	1300	519	390
Equity 1	100	9633	80	300	31	21	21	21	180	500	16	21	21
Equity 2	50	32016	180	550	10	0	12	12	300	1250	3	5	3

Table 1: Available liquidity and tiered costs

The optimal execution volumes are shown in Table 2. The panel labeled OLLP, referring to the Optimal Liquidity Linear Programming Problem, displays the result from the deterministic model in [4]. We observe that the bond positions bond 1 and bond 2 are sold at stage 2 in full to raise the required funds for the stage 2 liquidity stress. This is expected as the cost of executing this position is smaller than that of the equity position equity 2. The equity position equity 1, with a smaller cost than bond 2 and equity 2, is sold entirely over all stages on account of the volume restrictions associated with this equity. Since the tiered costs for the equity position

equity 2 are much higher in the second stage, the majority of this equity is sold in stage 1, with sales in stage 2 occurring below the market depth.

Model	Instrument	Distress stage 1			Distress stage 2	
		j=1	j=2	j=3	j=1	j=2
OLLP	Bond 1	0	0	218	891	891
	Bond 2	291	0	0	519	190
	Equity 1	21	21	21	21	16
	Equity 2	0	12	10	3	3
OLSPP (expected)	Bond 1	218	0	0	891	891
	Bond 2	486	46	110	325	26
	Equity 1	18	20	20	21	21
	Equity 2	0	8	2	0	1
OLSPP (worst case)	Bond 1	218	0	0	891	891
	Bond 2	383	14	0	213	390
	Equity 1	16	21	21	21	21
	Equity 2	0	10	4	1	3
OLSPP (normal case)	Bond 1	218	0	0	891	891
	Bond 2	666	168	153	0	0
	Equity 1	21	21	21	21	13
	Equity 2	0	4	0	0	0
OLRPP (robust)	Bond 1	246	862	0	891	0
	Bond 2	481	0	0	519	0
	Equity 1	0	21	21	21	16
	Equity 2	0	12	12	5	3

Table 2: Results: Optimal liquidity execution strategies

Three sets of results are displayed for the OLSPP model, namely the expected, the worst case and the normal case and each relating to the different states modeled in the scenario tree. The results for the three cases are comparable to the OLLP results, with the worst case selling off more of equity 2. This suggests that even with high transaction costs, the only way to narrow the liquidity gap is to sell more of equity 2.

The suggested volume of equity 2 to be executed for the OLRPP model shows that it is even more conservative compared to the worst case results from the OLSPP. This is confirmed by the fact that the objective value for the OLRPP is much higher compared to the objective function value for the OLSPP, indicating that higher transactions costs were generated in an attempt to narrow the liquidity gap. This is expected since in the OLRPP formulation, worst case prices are chosen as convex combinations of the the worst case scenario prices, making it very conservative. It should be noted that solving the OLRPP is computationally much more efficient, since its formulation involves fewer variables and constraints.

6 Conclusion

This paper has provided extensions to the flexible model proposed by [4]. Instead of using a "what if" analysis similar to the original OLLP model, our proposed models incorporate uncertainty in terms of scenarios that can be calibrated using past stress period data. These models are easily solved using standard methods and can be used to formulate a best liquidity execution approach in a liquidity distress period, as well as during strategic contingency planning.

ACKNOWLEDGEMENT

This work is based on the research supported in part by the National Research Foundation of South Africa reference number (UID: TP1207243988) and by industry (Absa and SAS Institute).

The Grantholder acknowledges that opinions, findings and conclusions or recommendations expressed in any publication generated by the NRF supported research are that of the author(s), and that the NRF accepts no liability whatsoever in this regard.

References

[1] A. Ben-Tal & A. Nemirovski, 199, Robust solutions of uncertain linear programs, *Operations Research Letters*, 25, pp 1–13

[2] BCBS (Basel Committee on Banking Supervision), 2010, Basel III: International framework for liquidity risk measurement, standards and monitoring, Bank for International Settlements, Switzerland, [WEB:] http://www.bis.org/publ/bcbs188.pdf

[3] BCBS (Basel Committee on Banking Supervision), 2011, Basel III: A global regulatory framework for more resilient banks and banking systems , Bank for International Settlements, Switzerland, [WEB:] http://www.bis.org/publ/bcbs189.pdf

[4] W. Chen, J. Skoglund & L. Cai, 2012, Planning for Optimal Liquidity Execution, *International Review of Applied Financial Issues and Economics*, 4(1), pp 43–46

[5] K. Hoyland & S.W. Wallace, 2001, Generating scenario trees for multistage decision problems, *Management Sci.*, 47(2), pp 295–307

Proceedings of the 2013 ORSSA Annual Conference
pp. 90–101
www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf

ORSSA Proceedings
ISBN
XXX-XXXXX-XXXX-X

Prerequisites for the design of a maritime law enforcement resource assignment decision support system

AA Colmant* & JH van Vuuren†

Abstract

In a maritime law enforcement environment, a coast guard operator is typically required to make counter-threat decisions following the detection and evaluation of threats at sea. These decisions reside within a so-called *resource assignment* process where maritime law enforcement resources, such as patrol vessels, military vessels and armed helicopters, have to intercept vessels that are deemed potential threats. Because the number and nature of potential maritime threats can be overwhelming, it is believed that the quality of resource assignment decisions can be improved by providing maritime surveillance operators with computerized decision support. However, evolving patterns of threat behaviour render the problem less predictable than it might appear, suggesting that the resource assignment process is by no means a repetitive clerical task, but rather that an element of learning and adaptation is present. Following a review of current and recent research into maritime threat evaluation and threat detection systems, the aim of this paper is to propose a generic approach towards the design of an automated decision support system for use by human operators as a tool assisting in maritime law enforcement resource assignment decision making.

1 Introduction

The globally connected economy relies on the seas and adjoined littorals for fishing, access to natural resources (such as oil and gas) and the transportation of most of the world's import and export commodities. Effective governance of these maritime regions has become essential for both the economic growth and the national security of coastal nations [21]. Today, maritime activities embody a very important part of global society, contributing, *inter alia*, to over 90 percent of global trade [2] as well as 91 million tons of food for human consumption annually[1] (compared to 85 million tons in the year 1990) [8]. According to the Maritime Security Sector Reform [21], "the maritime sector is fundamental, directly or indirectly, to the national defense, law enforcement, social, and economic goals and objectives of nearly every country. It is a crucial source of livelihood for many in developing nations, a platform for trade, and a theater for potential conflict or crime."

*Department of Logistics, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: 15427498@sun.ac.za

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

[1]This results in a global average of about 13 *kg* of food per capita per year, which excludes food sources originating from aquacultures.

A sizable portion of these maritime activities are indeed also responsible for many problems, ranging from being detrimental to only a few individuals to harming society on a global scale. These problems are caused by lawless vessels that choose to disrupt the harmony at sea for personal gain. Such activities, or *threats*, typically include piracy acts, illegal and unregulated fishing, narcotics trafficking, illegal immigration, environmental degradation, human trafficking, proliferation of weapons of mass destruction, and terrorism [21]. For instance, it is believed that five to fifteen percent of all large vessels (that is, 5 000 to 7 500 vessels) break the law each year by discharging waste into the seas, including 70 to 210 million gallons of illegal oil waste disposal [14]. Such negligence and inconsideration can potentially devastate the marine environment on a global scale. It is also estimated that a third of all fish populations are over-exploited or have collapsed because of illegal fishing [8]. As a result, some of these species face a constant danger of extinction, and over-exploitation is estimated to generate an indirect annual cost of US $50 billion in lost fishing opportunities [17], which accounts for about half of the value of the global seafood trade. Moreover, it is estimated that the global economic cost resulting from acts of piracy is estimated to lie between US $7 and US $12 billion per annum [3].

Maintaining peace and vigilance at sea requires coastal nations to establish strict maritime laws and regulations, as well as effective monitoring procedures aimed at vessels within their jurisdictions. These procedures are governed by a so-called *resource assignment* process, where, following the detection and evaluation of potentially threatening activities or *events* at sea, vessels, helicopters, and/or seaplanes are dispatched by coast guards to intercept and investigate these threats.

This process can, however, be rather challenging for many coastal nations. Shortages of law enforcement infrastructure, large jurisdiction coverage areas, high operating costs of law enforcement resources, the requirement of using complex threat detection and evaluation systems, scarce maritime intelligence and a lack of adequately trained operators are all factors that contribute to the difficulty of *Maritime Law Enforcement* (MLE) by coastal nations, inevitably affecting their overall ability to counter threats at sea. In addition, the operations carried out by *Maritime Law Enforcement Resources* (MLERs) of coastal nations are typically assigned to perform specific tasks which are distributed in both time and space, making the coordination of these resources, which operate in an adverse and unpredictable environment, a challenging problem. The ability to flexibly adapt to dynamic changes in the availability of MLERs and the services they provide is therefore critical for the success of MLE efforts.

Moreover, the decision making process involved in *Maritime Law Enforcement Resource Assignment* (MLERA) by developing coastal nations is often exclusively conducted on the basis of human judgment, without the use of automated decision support systems [6]. These decisions are usually made by one or more individuals, based on intuition and (often considerable) experience in the MLE domain, in contrast to being the result of an automated, analytical process. The aim in this paper is to identify and discuss the components required in a generic automated model aimed at assisting human operators in resource dispatch decisions within an MLE scenario.

The paper is structured as follows. The internationally recognized legislation and rights associated with the various international maritime zones of coastal nations are first described in §2. A brief literature review of previous work on maritime threat detection, threat evaluation, and resource assignment is then presented in §3. Next, the discussion in §4 seeks to identify the basic components of an MLE environment. A list of objectives that may plausibly be pursued in the design of an MLERA decision support system is then presented in §5. The problem underlying the MLERA process is ultimately that of a dynamic vehicle routing problem, as reasoned in §6. Finally, a dynamic tri-objective generic MLERA model is formulated in §7. This is followed by a brief conclusion in §8, which also contains pointers with respect to future work related to the

research project described in this paper.

2 The law of the seas

The global legal framework defining the rights and responsibilities of coastal nations at sea has, for centuries, been a frustrating, interest-conflicting and unfulfilled idealistic notion. However, in 1982 the *United Nations Convention on Law of the Sea* (UNCLOS) at last settled on a legal order for the seas and oceans with the aim of promoting international communication, peaceful uses of the seas, equitable and efficient utilization of maritime resources, jurisdiction over waters and conservation of maritime ecosystems [6, 21, 24]. Under UNCLOS, coastal nations have the right to establish territorial waters over which they exercise various controls as follows:

- The outer limit of the *territorial sea* of a coastal nation is the curve consisting of every point at a distance of twelve nautical miles from the nearest point of the coastline [24]. The standard baseline for measuring the width of the territorial sea is the low-water line along the coast as marked on large-scale maps officially recognized by the coastal nation[2]. Within this territory, the coastal state may exercise all sovereign rights over the seabed, water and associated airspace, but is obliged to afford the right of innocent passage to vessels of any states.

- The waters contiguous to the territorial sea, known as the *contiguous zone*, extends for twenty four nautical miles from the same baseline from which the width of the territorial sea is measured. Here, the coastal state may exercise the rights to prevent violations of its fiscal, customs, immigration and waste disposal laws [24]. Additionally, the coastal state may also take actions within this zone to punish violations of these laws previously committed within its land territory or territorial sea.

- The *exclusive economic zone (EEZ)* stretches out to sea for a further 188 miles from the end of the territorial sea. Here, the coastal state has the sovereign rights to explore, exploit, conserve and manage living and nonliving natural resources; to establish and use artificial islands, installations, and structures; to conduct marine scientific research; and to protect and preserve the marine environment [24]. UNCLOS is, however, in the process of allowing certain coastal nations to extend specific zones within their EEZ beyond 200 nautical miles, based on certain physical characteristics of the continental shelf [21].

Lastly, the waters beyond the EEZ (which are not considered archipelagic waters) are defined as the *high seas*. The sense of liberty enjoyed by seafarers in these waters is applicable to all states, whether coastal or not. Subjected to certain rules and regulations laid down by UNCLOS and other entities, navigators of the high seas have the right to freedom of navigation; the freedom to lay underwater infrastructure; the freedom to construct artificial islands, installations and structures; the freedom of fishing; and the freedom to perform scientific research [24]. It is, nevertheless, imperative that these waters remain crime-free and that activities in these regions are only aimed toward peaceful ends.

3 A literature review of MLE

Information related to the detection, evaluation and tracking of unfolding events at sea provides most of the input data for the MLERA process. These input data play a critical role in the

[2]In the case of islands situated on atolls or of islands having fringing reefs, the baseline for measuring the width of the territorial sea is the seaward low-water line of the reef.

design of accurate models of events at sea and, as a result, many studies have been devoted to the measurement, understanding and documentation of threat detection, evaluation and vessel tracking data [9, 11, 15, 19, 22, 25]. These publications generally contain work by groups of researchers from different nationalities who, together, possess a large set of diverse skills and accumulated experience with respect to the design of high-performance decision support systems.

The *threat detection* process should provide the MLERA operator with kinematic information, such as the locations and velocities of events at sea, which are required for the assignment decision making process. For example, a so-called Integrated Maritime Surveillance System is described by Ponsford *et al.* [19], which is a shore-based system that detects, tracks, classifies and identifies surface and air targets throughout the EEZ of a coastal nation. In general, the detection process is carried out using high-performance radars[3]. In [11], a so-called Automated Identification System is described, which is a large detection infrastructure including infrared cameras, an airborne platform carrying a radar and various video cameras. The overall functions of this automated system, however, encompass the detection, tracking, identification and classification of multiple targets, as well as the evaluation of their respective levels of threat and the selection of a course of intervention with respect to these threats (*i.e.* MLERA for the interception of these targets).

Analogously to the threat detection process, the *threat evaluation* process should provide the operator with expectations regarding the nature of events at sea, the degree of "oddness" associated with their behaviour (such as deviation from standard trajectories, suspect encounters between vessels at sea or quick in-and-out crossing of the EEZ boundary) as well as expectations with respect to threats that these events may induce. For example, Farina *et al.* [10] categorize events as being either "neutral," "suspect" or "threatening" for objects heading towards a coastal-based structure. They mention that threat evaluation depends on the target type, and is typically based on a deterministic comparison between the target kinematic parameters (namely speed, distance from the coast and course) and certain allowable values defined by tolerance thresholds. Furthermore, in [19], part of the system includes the use of a broad range of sensors that provide target information on an *ad-hoc* basis to distinguish non-cooperative from cooperative targets.

The *resource assignment* process typically comprises a series of complex tasks, each with specific resource capability requirements, that need to be matched with the capabilities of resources available to perform these tasks [7]. Although relatively many technical MLE-related papers have been published in the past, only few exclusively focus on resource allocation problems [5, 7, 13]. For instance, Darby-Dowman *et al.* [5] present an automated decision support system for the US Coast Guard offering an optimal fleet scheduling process in the form of an integer goal programming formulation using a set partitioning model. Their model seeks to optimize cutter scheduling with time windows in a dynamic environment. Furthermore, Malik *et al.* [13] develop a visual analytics system capable of analysing historic response operations and assessing the potential risks in a maritime environment associated with the allocation of US Coast Guard resources. This analysis is based on a real-life environment[4], which involves multiple (real) bases from whence MLERs are dispatched.

4 Components of an MLE environment

Simulating an MLERA environment for a specific coastal nation is built on the foundations of three major physical aspects. These aspects are a geographical information system model of the

[3] A radar's performance is measured by its ability to detect and track targets at long ranges, and to resolve targets in close proximity to one another [19].

[4] The Great Lakes region of North America.

jurisdiction area, defining clear boundaries at sea and incorporating the rules and regulations that are applicable within such boundaries; an MLE fleet of resources comprising one or more types of units, each type possessing a clear, unique set of characteristics; and a set of potential types of threats that may arise within the boundaries of the coastal nation.

4.1 Jurisdiction area

The set of the maritime zones discussed in §2 should be used as a guideline to define the maritime law enforcement boundaries of a coastal nation. These boundaries should therefore at the very least enclose the territorial seas, the contiguous zone and the EEZ as jurisdiction areas, but these areas may be subdivided or refined further, as required by the coastal nation in question. Farina *et al.* [10], for example, simulated a maritime border control environment with an off-limit zone and a warning zone, measured at 20 km and 50 km from the coastal baseline, respectively. The jurisdiction area of a coastal nation may be modelled using shape-lines generated by Geographic Information System software packages such as *ArcGIS* [12] or *R-Programming* [16].

4.2 Law enforcement resources

MLE operations require law enforcement resources capable of neutralizing a variety of threats at sea. These resources perform differently, depending on their physical characteristics (such as size, maximum speed, manpower on board, weapon infrastructure and other defense mechanisms, autonomy at sea, and setup time). Here, the speed of a resource influences its ability to intercept events rapidly (*i.e.* the *MLER response time*), while its weapon infrastructure and other defense mechanisms determines its effectiveness at neutralizing threats. The autonomy of a resource refers to its ability to be self-sufficient at sea for an extended period of time while on a mission, whilst its setup time refers to the time required to prepare the resource for departure on a mission. The optimal resource fleet composition of an MLE agency is usually not addressed as part of the MLERA problem. The set of MLERs at the disposal of a coastal nation is normally assumed fixed and, consequently, the fixed costs associated with the acquisition of MLERs are typically not considered in the problem. It must, however, be noted that setup and operating costs of these resources need to be taken into consideration.

The simulator in [10], for example, incorporates two types of resources: a helicopter and a patrol boat. Here, resource response time is calculated based on the velocity of the target and the resource parameters. The system considers the current MLERs available in order to select an appropriate resource which is dispatched to intercept the target. The resource parameters in [10] are availability, speed, inspection time and departure time.

In general, however, an MLE fleet consists of m resources divided into M types, where there are m_r units of resource type $r \in \{1, 2, \ldots, M\}$, such that $M \leq m$ and $\sum_{r=1}^{M} m_r = m$. Furthermore, it is assumed that the various types of resources are evaluated based on a set of multiple attributes.

4.3 Threatening events

The nature and frequency of threats vary in different regions of the world, and different coastal nations typically face different types of threats at different levels of harm or intensity. The effectiveness of the threat detection and threat evaluation processes plays a critical role in providing accurate and complete input data for an efficient MLERA system. In particular, such input should include the correct positions of all events at sea, inferences with respect to the potential natures of such events as well as the types of threats that they potentially embody. Two important assumptions are made in the MLERA problem. Firstly, it is assumed that the

position and velocity of each event is known at all times (these values may, of course, change over time for any event). Secondly, because the threat evaluation operator cannot always know with certainty the true nature of an event under investigation, a distribution of approximate probabilities matching each event with each type of threat, an unknown type of threat and a false alarm (*i.e.* the possibility that an event does not embody any threat) is assumed to form part of the problem input data.

In general, the detection system of a coastal nation tracks, at any given time τ, $n(\tau)$ events, which are individually matched by means of estimated probabilities to each of $N-2$ known threat classes, an unknown threat class and a false alarm class. If $p_{ih}(\tau)$ is the probability at time τ that event i is a threat in class h, then, of course, $\sum_{h=1}^{N} p_{ih}(\tau) = 1$ for all $i \in \{1, 2, \ldots, n(\tau)\}$. Furthermore, a score $Q_h \in [0, 1]$ is associated with threat class h, representing the priority level assigned by the coastal nation to neutralize threats in this class.

5 The objectives informing MLE response strategy

Since each coastal nation has its own values, preferences and perceptions of the desirability of trade-offs between objectives when dealing with threats at sea, responses following the detection and evaluation of new events at sea typically vary from nation to nation. These responses should, however, be coherent and be carried out according to a pre-determined protocol, based on a set of goals and objectives. Understanding and identifying these fundamental objectives is critical in the design of an MLERA decision support system. For security reasons, coastal nations are typically averse to declaring their MLE response policies and strategies in detail in the open literature. For this reason the current discussion leans towards a generic approach to identifying plausible and realistic, but general objectives during the formulation of MLE response strategies. It is, however, acknowledged that a deep understanding of the specific strategic aims of a coastal nation's MLE efforts is necessary in order to identify a suitable set of fundamental objectives for use in the formulation of that nation's MLE response strategy. These objectives should be derived from the subjective preferences of the specific coastal nation involved.

According to Clemen & Reilly [4], the set of objectives in a multiple objective decision problem should include all relevant aspects of the underlying decision and this set of objectives should be as small as possible (so as to avoid unnecessary computational complexity). Furthermore, the set of objectives should not contain redundant elements and should be decomposable[5]. Finally, the objectives in the set should be clearly distinguishable and the relevant attribute scales should provide a simple way of measuring the performance of alternatives with respect to the objectives.

The following set of fundamental objectives is henceforth proposed to inform the formulation of MLE response strategies in general:

I Maximize the combined so-called *visitation score* of events at sea actually intercepted and investigated, weighted by (a) the probabilities of these events being various types of threats from a pre-specified list and (b) the priorities assigned by the coastal nation to neutralizing various threat types from this list.

II Minimize the combined so-called *delay score* achieved by intercepting and investigating events at sea in minimal time, weighted in the same manner as Objective I above.

III Minimize the total *operating costs* of MLERs assigned to intercept and investigate events at sea.

[5]A set of objectives is decomposable if the decision maker is able to think about each objective easily without having to consider the others simultaneously.

In the above list, Objective I refers to the significance of the expected benefits associated with the successful interception and neutralisation of various types of threats at sea. It is assumed that the level of threat posed by an event at sea is directly proportional to both the probability that the event is actually a threat of a specific type and to the necessity or importance of neutralising such a type of threat. In addition, pursuit of this objective is expected to increase the probability of assigning suitable or highly appropriate MLERs to investigate events that are expected to be highly threatening or compolicated to neutralize. This objective may be expressed as a linear combination of the probabilities of events being the types of threats mentioned in §4.3 and the priority scores associated with neutralizing threats from these types.

The second objective refers to the total time that elapses between the detection and interception of the events at sea that are scheduled for interception. It is assumed that taking long to intercept an event (*i.e.* incurring a long MLE response time) impacts negatively on the success of MLE operations. More specifically, it is assumed that a longer response time increases the chance that tracking operators lose sight of the event or that the vessels involved in the event exit the EEZ before interception. This objective may be expressed as a linear combination of the probabilities of events being the types of threats mentioned in §4.3, the priority scores associated with neutralizing threats from these types and the expected response times of the MLERs assigned to investigate the events.

Finally, Objective III refers to the costliness of committing MLERs to investigating events at sea. It is assumed that an MLER incurs a fixed setup cost every time it is dispatched from its base and a variable operating cost which depends on the overall distance traveled on a mission at sea.

Objectives I and II share a certain degree of inter-dependency. It should, however, be observed that these two objectives complement each other in a manner that would not be possible if they were to be pursued in isolation. On the one hand, solely adopting Objective I may lead to increased levels of interception of those events at sea expected to embody highly dangerous threats, but will not account for the urgency factor involved when planning the trips of the MLERs assigned to carry out these interceptions. On the other hand, solely adopting Objective II will lead to the situation where no events at sea are actually intercepted (the situation of a zero delay score) and would analogously benefit Objective III as well.

6 A dynamic vehicle routing problem

The MLERA problem can, in essence, be formulated as a special kind of *vehicle routing problem* (VRP) in which the depot represents the base from whence MLERs are dispatched, the fleet of vehicles represents the fleet of MLERs at the disposal of the coastal nation and the customers represent the events tracked at sea. The MLERA problem, however, differs from standard capacitated VRP formulations in the literature as a result of the following observations:

1. The fleet of vehicles is not necessarily homogeneous,
2. Servicing a customer is optional[6],
3. Assigning a vehicle a route (*i.e.* utilising it at all) is optional,
4. Certain types of vehicles are not suitable to visit certain types of customers,
5. Certain types of vehicles are favoured to visit certain types of customers,

[6]VRPs of this kind are known as VRPs with profits [1].

6. Vehicles are not subjected to capacity restrictions,
7. Idle vehicles do not have to be stationed at the depot,
8. The number of customers is not fixed and neither are their locations,
9. The distance between any two vertices is not necessarily the same in both directions[7],
10. An arc between any two vertices is measured as the shortest possible path linking them aquatically (except in the case of aerial MLERs), and
11. A vehicle may start out along a route that does not originate or end at the depot.

In the above list, Observations 1–3 follow from the discussions in previous sections and are incorporated in certain variants of the VRP in the literature. Observation 4 places the restriction that certain types of MLERs should preferably not be scheduled to intercept events that are suspected to be of a certain type, as their infrastructure characteristics may not be effective enough to successfully neutralize threats of that type. Similarly, taking into account the unique properties that are regarded to be advantageous in successfully neutralizing threats of certain types, Observation 5 encourages certain kinds of MLERs to intercept events that are suspected to be threats of these types. Observation 6 stresses that the MLERs are typically not restricted to a capacity constraint (in terms of delivering commodities to customers), but are rather constrained in terms of maximum distances travelled or lengths of time associated with the routes of the vehicles as a result of various practicalities such as the need to refuel or resupply materials or ammunition, the need to perform routine MLER maintenance and the need to allocate leave or periods of rest to crew members[8]. Observation 7 allows for the resources to patrol at sea in strategic locations until assigned to a specific interception route, while Observation 8 acknowledges the kinematic nature of events at sea, calling for interception points to be calculated. Observation 9 suggests that MLER travel directions are influenced by velocities of events, ocean currents and adverse meteorological conditions, while Observation 10 acknowledges that the length of a shortest path between two events is in most cases not the Euclidean distance between these events. Finally, Observation 11 focuses attention to the fact that an MLER may be diverted to a new route while out at sea (*i.e.* it is not required that an MLER should first return to the base prior to starting a new route; *e.g.* see Observation 7).

Events at sea occur in a stochastic manner as a function of time. Consequently, once an acceptable solution to the MLERA problem has been found, it is merely a matter of time before a disturbance in the MLE environment occurs, requiring the situation at hand to be reconsidered. The MLERA problem therefore resides in the class of so-called *dynamic (or stochastic) VRPs*. Input data to problem instances in this class are made known to the operator/decision maker in a continual fashion and are updated concurrently with the determination of the set of vehicle routes [18, 20]. In other words, the input data (involving the detection of events) are not known in their entirety ahead of time, but are rather revealed as time goes by. The operator must therefore solve part of the problem on the basis of the information currently available, and then resolve part of the problem as new input data are revealed. It follows that the trips generated for the MLERs to intercept and investigate events need to be reconsidered every time the situation at sea changes, as parts of the MLERA problem solution which have not yet realised (*i.e.* planned MLER routes which have not yet been completely traversed) might no longer be feasible or preferred.

Dynamic VRPs require making decisions in which a compromise or trade-off is sought between the speed of generating a solution and the overall solution quality; that is, the time invested in

[7]VRPs of this kind are known as Asymmetric VRPs [23].
[8]VRPs of this kind are known as Distance-Constrained VRPs [23].

searching for good decisions comes at the price of less reactiveness to changes in the input data [18]. In terms of performance, it is therefore critical to design a model and identify a search technique that produces satisfactory solutions quickly after having received an update in input data. In this particular problem, the amount of time taken between the detection and evaluation of new events, and the generation of a new, preferred solution may increase the MLER response times (consequently having a negative impact on Objective II), while increasing the chances of detours (consequently having a negative impact on Objective III), or reaching some events too late (consequently having a negative impact on Objective I). A solution search technique that can *quickly* generate *near-optimal* solutions is therefore preferred to one that *slowly* generates *optimal* solutions, as the former technique would most probably still outperform a human operator in the same amount of time.

7 Formulation of an MLERA model

In the linear formulation of the MLERA problem in this section, the aim is to assign a subset of fixed MLERs to a feasible set of routes intercepting a subset of events at sea so as to optimize the objectives proposed in §5 while respecting the observations listed in §6. However, in this formulation, Observations 7 and 9 are not (yet) addressed.

Define, in accordance with the dynamic VRP, the notion of a *time stage* τ as a temporal interval that only exists while the current MLE situation at sea remains unchanged. Different kinds of occurences may trigger the beginning of a new time stage, such as the detection of a new event at sea, a sudden change in the velocity of an event or changes in environmental vectors.

Henceforth define, at the beginning of any time stage τ, a complete graph $G(V(\tau), E(\tau))$ whose vertex set $V(\tau) = \{0, 1, \ldots, n(\tau)\}$ represents the base (vertex 0) and the events scheduled to be intercepted (vertices 1 to $n(\tau)$), and whose edge set $E(\tau)$ contains the pre-calculated arcs linking any two vertices in $V(\tau)$ in both directions. Furthermore, define $U = \{1, \ldots, m\}$ as the set of MLERs (as described in §4.2) and $Z = \{1, \ldots, N\}$ as the set containing all plausible scenarios regarding the nature of events at sea (as described in §4.3).

Let $d_k(\tau)$ be the maximum distance that MLER k may travel at sea from the beginning of time stage (τ), and let $d'_k(\tau)$ be the maximum time that MLER k may spend at sea from the beginning of time stage (τ). Additionally, define a cost c_k incurred when preparing MLER k for departure on a mission, which may be expressed as a step-function for the costs associated with dispatching it from the base (*i.e.* if it was idle at the end of time stage $\tau - 1$), the costs associated with deviation from its current route (*i.e.* if it was not idle at the end of time stage $\tau - 1$ and was required to change its initial trajectory at the beginning of time stage τ) and a zero cost associated with no deviation from its current route (*i.e.* if it was not idle at the end of time stage $\tau - 1$ and was not required to change its initial trajectory at the beginning of time stage τ). In order to address Observations 4 and 5 of §6, it is necessary to establish parametric values causing MLERs to be assigned according to their designed purposes. So, let $W_{kh} \in [0, 1]$ be the score associated with the efficiency of MLER k in terms of neutralizing a class h threat.

For any given feasible solution, let $D_{ijk}(\tau)$ be the length of arc $(i, j)(\tau)$ traversed by MLER k and set $D_{iik}(\tau) = +\infty$ for all vertices in $V(\tau)$. Similarly, let $C_{ijk}(\tau)$ be the cost associated with MLER k traversing arc $(i, j)(\tau)$ and set $C_{iik}(\tau) = +\infty$ for all vertices in $V(\tau)$. If it is assumed that an MLER k maintains a fixed average speed of α_k between any two events that it visits at sea, the time that MLER k spends traversing the arc $(i, j)(\tau)$ may be expressed as the quotient of $D_{ijk}(\tau)$ and α_k. Furthermore, let $s_{ik}(\tau)$ be the expected *service time* that MLER k spends investigating event i and let $t_{ik}(\tau)$ be the expected response time that MLER k takes to visit event i, provided that it is scheduled for investigation during time stage τ. Finally, let $p_{ih}(\tau)$

and Q_h be as defined in §4.3, and define the binary variables

$$x_{ijk}(\tau) = \begin{cases} 1, & \text{if MLER } k \text{ is scheduled to traverse arc } (i,j)(\tau) \\ 0, & \text{otherwise,} \end{cases}$$

$$\text{and} \qquad y_k(\tau) = \begin{cases} 1, & \text{if MLER } k \text{ is utilised during time stage } \tau \\ 0, & \text{otherwise.} \end{cases}$$

The aim in this tri-objective dynamic VRP is to

$$\text{maximise} \sum_{\substack{i \in V(\tau) \\ i \neq 0}} \sum_{j \in V(\tau)} \sum_{k \in U} x_{ijk}(\tau) \sum_{h \in Z} Q_h W_{kh} p_{ih}(\tau) \tag{1}$$

$$\text{minimise} \sum_{\substack{i \in V(\tau) \\ i \neq 0}} \sum_{j \in V(\tau)} \sum_{k \in U} t_{ik}(\tau) x_{ijk}(\tau) \sum_{h \in Z} p_{ih}(\tau) Q_h \tag{2}$$

$$\text{minimise} \sum_{k \in U} c_k y_k(\tau) + \sum_{i \in V(\tau)} \sum_{j \in V(\tau)} C_{ijk}(\tau) x_{ijk}(\tau) \tag{3}$$

subject to the constraints

$$1 - y_k(\tau) \leq n(\tau) w_k(\tau), \qquad k \in U, \tag{4}$$

$$\sum_{i \in V(\tau)} \sum_{j \in V(\tau)} x_{ijk}(\tau) \leq n(\tau)(1 - w_k(\tau)), \quad k \in U, \tag{5}$$

$$\sum_{i \in V(\tau)} \sum_{k \in U} x_{ijk}(\tau) \leq 1, \qquad j \in V(\tau) \backslash \{0\}, \tag{6}$$

$$\sum_{j \in V(\tau)} \sum_{k \in U} x_{ijk}(\tau) \leq 1, \qquad i \in V(\tau) \backslash \{0\}, \tag{7}$$

$$\sum_{i \in V(\tau)} x_{ijk}(\tau) - \sum_{\ell \in V(\tau)} x_{j\ell k}(\tau) = 0, \qquad \begin{matrix} j \in V(\tau) \backslash \{0\}, \\ k \in U, \end{matrix} \tag{8}$$

$$\sum_{i \in V(\tau)} \sum_{k \in U} x_{i0k}(\tau) = \sum_{k \in U} y_k(\tau), \tag{9}$$

$$\sum_{i \in V(\tau)} \sum_{k \in U} x_{0ik}(\tau) \leq \sum_{k \in U} y_k(\tau), \tag{10}$$

$$\sum_{i \in V(\tau)} \sum_{j \in V(\tau)} D_{ijk}(\tau) x_{ijk}(\tau) \leq d_k(\tau), \qquad k \in U, \tag{11}$$

$$\sum_{\substack{i \in V(\tau) \\ i \neq 0}} s_{ik}(\tau) x_{ijk}(\tau) + \sum_{\ell \in V(\tau)} \sum_{\ell' \in V(\tau)} \frac{D_{\ell\ell' k}(\tau)}{\alpha_k} x_{\ell\ell' k}(\tau) \leq d'_k(\tau), \qquad \begin{matrix} j \in V(\tau), \\ k \in U, \end{matrix} \tag{12}$$

$$w_k(\tau), x_{ijk}(\tau), y_k(\tau) \in \{0,1\}, \qquad \begin{matrix} i \in V(\tau), \\ j \in V(\tau), \\ k \in U. \end{matrix} \tag{13}$$

In the model formulation above, constraint sets (4) and (5) prevent any arcs to be traversed by an MLER that is not scheduled for utilization during time stage τ, where $w_k(\tau)$ is a linking constraint variable. Constraint sets (6) and (7) ensure that no more than one arc may enter

or leave any particular event along an MLER route (see Observation 2 in §6), while constraint set (8) ensures that, if an event is intercepted by a certain MLER during time stage τ, then the MLER must depart from that event after investigation. Constraint (9) ensures that the number of MLERs utilised during time stage τ coincides with the number of arcs entering the base (*i.e.* all routes are initially scheduled to end at the base), while constraint (10) ensures that no more MLERs leave the base during time stage τ than are utilised during that time stage (see Observation 11 in §6). Furthermore, constraint set (11) imposes an upper bound on the maximum distance traveled by MLER k from the beginning of time stage (τ), while constraint set (12) imposes an upper bound on the maximum length of time that MLER k spends at sea from the beginning of time stage (τ). Finally, constraint set (13) enforces the binary nature of the decision variables in the model.

8 Conclusion

The objective in this paper was to identify and elucidate the components required in the design of a generic automated model aimed at assisting human operators in resource dispatch decisions within an MLE environment. Due to the scarcity of related studies in the open literature and MLE professional assistance from external sources, it is difficult to evaluate the fundamental completeness and realism of the MLE model proposed in this study *versus* that of other MLERA studies. It is, however, believed that enough progress has been made at this point toward the formulation of such a model to allow for the design of an MLERA simulation experiment aimed at testing the effectiveness of the proposed MLE model within a specific coastal environment.

The next phase in this research project is to design more advanced models incorporating additional MLERA aspects, and to implement one or more search methods for solving these models (approximately) within some MLE scenario. It is expected that the use of metaheuristic search techniques (such as simulated annealing or a genetic algorithm) will generate relatively good solutions in limited computational time.

References

[1] Archetti C, Speranza MG & Vigo D, 2012, *Vehicle routing problems with profits*, [Technical Report], Department of Economics and Management, University of Brescia, Brescia.

[2] Borgerson SG, 2009, *The national interest and the law of the sea*, Council Special Report No. 46, Council on Foreign Relations Press, Washington (DC).

[3] Bowden A, 2010, *The economic costs of maritime piracy*, One Earth Future Foundation, [Working Paper], Louisville (CO).

[4] Clemen RT & Reilly T, 2001, *Making hard decisions with decision tools*, Duxbury, Belmont (CA).

[5] Darby-Dowman K, Fink RK, Mitra G & Smith JW, 1995, *An intelligent system for US Coast Guard cutter scheduling*, European Journal of Operational Research, **87**, pp. 574–585.

[6] Du Toit J, 2013, Stellenbosch University, [Personal Communication], Contactable at `jacques@dip.sun.ac.za`.

[7] Farahbod R, Glasser U & Khalili A, 2009, *A multi-layer network architecture for dynamic resource configuration & management of multiple mobile resources in maritime surveillance*, SPIE Defense, International Society for Optics and Photonics, **7345**, pp. 734508–734508.

[8] Food and Agricultural Organization (FAO), 2010, *The state of world fisheries and aquaculture*, Food and Agriculture Organization of the United Nations, Rome.

[9] Gad A & Farooq M, 2002, *Data fusion architecture for maritime surveillance*, Proceedings of the Fifth IEEE International Conference on Information Fusion, **1**, pp. 448–455.

[10] Farina A, Giompapa S, Gini F, Graziano A & Di Stefano R, 2007, *Computer simulation of an integrated multi-sensor system for maritime border control*, Proceedings of the IEEE Radar Conference, Boston (MA).

[11] Giompapa S, Gini F, Farina A, Graziano A, Croci R & Di Stefano R, 2009, *Maritime border control multisensor system*, IEEE Aerospace and Electronic Systems Magazine, **24(8)**, pp. 9–15.

[12] Harlow M, 2005, *ArcGIS reference documentation*, Environmental Systems Research Institute (ESRI), Redlands (CA).

[13] Malik A, Maciejewski R, Jang Y, Oliveros S, Yang Y, Maule B, White M & Ebert DS, 2011, *A visual analytics process for maritime resource allocation and risk assessment*, Proceedings of the IEEE Conference on Visual Analytics Science and Technology, Providence (RI).

[14] Marine Defenders, 2012, *Marine defenders project*, [Online], [Cited April 24th, 2013], Available from `http://www.marinedefenders.com/oilpollutionfacts/illegal.php`.

[15] Moore K, 2005, *Predictive analysis for naval deployment activities*, [Online], [Cited February 13th, 2013], Available from `https://www.fbo.gov/utils/view?id=4b9eec01c0278620e50704ed51e24785`.

[16] Murison R, 2005, *R-Statistical and graphical software notes*, School of Mathematics, Statistics and Computer Science, University of New England, Armidale (NSW).

[17] Pascoa S, 2012, *The sunken billions: The economic justification of fisheries reform*, Marine Resource Economics, **27(2)**, pp. 193–194.

[18] Pillac V, Gendreau M, Guéret C & Medaglia AL, 2012, *A review of dynamic vehicle routing problems*, [Technical Report], Centre Interuniversitaire de Recherche sur les Reseaux d'Entreprise, la Logistique et le Transport, Montréal.

[19] Ponsford AM, Sevgi L & Chan HC, 2001, *An integrated maritime surveillance system based on high-frequency surface-wave radars, part 2: Operational status and system performance*, IEEE Antennas and Propagation Magazine, **43(5)**, pp. 52–63.

[20] Psaraftis HN, 1995, *Dynamic vehicle routing: Status and prospects*, Annals of Operations Research, **61**, pp. 143–164.

[21] Sandoz JF, 2012, *Maritime security sector reform*, United States Institute of Peace, [Government Report], Washington (DC).

[22] Scheepens R, Willems N, van de Wetering H & van Wijk JJ, 2011, *Interactive visualization of multivariate trajectory data with density maps*, Proceedings of the IEEE Pacific Visualization Symposium, pp. 147–154.

[23] Toth P & Vigo D, 2005, *The vehicle routing problem*, SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics, Camden (PA).

[24] Treves T, 2008, *United Nations Convention on the Law of the Sea*, [Online], [Cited April 29th, 2013], Available from `https://www.un.org/depts/los/convention_agreements/texts/unclos/UNCLOS-TOC.htm`.

[25] Wehn H, Valin P, Guitouni A, Bosse E, Dlugan A & Zwick H, 2006, *A distributed information fusion testbed for coastal surveillance*, Proceedings of the Tenth International Conference on Information Fusion, Quebec City.

Proceedings of the 2013 ORSSA Annual Conference
pp. 102–111
www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf

ORSSA Proceedings
ISBN
xxx-xxxxx-xxxxx-x

A Study of Cell Depletion in the Developmental Approach for the Uncapacitated Examination Timetabling Problem

Christopher Rajah[1] Nelishia Pillay[2]

Abstract

Quite often it becomes useful to turn to nature to solve real world problems including combinatorial optimisation problems. Examples of these types of problems include educational timetabling. The developmental approach (DA) is one of many biologically inspired methods successfully applied to examination timetabling. Organisms are developed in nature through processes of cell biology. The DA closely mimics these processes in developing an organism representing an exam timetable. These processes are cell division, cell creation, cell interaction and cell migration. This paper introduces a new process called cell depletion. A case is made for the inclusion of a cell depletion operator in the DA. The performance of this revised version (DAD) is compared to other versions of the DA and other biologically inspired methods in solving problems from the Carter benchmark. The DAD generally performed better than the DA. Furthermore, its performance was comparative to that of methods producing the best results and other biologically-inspired methods, outperforming the latter in some instances.

Key words: developmental approach, examination timetabling, cell depletion

1 Introduction

The examination timetabling problem is a scheduling problem where a fixed number of exams needs to be allocated to a fixed number of periods. The scheduling process is subject to certain problem specific constraints. The constraints can be hard or soft constraints. An exam timetable is only considered feasible it satisfies all the problem hard constraints. An example of a hard constraint is that the timetable must not contain any exam clashes, i.e. no student must be

[1]School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, email: CHRISTOPHER.RAJAH@kzntreasury.gov.za
[2] School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, email: pillayn32@ukzn.ac.za

scheduled to write two examinations at the same time.. Soft constraints do not affect the feasibility of the timetable but do influence the quality of the timetable, for example, exams must be evenly spaced for students or some exams must be written in the morning. A cost function is used to measure the quality of the timetable, for example, a proximity function may score timetables with exams that are more evenly spread as being better than timetables with no free periods between exams. The aim is to develop a timetable that is both feasible and as fair as possible to all students. A higher quality timetable also influences the pass rate of students because they are given more time to study between exams.

There are two versions of the examination timetabling problem, namely, the capacitated version and the uncapacitated version. In the capacitated version we take into consideration the size of the venue required to write the exam. This may be represented as a hard constraint to the problem as it may be required that the room reserved for the exam must be large enough to accommodate all students and must be available when that exam is scheduled. In the uncapacitated version, room capacitates are not taken into consideration. A survey of the various methods that have been applied to the examination timetabling problem is presented in [14].

Carter et al. [7] introduced the Toronto benchmark set to allow for the evaluation and comparison of the different techniques used to solve the examination timetabling problem. The method employed by Caramia [6] has produced the best known results for four of the problems in this benchmark set. The authors make use of a greedy scheduler to schedule exams. A penalty-decreaser is used to reduce costs by changing the timeslots of exams. A penalty-trader also decreases costs by adding additional timeslots one at a time. Abdullah et al. [1] have the best results for three of the Carter problems. The authors combined the great deluge with an electromagnetic-like mechanism. A timetable is represented by a particle. The charge of the particle represents the cost of the timetable. The greater the charge the better the quality of the timetable produced. The electromagnetic-like mechanism is used to calculate a force of the particle or timetable. This value is then used to determine parameters required for the great deluge algorithm. Burke et al. [4] have the best result for one of the Toronto problems. In this study the great deluge algorithm is extended by making use of a coefficient which allows the original deluge algorithm to share characteristics with that of a greedy hill climber. The coefficient is a ratio of the degree of the exam to the maximum degree of the graph. The late acceptance algorithm by Burke et al. [5] has produced the best results for two of the problems. The authors make use of a variation of the classic hill climbing method. The current solution is accepted if it's score is less than or equal to the best score L moves ago. A larger L value results in longer runtimes but better quality solutions.

Many biologically-inspired techniques have also been used to solve the examination timetabling problem. Those biologically-inspired methods that have been applied to the Carter benchmark set are discussed here to allow for performance comparison with the DAD in section 5. Cote et al. [8] applied a hybrid multi-objective evolutionary algorithm (hMOEA) to the examination timetabling problem which has four phases. The algorithm is applied to an archive and population of non-dominated timetables. The first phase uses local search operators to remove hard constraints and reduce proximity costs. The second phase ranks the population and archive and the third phase replaces weaker elements of the archive. The last phase mutates the population by changing the timeslot of randomly chosen exams and then uses tournament

selection to create the next generation. This method has produced the best result for one of the Toronto datasets. Eley [9] looked at an ant system (MMAS) and showed that by employing hill climbing as a local search method and exchanging information by means of interaction between ants, improved results. Alzaqebah et al. [3] applied an artificial bee colony algorithm to the exam timetabling problem with promising results. The algorithm was introduced by Karaboga et al. [10] and combines both local and global search methods to mimic the behaviour of honey bees in maximizing nectar quantities and sharing food sources with other bees. A feasible solution is first developed using a saturation degree heuristic and backtracking algorithm. Thereafter, the solution is improved using an artificial bee colony algorithm. Pillay et al. [13] make use of an informed genetic algorithm to solve the Carter benchmark set of problems. Genetic algorithms are used both in the generation of the timetable and in the improvement of the timetables. The work introduces a new low-level heuristic and allows for the comparison of a genetic algorithm approach with other methods using the Carter benchmarks.

The developmental approach [12] was introduced as another biologically-inspired method for the examination timetabling problem. In later work the DA was revised to improve its performance and runtimes [11]. The following section describes the most recent version of the DA. Section 3 introduces cell depletion as a possible extension to the DA. Section 4 describes the methodology used to evaluate the performance of the DA. Section 5 discusses the performance of the DA incorporating depletion and section 6 concludes the study.

2 The Developmental Approach

This section describes the DA and its implementation in more detail. The DA has its roots in nature. The algorithm is used to develop an organism in the same way nature does through the processes of cell creation, cell division, cell migration and cell interaction. In the context of exam timetabling, the fully grown organism represents an exam timetable and each cell in the organism represents a timeslot or period in the timetable. Algorithm 1 is the algorithm for the developmental approach presented in [11].

```
Procedure Create Organism
Begin
  Sort the exams to be allocated according to saturation degree
  Create a single cell
  Allocate the exam with lowest saturation degree to the cell
  Repeat
    Resort the remaining exams
    If there are two or more cells perform cell migration
    If there is a feasible cell available
      Add the exam to the cell
  Else if the cell limit is not reached
      Perform cell division
  Else
     Allocate the exam to a random cell
  Perform Cell interaction
 Until (all exams have been scheduled)
End
```

Algorithm 1. Developmental Approach Algorithm [11]

A saturation degree heuristic is used to sort all unallocated exams. It measures the number of feasible cells that an exam can be allocated to. Exams with the least number of options, i.e. lower saturation degree, are allocated first. The current growth stage of the organism influences the application of the heuristic. Therefore after every allocation, the saturation degree of the remaining unallocated exams is recalculated and the exams resorted. At the beginning of the algorithm a single cell is created and allocated a randomly selected position in the timetable, i.e. a period. Exams are then allocated to the cell one at a time. If an allocation causes a hard-constraint violation cell division occurs. This involves the current cell dividing into two with one cell containing the examination causing the violation and the second cell containing the remaining exams. In this way the number of cells increases until the total permissible number of cells is reached. The organism is not considered fully grown until all exams have been allocated. The algorithm does not always induce feasible timetables. To improve the quality of the timetable two processes are used namely cell migration and cell interaction. In cell migration, the positions of two randomly chosen cells are swapped if the swap results in a reduction in the soft constraint cost and no hard constraint is violated. In cell interaction, randomly selected exams from two randomly chosen cells are swapped if the swap results in a reduction in the soft constraint cost and no hard constraint violations.

3 Introduction of Cell Depletion into the DA

This section introduces the cell depletion operator and describes how it is incorporated into the DA. During organism growth, programmed cell death (PCD) [2] may be used to remove unwanted cells from a variety of tissue. PCD is carefully regulated so that the fate of an individual cell meets the needs of the organism as a whole. For instance, once the organism is fully is grown, PCD is needed to balance cell proliferation and maintain constant cell numbers in tissue. Other examples where PCD is used in cell biology include: the elimination of tissue between the fingers and toes during their formation. It may also be used to kill cells infected with a virus. Therefore, programmed cell death is intentional and necessary for the growth of an organism. PCD may improve an organism's fitness. This improvement is problem dependent, for example in the context of examination timetabling removing a cell could reduce the hard and/or soft constraint cost. This is different from unintentional cell death, for example as a result of an accident.

In nature cells must proliferate before death can take place. Thus, cell depletion will take place after each cell interaction in the DA. As is the case with cell migration and cell interaction cell depletion is evoked on every iteration during organism development. Cell depletion does not take place if there are no cells in the organism. Algorithm 2 is the pseudo-code for the cell depletion operator.

In the beginning a random integer between 0 and the value r is generated. The linear congruential method is used to generate the random number. The depletion operator is only evoked if r is 0. This is done to avoid the operator from being called on every iteration as this would cause a single allocation and deallocation after each cell interaction resulting in an infinite loop. The value of r dictates how often the operator is called. If, for example, $r = 5$ then the operator has a 1 in 5 chance or 20% chance of being called. Therefore, for a timetable of 100 exams the operator would be called on average 20 times.

```
Procedure Cell Depletion (r)
Begin
  option = generate a random integer in the range 0 to r
  If (number of cells in organism >=1 and option =0)
    Randomly choose a cell.
    Randomly choose an exam from the chosen cell and  remove it from the cell and
    add it to the list of unallocated exams.
  If after removing an exam from a cell, it becomes empty, then remove the cell from
  the organism and make its position available for reallocation.
End
```

Algorithm 2. *Pseudo Code for the Depletion Operator*

The number of exams affects the number of times the procedure is called. It is hypothesized that removing an exam from a cell has the following benefits:

1. There is a chance that if the exam is allowed to be reallocated at a later stage in growth of the organism it may lead to a better quality timetable.
2. The removal of an exam from a cell may allow for the allocation of a different exam to the same cell which could lead to a better solution. If that exam was allowed to remain, then the allocation of the new exam may have resulted in a hard constraint violation and a non-feasible solution.
3. Empty cells are removed as well. This makes sense, as cells should only be allowed to exist if they have exams allocated to them. By deleting the cell, the position it occupies is freed which may allow for a better solution later on in the development of the organism.

4 Methodology

The revised system was applied to twelve of the Carter benchmark problems [7]. The Carter benchmark data sets consists of real-world problems which have generally been used to test and compare different methodologies applied to the uncapacitated examination timetabling problem. The hard constraint for this set of benchmarks is that there are no clashes between exams for students. The soft constraint is that the exams are well spaced. The spacing of exams is calculated using the following equation:

$$\frac{\sum w(|e_i - e_j|)N_{ij}}{S} \qquad (1)$$

where:

1) $|e_i - e_j|$ is the distance between the periods of each pair of examinations (e_i, e_j) with common students.
2) N_{ij} is the number of students common to both examinations.
3) S is the total number of students
4) w(1) = 16, w(2) = 8, w(3) = 4, w(4) = 2, w(5) = 1, w(n) =0 for n >5, i.e. the smaller the distance between periods the higher the weight allocated.

Details of the benchmark set are listed in Table 1. The density of the conflict matrix is an estimate of the problem difficulty and is calculated to be the number of potential student clashes over the total number of students.

Table 1: Carter Benchmarks

Problem	**Institution**	**Periods**	**No. of Exams**	**No. of Students**	**Density of Conflict Matrix**
car-f-92	Carleton University, Ottawa	32	543	18419	0.14
car-s-91	Carleton University, Ottawa	35	682	16925	0.13
ear-f-83	Earl Haig Collegiate Institute, Toronto	24	190	1125	0.27
hec-s-92	Ecole des Hautes Etudes Commerciales, Montreal	18	81	2823	0.42
kfu-s-93	King Fahd University of Petroleum and Minerals, Dharan	20	461	5349	0.06
lse-f-91	London School of Economics	18	381	2726	0.06
rye-s-93	Ryerson University, Toronto	23	486	11483	0.08
sta-f-83	St Andrew's Junior High School, Toronto	13	139	611	0.14
tre-s-92	Trent University, Peterborough, Ontario	23	261	4360	0.18
uta-s-92	Faculty of Arts and Sciences, University of Toronto	35	622	21266	0.13
ute-s-92	Faculty of Engineering, University of Toronto	10	184	2749	0.08
yor-f-83	York Mills Collegiate Institute, Toronto	21	181	941	0.29

5 Results and discussion

As is typical in the evaluation of methods applied to the examination timetabling problem, the performance of the DAD is assessed in terms of the feasibility and quality of the timetables produced by the DAD. The DAD generated feasible solutions for all 12 problems in Table 1. For each value of r, ten runs were conducted for each problem. In each run three hundred organisms were generated. A different random number generator seed was used in the creation of each organism. The values of r used are 5, 10, 15, 20, 30. Please note that r is a parameter value for the DA in the same way that values like population size and genetic operator probabilities are parameter values for a genetic algorithm and as such need to tuned for each problem domain. Table 2 lists the best soft constraint cost over all ten runs. The best results are in bold.

Table 3 performs an empirical comparison of the runtimes of the DAD and the DA [11]. The runtimes listed for each problem is that for the run producing the best result.

Table 2: Performance of DAD with Different Values of k

Problem	**5**	**10**	**15**	**20**	**30**
car-f-92	4.09	**4.04**	4.10	4.12	4.09
car-s-91	4.87	4.93	**4.85**	4.88	4.95
ear-f-83	**34.85**	35.15	35.12	35.16	35.18
hec-s-92	10.83	10.86	10.93	**10.78**	10.98
kfu-s-93	14.02	13.94	13.91	**13.84**	13.92
lse-f-91	10.58	10.56	**10.48**	10.59	10.53
rye-s-93	9.16	9.16	9.17	**9.06**	**9.06**
sta-f-83	157.22	157.17	157.22	**157.13**	157.17
tre-s-92	8.36	**8.27**	8.36	8.40	8.36
uta-s-92	**3.29**	3.30	**3.29**	**3.29**	**3.29**
ute-s-92	26.30	**26.03**	26.31	26.26	26.17
yor-f-83	**37.38**	38.27	38.88	38.23	37.91

Table 3: Comparison of Runtime of DAD and the Revised DA from [11]

Problem	**DA [11]**	**DAD**
car-f-92	1 hr 11 mins	1 hr 19 mins
car-s-91	2 hrs 11 mins	1 hr 53 mins
ear-f-83	6 mins	1 min
hec-s-92	1 min	1 min
kfu-s-93	35 mins	41 mins
lse-f-91	22 mins	17 mins
rye-s-93	44 mins	1 hr 32 mins
sta-f-83	3 mins	1 min
tre-s-92	11 mins	3 mins
uta-s-92	1 hr 41 mins	1 hr 50 mins
ute-s-92	4 mins	3 mins
yor-f-83	5 mins	7 mins

Table 4 compares the performance of the DAD with the current best known results for the benchmark set. It can be seen that while the DAD does not produce a best result for the Carter benchmark set, its performance is comparative to that of the best performing methods. Table 5 compares the results obtained by the DAD with that of other biologically-inspired methods and the revised DA. From Table 5 the DAD appears to perform better than the DA with the DAD producing better results for 9 of the 12 problems. For the remaining 3 problems the percentage difference in the results produced by the DAD compared to the DA is 1.04%, 0.12% and 1.53%. The DAD produces the best results for three of the datasets when compared to the other four

methods. Hypothesis tests were conducted to determine the significance of the result that the DAD performs better than the DA. The null hypothesis is that DAD performs just as well as the DA. The alternate hypothesis is that DAD performs better. Fifty runs were performed for each problem. The result that DAD performs better than DA for the car-f-92, car-s-91, kfu-s-93 and yor-f-83 problems was found to be significant at the 1% level of significance and at the 10% level of significance for the ear-f-83, hec-s-92, lse-f-91, rye-s-93, sta-f-83, tre-s-92, uta-s-92 and ute-s-92 datasets.

Table 4: Comparison of the DAD Performance with the Best Known Results

Problem	**DAD**	**Best Known**	**% Distance from Best Known**	**Source of Best Known**
car-f-92	4.04	3.76	7.18	Abdullah et al. [1]
car-s-91	4.85	3.74	25.84	Burke et al. [4]
ear-f-83	34.85	29.3	17.3	Caramia et al. [6]
hec-s-92	10.78	9.2	15.82	Caramia et al. [6]
kfu-s-93	13.84	12.62	9.22	Abdullah et al. [1]
lse-f-91	10.48	9.6	8.76	Caramia et al. [6]
rye-s-93	9.06	6.8	28.5	Caramia et al. [6]
sta-f-83	157.13	156.94	0.12	Abdullah et al. [1]
tre-s-92	8.27	7.72	6.88	Burke et al. [5]
uta-s-92	3.29	2.99	9.55	Abdullah et al. [1]
ute-s-92	26.03	24.4	6.46	Caramia et al. [6]
yor-f-83	37.38	34.78	7.21	Burke et al. [5]

Table 5: Comparison of the DAD Performance with that of Other Biologically-Inspired Methods

Problem	**DAD**	**DA [11]**	**IGA [13]**	**hMOEA [8]**	**MMAS[9]**
car-f-92	**4.04**	4.1	4.22	4.2	4.8
car-s-91	4.85	**4.8**	4.92	5.4	5.7
ear-f-83	34.85	34.97	35.87	**34.2**	36.8
hec-s-92	10.78	10.99	11.5	**10.4**	11.3
kfu-s-93	**13.84**	13.89	14.37	14.3	15
lse-f-91	**10.48**	10.6	10.89	11.3	12.1
rye-s-93	9.06	9.08	9.3	**8.8**	10.2
sta-f-83	157.13	157.22	158.81	**157**	157.2
tre-s-92	8.27	**8.26**	8.38	8.6	8.8
uta-s-92	3.29	**3.24**	3.35	3.5	3.8
ute-s-92	26.03	26.23	27.24	**25.3**	27.7
yor-f-83	37.38	38.38	39.33	36.4	39.6

6 Conclusion

The aim of this study was to test the effectiveness of incorporating a new operator, namely, cell depletion into the DA algorithm. DAD produces better results on 9 of the 12 benchmarks. Even the other 3 benchmark results are comparable to that achieved by the other versions of the DA. Enough evidence has been provided to justify the inclusion of the operator in the DA algorithm. Furthermore, by including this operator, the DA now mimics nature in cell biology more closely. An empirical comparison of the runtimes of the DAD and DA show that the runtimes of the DAD are comparative to that of the DA. Future work will involve investigating a more sophisticated and intelligent way of introducing cell depletion into the DA to keep runtimes down but achieve better results as well as examining stimulus driven cell depletion, similar to cell interaction and migration, which only accepts deletions improving the fitness of the organism.

References

[1] Abdullah, S., Turabieh, H., McCollum, B. (2009). A Hybridization of Electromagnetic-Like Mechanism and Great Deluge for Examination Timetabling Problems. *Hybrid Metaheuristics, Lecture Notes in Computer Science*. 5818: 60–72.

[2] Alberts, B., Johnson, A., Lewis, J. (2002) Molecular Biology of the Cell, 4th Edition, New York : Garland Science.

[3] Alzaqebah, M., Abdullah, S. (2011). Artificial Bee Colony Algorithm for Examination Timetabling Problems. *International Journal of Physical Sciences*. 6(17):4264–4272.

[4] Burke, E.K., Bykov, Y. (2006). Solving Exam Timetabling Problems with the Fex-Deluge Algorithm *Proceedings of the International Conference on the Practice and Theory of Automated Timetabling (PATAT 2006)*. 370–372.

[5] Burke, E.K., Bykov, Y. (2008). A Late Acceptance Strategy in Hill-Climbing for Examination Timetabling Problems. *Proceedings of the International Conference on the Practice and Theory of Automated Timetabling (PATAT 2008)*. Online. Available: http://www.asap.cs.nott.ac.uk/external/watt /patat/patat08/Papers/Papers.htm [Cited: 28 October 2013].

[6] Caramia, M., Del Olomo, P., Italiano, G.F. (2008). Novel Local-Search-Based Approaches to University Examination Timetabling. *INFORMS Journal of Computing*. 20(1): 86-99.

[7] Carter, M.W., Laporte, G., Lee, S.Y. (1996). Examination timetabling: Algorithmic Strategies and Applications. *Journal of Operational Research Society*. 47:373–383.

[8] Cote, P., Wong, T., Sabourin, R. (2004). Application of a Hybrid Multi-Objective Evolutionary Algorithm to the Uncapacitated Exam Proximity Problem. *Proceedings of the 5th International Conference on Practice and Theory of Automated Timetabling (PATAT 2004)*, 151–167.

[9] Eley, M. (2007) Ant algorithms for the Exam Timetabling Problem. *Practice and Theory of Automated Timetabling VI, Lecture Notes in Computer Science*, 3867:364-382.

[10] Karaboga, D., Basturk, B. (2007). A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony Algorithm. *Journal of Global Optimization.* 39:459–471.

[11] Pillay, N. (2009) The Revised Developmental Approach for the Uncapacitated Examination Timetabling Problem. *Proceedings of SAICSIT 2009*. 187–192.

[12] Pillay, N., Banzhaf W. (2008). A Developmental Approach to the Uncapacitated Examination Timetabling Problem. *Lecture Notes in Computer Science*, 5199:276–285.

[13] Pillay, N., Banzhaf, W. (2010). An Informed Genetic Algorithm for the Examination Timetabling Problem. Applied Soft Computing, 10:457–467.

[14] Qu, R., Burke, E. K., McCollum, B., Merlot, L.T.G., Lee, S. Y. (2009). A Survey of Search Methodologies and Automated System Development for Examination Timetabling. Journal *of Scheduling.* 12(1):55–89.

Proceedings of the 2013 ORSSA Annual Conference
pp. 112–121
www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf

ORSSA Proceedings
ISBN
XXX-XXXXX-XXXX-X

Terrain visibility-dependent facility location through fast dynamic step-distance viewshed estimation within a raster environment

AM Heyns* & JH van Vuuren†

Abstract

The placement of facilities, such as radar, telecommunication towers, telescopes, surveillance cameras and watchtowers, requires careful planning. To operate optimally, these facilities have to be placed according to very specific terrain-related requirements which vary widely, such as intervisibility, proximity and sunlight exposure. Another important criterion to consider for facility placement is that of the *viewshed* of the facility, which is a representation of the visible area of terrain surface within a specified perimeter around the facility. From this viewshed a *viewshed visibility percentage* (VVP) may be computed, indicating the percentage of terrain surface lying within the analysed viewshed area that is visible to the facility. The focus in this paper falls on the estimation of such VVPs — specifically aiming to reduce the computation time of viewsheds through estimated results instead of exact ones, while simultaneously aiming to minimise loss of accuracy. Firstly, the notion of a viewshed and the computation thereof in a raster data environment is elucidated upon, after which a method for estimating viewsheds is proposed. The estimation method relies on resolution-sensitive analyses performed at angular intervals from the facility location. A case study involving a large area of terrain is then performed and the results and effectiveness of the proposed methods are investigated within the context of a bi-objective optimisation model.

Key words: Terrain modelling, line-of-sight, viewshed analysis, facility location

1 Introduction

The class of facility location problems is well documented in the operations research literature [1, 2, 6]. The prototype example of a facility location problem in the manufacturing sector is the optimal placement of a number of factories with the objective of minimising combined transportation costs and delivery times between these factories. It is, however, possible to adapt traditional facility location problem formulations to include placement criteria involving the physical terrain surrounding the facilities. An example of a single-objective facility location problem involving (inter)visibility and terrain-related facility location criteria is the problem of locating n radar facilities on the earth's surface with the objective of maximising the proportion of some pre-specified area of the terrain surface that is visible to at least m of the radar facilities, with $m < n$. The computational complexity of solving this problem grows substantially as a function of increasing values of the parameters m and/or n, as a function of increasing resolution of terrain representation data and as a function of increasing range of the pre-specified area of the terrain surface over which the problem is solved. When the problem becomes multi-objective, multiple terrain-related objectives and additional constraints magnify the increase in model complexity (very substantially). The reason for this magnification lies in the conflicting nature of the objectives. For example, increasing a

*Corresponding author: Department of Logistics, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: andriesheyns@gmail.com

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

telecommunication tower's terrain surface visibility may result in a decrease in the number of towers visible from that tower which can see the fewest remaining towers because of terrain interference.

In order to arrive at good candidate solutions to facility location problems, such as those mentioned above, powerful single- or multi-objective optimisation metaheuristic procedures (such as simulated annealing or genetic algorithms) are typically required[1]. However, since it may be very costly (in terms of the number of floating point operations performed) to evaluate the desirability of a candidate solution in terms of the model objective functions, extreme care should be taken to employ temporally efficient yet sufficiently accurate terrain-related analyses. The purpose of this paper is to make a recommendation in this respect by considering terrain data resolution effects on viewshed analyses.

The criterion of terrain surface visibility refers to the portion of terrain visible from a specific location, which is limited by terrain interference. Here visibility may refer either to optical visibility or merely to telecommunications or radar detection capability. The determination of *viewshed visibility percentages* (VVPs) depends heavily on factors such as data resolution, terrain roughness and distance scale.

This paper opens in §2 with an explanation of the notion of viewshed analysis and the computations involved, specifically in a raster (gridded) terrain environment. An investigation into previously analysed data that exhibit interesting characteristics in terms of viewshed analysis follows in §3, leading to the proposal of a viewshed estimation method which aims to take advantage of certain terrain analysis characteristics in order to compute (as estimations) VVPs with the aim of finding an acceptable trade-off between maximising computation speed and minimising loss of accuracy, in §4. An estimation of viewsheds is thereafter formulated as a bi-objective optimisation problem in §5. Section 6 is a case study in which the newly proposed viewshed estimation technique is applied. Some ideas with respect to possible future work are presented in the final section.

2 Viewshed analysis in a raster environment

Typically, the determination of viewsheds may be performed with respect to two data structure types employed to represent the earth's terrain surface, *vis.* (1) raster data, or (2) *Triangulated Irregular Networks* (TINs). Raster data, otherwise known as *digital elevation models*, represent the earth's surface as a matrix of elevations of regularly spaced points (also called *gridposts*) above sea-level. The gridposts in raster data are measured uniformly across latitude and longitude and may additionally be manipulated to achieve uniform spacing (which is standard practice [3]). TINs are derived from raster data, identifying important gridposts from the raster data model with respect to terrain relief characteristics, while simultaneously disregarding other, lesser important gridposts. The important gridposts are then joined to each other by straight lines to create planar triangles which represent the terrain surface. The effectiveness of using raster data and TINs have been compared extensively before (see [4, 7], for example). The results of these comparisons indicate that no clear preferred data structure type is evident for viewshed determination. For the sake of simplicity, raster data are therefore used in the determination of viewsheds in this paper.

Figure 1 contains an example of what a viewshed may look like for a central gridpost, called the *observer*, within a 50 kilometre radius around it. The darker shaded areas within the circle are portions of the earth's surface that are invisible to the observer. Figure 2 contains an explanation of the process involved in computing a viewshed within a raster environment. In Figure 2 (a), viewed from above, a perimeter is specified within which the viewshed of the central observer gridpost is required. From the observer gridpost, a number of angular analyses are performed. Moving away from the gridpost along each azimuth in fixed step distances, gridposts are identified which lie nearest the azimuth line after each step, after which these target gridposts are determined to be either visible or invisible to the observer. Three such azimuths and their resulting target gridposts are shown in the figure. For a complete analysis, angles within the range $[0, 360)$ have to be considered. It may, however, be specified that only a specific sector or a combination of sectors are to be considered in the event that only certain important terrain surface areas are of concern. Whichever the specification, the spacing between the azimuths is a crucial factor of the analysis. It is clear in Figure 2 (a), where the azimuth spacings are very large, that an excessive number of gridposts are excluded from the analysis if the spacings are too large. This is an important

[1] See [5] for an example of a single-objective facility location problem featuring multiple observer and combined terrain visibility objectives using such metaheuristic procedures.

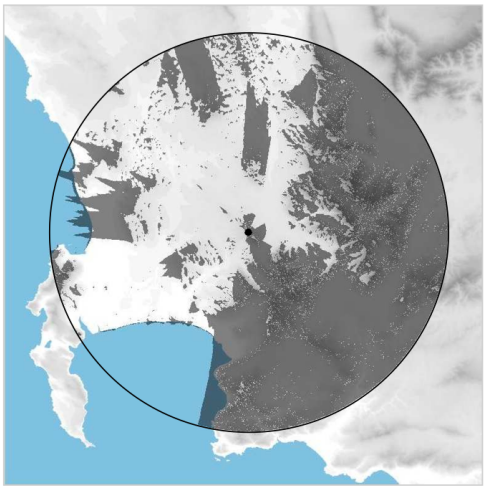


Figure 1: An example of viewshed analysis with respect to an observer gridpost (the black dot) on the earth's surface within a specified radius of 50 kilometres. The darker shaded areas represent earth surface portions that are invisible to the observer as a result of terrain obstruction.

fact to consider in cases where a complete terrain surface analysis is desired.

Figure 2 (b) contains an example of an analysis along a single azimuth. In this example, ten target gridposts are analysed, of which six are determined to be visible to the observer. This results in a gridpost visibility percentage of 60% along the particular azimuth which, in general, is used as indication of overall terrain surface visibility along the azimuth. Suppose that $\mathbb{A}$ denotes the set of angles analysed

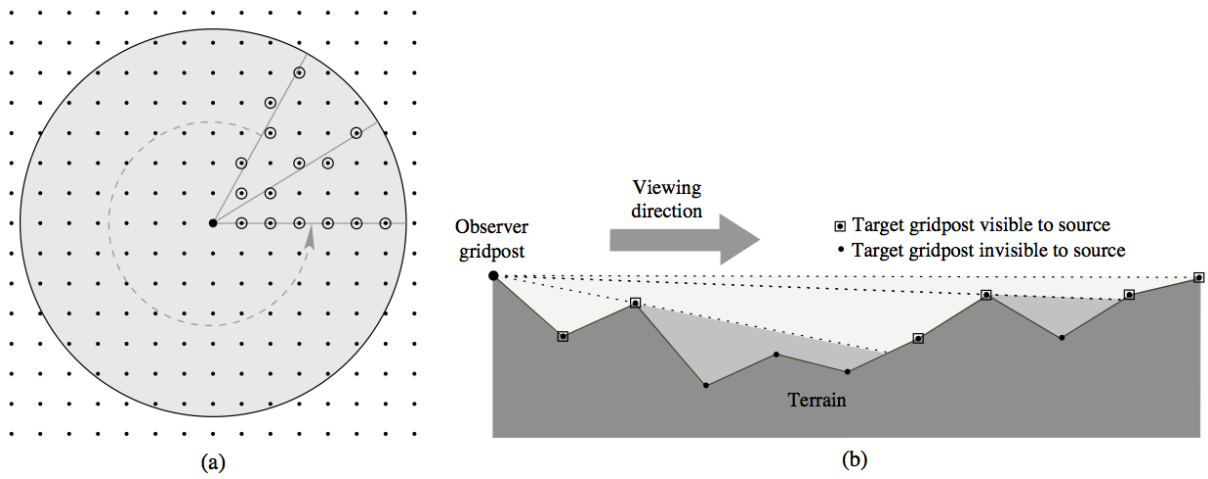


Figure 2: The determination of viewsheds in a raster environment. The top view in (a) illustrates how target gridposts are identified along a set of azimuths, while (b) shows an example of a side profile of an analysis along a single azimuth.

around the observer gridpost and denote the number of angles by $n_a = |\mathbb{A}|$. Also, let $v(\theta, r)$ denote the visibility percentage along azimuth $\theta \in \mathbb{A}$ within a specified range r from the observer. If $n_v(\theta, r)$ denotes the number of visible gridposts along the azimuth and $n_t(\theta, r)$ denotes the total number of target gridposts analysed, then

$$v(\theta, r) = \frac{n_v(\theta, r)}{n_t(\theta, r)} \times 100\ \%. \tag{1}$$

The VVP is defined as the mean visibility percentage determined over all the angles in $\mathbb{A}$. The VVP may therefore be computed as

$$V(r) = \frac{\sum_{\theta \in \mathbb{A}} v(\theta, r)}{n_a}\ \%. \tag{2}$$

3 Analysis of viewshed patterns under different data resolutions

The work presented in this paper is the result of numerous previous viewshed analyses, of which the results consistently exhibited important similarities and characteristics at varying raster data resolutions. One such analysis is discussed to illustrate these similarities and characteristics.

The use of *Digital Terrain Elevation Data* (DTED) is standard practice when modelling and simulating terrain [3]. A single DTED file contains earth surface elevation data for a one degree by one degree area of latitude and longitude. These data files are available at various resolutions and comprise square digital arrays of elevation data. The standard resolutions of DTED files generally have spacings between gridposts of approximately 1 kilometre for *DTED level* 0 (DL0), 100 metres for *DTED level* 1 (DL1) and 30 metres for *DTED level* 2 (DL2), respectively, though varying slightly as a function of latitude North or South from the equator. The categorisation of raster data into different levels of resolution files may be seen as superfluous, since the data in a DL0 file is simply a subset of the data in a DL1 file, and likewise for DL1 with respect to DL2 files. It is therefore not necessary to store different levels of data files — storing the highest resolution data available and extracting gridposts if lower resolution data are required is a more efficient way of storing data, while at the same time facilitating the possibility of extracting data at different resolutions than the standard DTED ones.

The results of a viewshed analysis performed at the three DTED resolutions from the same observer gridpost are shown in Figure 3. Three important sectors are identified and indicated in the figure. Consider sector 1. The visibility of terrain at DL0 is remarkably higher than those at higher resolutions. This is a result of the resolution of DL0 data with respect to points near the observer gridpost. Because the distances between the gridposts are so large, important features of the terrain are skipped, resulting in the terrain interference at low resolution being reduced close to the observer point, resulting in increased visibility of far lying target gridposts. For this specific analysis and observer it is interesting to note that there are high-altitude points very close to the observer, which are skipped in the low resolution analyses, but which increasingly affect visibility as they are included at higher resolutions. At the highest resolution there is no visibility of terrain surface in sector 1, due to gridpost interference adjacent to the observer gridpost. It follows that the use of the highest resolution data available in the vicinity of the observer gridpost is essential to the accuracy of a viewshed analysis further away from the observer.

Considering sectors 2 and 3, it is interesting that the patterns of visible target gridposts further away from the observer are very similar at all three resolution levels, despite some interference near the observer gridpost at higher resolutions. This would result in similar VVPs for the sectors in question beyond the ranges of interference if the observer gridpost were not to experience the interference it does in its close vicinity. This indicates that, at least for viewshed analyses, using the maximum resolution data available becomes less important further away from the observer gridpost.

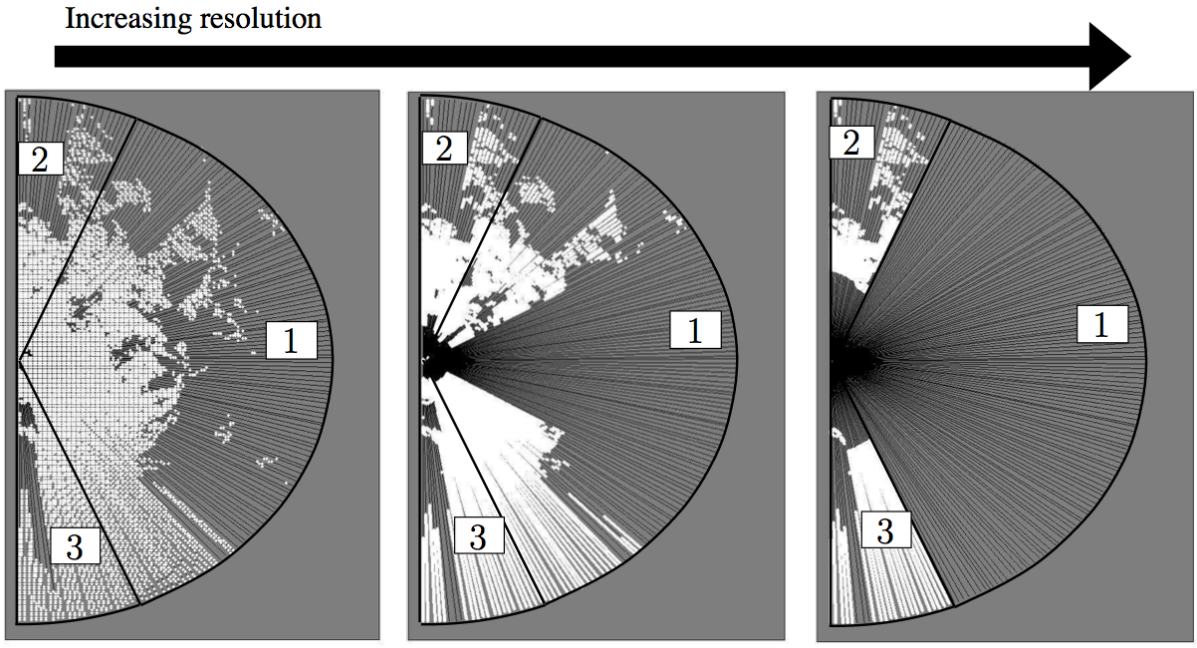
Observations similar to those above were consistently made in other viewshed analyses, resulting in the following conclusions with respect to viewshed analysis accuracy:

- The highest resolution data available should always be used for viewshed analysis in the vicinity of the observer gridpost.
- The importance of high-resolution data required for viewshed analysis decreases as the distance from the observer gridpost increases.

4 Viewshed estimation through dynamic step distance

Following the conclusions made in §3, a dynamic step distance method is proposed in this section for the computation of angular terrain visibility percentages — with the aim of facilitating the estimation of VVPs at reduced computation times.

As discussed in §2 and illustrated in Figure 2, the computation of VVPs requires analyses along a set of azimuths at fixed step distances and determining the nearest target gridposts to be evaluated in terms of their visibility with respect to the observer gridpost along each azimuth in question. The computation time of such an analysis depends heavily on the number of target gridposts analysed which, in turn, depends on the step distance used. If the number of gridposts analysed per azimuth is decreased by



(a) ≈ 1000 metre gridpost spacing (b) ≈ 100 metre gridpost spacing (c) ≈ 30 metre gridpost spacing

Figure 3: Viewsheds determined at different resolutions with respect to the same observer gridpost. The angles of the analysis are between 90 degrees North and 90 degrees South at 1 degree angular intervals . The black lines indicate the path of the visibility analysis along each azimuth and white markers indicate visible target gridposts along each line.

increasing the step distance, the overall VVP computation time would decrease as a result. However, increasing the step distance is comparable to performing analyses at lower resolutions which are expected to return less accurate results than those obtained at higher resolutions — most notably as a result of terrain interference considerations in the vicinity of the observer gridpost, as discussed in §3.

It may be possible to adapt the step distance with the aim of maximising accuracy in the nearby vicinity of the observer gridpost, while minimising the total number of target gridposts by increasing the step distance as the distance from the observer gridposts increases. This incremental step-distance process is similar to the concept of compound interest on financial investments — a slow initial increase in investment amount followed by exponential increase, with the rate of increase depending on the interest rate. If the initial smallest step distance is seen as the initial investment amount, with the rate of change in the step distance seen as the interest rate of the investment, a formula for the factor by which the initial step distance is to be scaled at each step is given by the standard formula

$$f(n) = (1 + i)^n, \tag{3}$$

where $f(n)$ denotes the factor by which the initial step distance is scaled at the n^{th} step (investment growth after n years) and i denotes the rate of increase in the factor (investment interest rate). Additionally, due to the terrain-specific nature of the data, it may be desirable to define an initial window range from the observer gridpost, r_w, within which the step distance remains at the smallest possible value (therefore a factor of 1), before switching over to the incremental adaptation in (3). This will ensure that the analysis in the vicinity of the observer gridpost remains at the highest level resolution possible within the window range, with the aim of improving the accuracy of estimated VVPs as a result of the importance of high resolution near the observer. Furthermore, a maximum step distance factor, f_M, should also be enforced to keep the step distance within reasonable levels, since analyses over large distances or larger values of i may ultimately result in impractically large step distances due to the exponential rate of increase in the factor.

Figure 4 contains an example of the suggested profile of the step-distance scaling factor against the distance from the observer in terms of the parameters introduced above.

The values assigned to r_w, i and f_M have to be considered carefully. If r_w is very small, the step distance

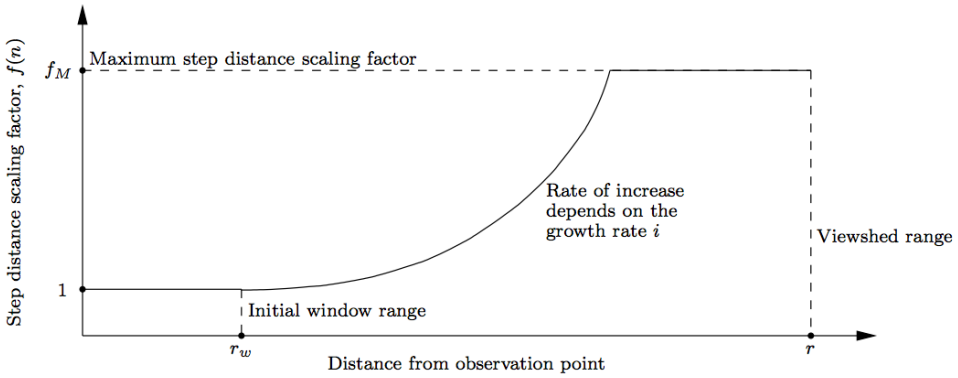


Figure 4: Proposed step distance scaling factor $f(n)$ as a function of distance away from the observer gridpost. The use of such a scaling factor results in dynamic step distances along azimuths used for viewshed analyses, aiming to maximise gains in computation time while minimising the loss of accuracy.

may start increasing too close to the observer, resulting in a large loss of accuracy. Moreover, large values of i may result in the step distance growing too fast, while small values of i may result in the step distance not growing fast enough to reduce the number of target gridposts analysed. The value of f_M is equally important, because values of f_M that are too large or too small may result in very large eventual step distances or minimal step distance increase gains, respectively. The above-mentioned parameters are, furthermore, sensitive to the overall range over which the viewshed analysis is to be performed. It is therefore necessary to determine suitable combinations of these parameters that yield acceptable results (both in terms of accuracy and speed) for the viewshed range in question.

5 Formulation of a bi-objective viewshed estimation model

The trade-off achieved between the accuracy of viewshed results and the speed-up in computation times required to arrive at these results depends on the choice of the initial window range r_w, the scaling factor growth rate i and the maximum scaling factor f_M, described in the previous section. The two criteria — viewshed accuracy and decrease in computation time — may be modelled as objectives in a bi-objective optimisation viewshed estimation model.

The objectives of viewshed estimation are, of course, conflicting in nature — reducing the computation time of a viewshed typically results in a loss of accuracy. A trade-off between extremal values of the objective functions is therefore sought in the form of a subset of candidate solutions to the model (where a candidate solution is a combination of values for the parameters r_w, i and f_M), known as a *Pareto-optimal set of solutions* or a *non-dominated set of solutions*. This set of solutions has the property of being superior to the remaining points in the solution space with respect to *all* the objectives, while at the same time being inferior to the solutions of *one or more*, but not all, of the objectives in the Pareto-optimal set.

Let the VVP of an observer gridpost within a specified range r, determined at the smallest possible uniform step distances (highest resolution), be given by the expression for $V(r)$ in (2). Then, let the estimated VVP of the observer within the same specified range using factor-scaled step distances be denoted by $V'(\mathcal{C}, r)$, where $\mathcal{C}$ is a parameter combination of values for r_w, i and f_M. Then the VVP estimation percentage error, $V_e(\mathcal{C}, r)$, is determined by

$$V_e(\mathcal{C}, r) = \frac{V'(\mathcal{C}, r) - V(r)}{V(r)} \times 100\ \%. \tag{4}$$

Similarly, let $T(r)$ denote the time spent computing the viewshed of an observer gridpost at uniform minimum step distances and let $T'(\mathcal{C}, r)$ denote the time spent computing the estimated viewshed under the parameter combination $\mathcal{C}$. Then the percentage decrease in computation time, $T_d(\mathcal{C}, r)$, is determined as

$$T_d(\mathcal{C}, r) = \frac{T(r) - T'(\mathcal{C}, r)}{T(r)} \times 100\ \%. \tag{5}$$

To determine near optimal values for the parameter combination $\mathcal{C}$ with respect to different values of r, the viewshed estimation model requires that multiple observer gridposts be analysed, after which mean VVP percentage errors and mean percentage decreases in computation time are calculated for different $\mathcal{C}$. These mean VVP percentage errors and mean percentage decreases in computation time may be used to determine which combinations of $\mathcal{C}$ return suitable trade-off results with respect to the objectives. Let $\mathbb{O}$ be a population of observer gridposts analysed to determine mean percentage values of $V_e(\mathcal{C}, r)$ and $T_d(\mathcal{C}, r)$, where the number of observers is $n_o = |\mathbb{O}|$. Also, let e_M denote the maximum mean VVP error percentage threshold values considered acceptable, while t_m denotes the minimum mean percentage decrease in computation time threshold value considered acceptable. The objectives of the bi-objective model is then to find combinations of $\mathcal{C}$ that

$$\text{minimise} \quad \overline{V_e(\mathcal{C}, r)} = \frac{\sum_{o \in \mathbb{O}} V_e(\mathcal{C}, r)}{n_o}, \tag{6}$$

$$\text{maximise} \quad \overline{T_d(\mathcal{C}, r)} = \frac{\sum_{o \in \mathbb{O}} T_d(\mathcal{C}, r)}{n_o}, \tag{7}$$

subject to the constraints

$$\overline{V_e(\mathcal{C}, r)} < e_M, \tag{8}$$

$$\overline{T_d(\mathcal{C}, r)} > t_m. \tag{9}$$

6 Viewshed estimation case study

In §4 a dynamic step distance viewshed estimation method was proposed, with the aim of reducing the viewshed computation time at a minimal loss of accuracy. A case study utilising the dynamic step-distance method and the bi-objective viewshed estimation model of §5 to find near-optimal and practically feasible values for the parameter combination $\mathcal{C}$ is provided in this section.

In order to evaluate the objective functions in (4) and (5), a set of observer gridposts is required. For this purpose a 250 × 150 kilometre section of terrain was selected, shown in Figure 5. In this area, a point selection area was identified within which fifty sample observer gridposts were randomly selected, as indicated in the figure. To simulate real life scenarios with respect to facilities such as telecommunication towers and radars, the observer was placed at an offset of 10 metres above the terrain surface. Additionally, these types of facilities would typically be placed in areas with reasonably high visibility. Therefore, only observers with a minimum VVP of 15% were selected, where the viewshed range r was chosen to be 30 kilometres. The initial window range values of r_w were chosen at uniform intervals of 2 500 metres, starting at 0 and ending at 15 000 metres. The factor change rate i was chosen at uniform intervals of 0.05%, starting at 0.05% and ending at 0.5%. The maximum factor scale was simply chosen as $f_M = 16$. The azimuth angles analysed were at 1 degree intervals in the range $[0, 360)$. The resolution of terrain data was chosen as DL2 (the smallest step distances are therefore approximately 30 metres).

The results of the analysis are presented in Figure 6 for all seventy possible choices of the combination $\mathcal{C}$. The pareto front of non-dominated solutions is shown as filled markers (there are twenty four non-dominated solutions) and the relevant values corresponding to the pareto front are summarised in Table 1 and listed in order of increasing mean decrease in operation time. The standard deviations of the mean error percentages are also provided. Analysing the results of the pareto optimal solutions provides encouraging results, indicating the potential benefit of the dynamic step-distance method in estimating VVPs. A particularly good result is that of the combination of $r_w = 0$ metres and $i = 0.05\%$, with a mean percentage decrease in computation time of 20.1% while suffering only a 10% mean percentage error. The results may be refined further by enforcing the constraints of the form (8) and/or (9).

7 Future work

The work described in this paper is ongoing research. The results presented here are based on the results of a number of viewshed analyses. In order to further validate and improve the dynamic step-distance

Number	Non-dominated solution (r_w (metres), i (%))	$\overline{T_d(\mathcal{C}, r)}$ (%)	$\overline{V_e(\mathcal{C}, r)}$ (%)	$\sigma(V_e)$
1	$(15\,000, 0.05)$	6.9	4.7	1.0
2	$(12\,500, 0.05)$	8.6	5.9	1.4
3	$(10\,000, 0.05)$	10.7	7.3	1.9
4	$(15\,000, 0.10)$	11.0	8.1	1.9
5	$(7\,500, 0.05)$	12.7	8.7	2.4
6	$(5\,000, 0.05)$	15.0	10.0	2.9
7	$(2\,500, 0.05)$	17.8	11.1	3.3
8	$(0, 0.05)$	20.1	11.8	3.5
9	$(7\,500, 0.10)$	20.4	15.4	4.4
10	$(12\,500, 0.20)$	20.7	17.6	4.7
11	$(10\,000, 0.15)$	21.5	17.6	4.9
12	$(5\,000, 0.10)$	23.9	17.8	5.3
13	$(2\,500, 0.10)$	27.8	19.7	6.1
14	$(0, 0.10)$	31.5	20.9	6.5
15	$(2\,500, 0.15)$	34.8	27.0	8.6
16	$(0, 0.15)$	39.3	28.5	9.1
17	$(2\,500, 0.20)$	40.3	33.4	10.9
18	$(0, 0.20)$	45.5	35.0	11.4
19	$(0, 0.25)$	50.1	40.8	13.5
20	$(0, 0.30)$	54.0	46.0	15.5
21	$(0, 0.35)$	57.1	50.8	17.3
22	$(0, 0.40)$	59.8	55.1	19.0
23	$(0, 0.45)$	62.1	59.2	20.5
24	$(0, 0.50)$	64.1	63.0	22.0

Table 1: Solution data for the solutions along the pareto front in Figure 6, numbered the same as in the figure.

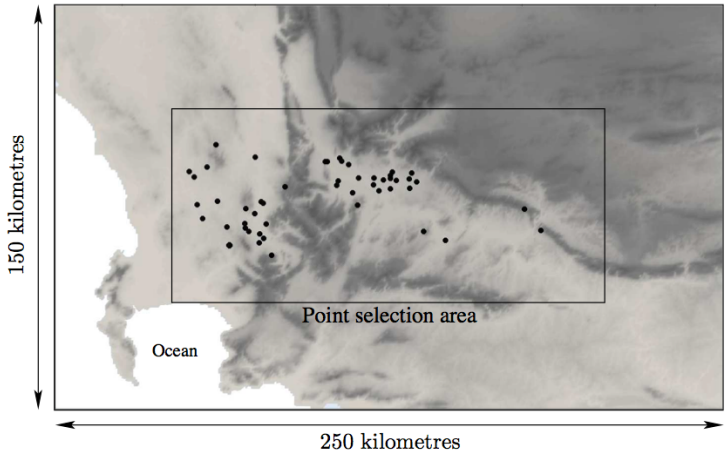


Figure 5: A 250×150 kilometre section of terrain chosen to investigate the potential benefit of the dynamic step-distance method of §4 in the context of the bi-objective viewshed estimation model of §5. The fifty sample observer gridposts used in the study are indicated by dark markers.

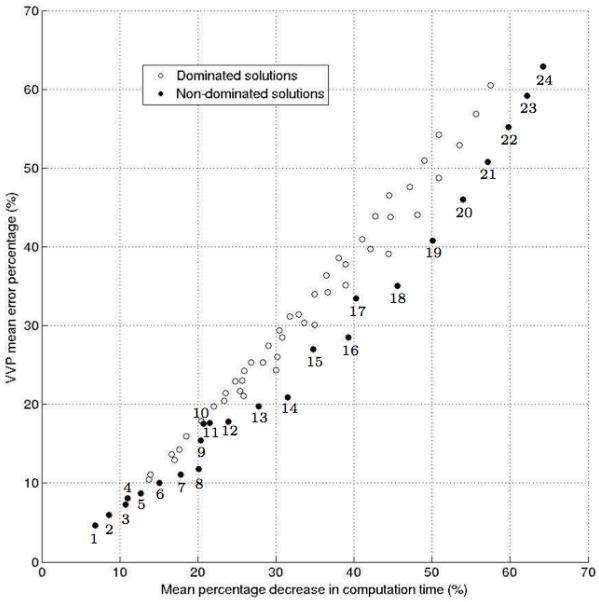


Figure 6: Mean VVP percentage errors and mean percentage decreases in computation time for the different combinations of values for the parameters r_w and i, at a viewshed range of $r = 30$ kilometres and maximum scaling factor of $f_M = 16$.

method, such as determining suitable values for r_w, i and f_M at different viewshed ranges, more extensive analyses should, however, be performed on various terrain areas. The preliminary results obtained here nevertheless indicate that optimisation possibilities exist and that the method proposed in this paper warrants further research. Such further research may include analysing the effect of different values of f_M and r on the results of viewshed analyses using the dynamic step-distance method. The concentration of target gridposts of estimated viewsheds vary at different combinations of $\mathcal{C}$ as a function of distance from the observer. Estimated viewsheds are therefore expected to have VVPs closer to that of the maximum resolution analyses if a distance-weighted gridpost approach is followed.

References

[1] Z Drezner, 2012. *Facility location: A survey of applications and methods*, Springer, New York (NY).

[2] RZ Farahani, M SteadieSeifi & N Asgari, 2010. *Multiple criteria facility location problems: a survey*, Applied Mathematical Modelling, **49(7)**, 1689–1709.

[3] FAS Intelligence Resource Program, 2000. *Digital terrain elevation data [DTED]*, [Online], [Cited June 13th, 2013], Available from `http://www.fas.org/irp/program/core/dted.htm`.

[4] LD Floriani & P Magillo, 2003. *Algorithms for visibility computation on terrains: a survey*, Environment and Planning B: Planning and Design, **30(5)**, 709–728.

[5] Y Kim, S Rana & S Wise, 2004. *Exploring multiple viewshed analysis using terrain features and optimisation techniques*, Computers and Geosciences, **30(9–10)**, 1019–1032.

[6] A Klose & A Drexl, 2005. *Facility location models for distribution system design*, European Journal of Operational Research, **162(1)**, 4–29.

[7] MA Maloy & DJ Dean, 2001. *An accuracy assessment of various GIS-based viewshed delineation techniques*, Photogrammetric Engineering and Remote Sensing, **67(11)**, 1293–1298.

Proceedings of the 2013 ORSSA Annual Conference
pp. 122–132
www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf

ORSSA Proceedings
ISBN
xxx-xxxxx-xxxxx-x

The use of different clustering algorithms and distortion functions in semi supervised segmentation.

D.G. Breed[1] T. de la Rey[2] S. E. Terblanche[3]

Abstract

Methodologies used for model segmentation is typically based on either optimising the target variable separation (like decision trees) or maximising independent variable dissimilarity (like clustering). We explore and compare some methodologies which balance both the use of the target variable as well as the distribution of the independent variables during segmentation. We compare the results of a k-means semi supervised segmentation approach with a more complex density based semi supervised approach on industry data

Key words: Predictive modelling; Semi Supervised Segmentation; K-Means Clustering; Density Clustering;

1 Background to segmentation

The segmentation of data as part of predictive modeling is a well-established practice in the industry today [1, 2, 3] and it is done for many reasons and in many ways. Segmentation in this study is defined as the practice of classifying data observations into distinct groups or subsets,

[1] Corresponding author: North West University, South Africa, PO Box 1819, Northwold Randburg, email: Gerbrand.breed@gmail.com
[2] North West University, South Africa, Box 576, North West University Private Bag X6001 Potchefstroom 2520, email: tanja.delarey@nwu.ac.za
[3] North West University, South Africa, Box 576, North West University Private Bag X6001 Potchefstroom 2520, email: 10794549@nwu.ac.za
This work is based on the research supported in part by the National Research Foundation of South Africa reference number (UID: TP1207243988). The Grant holder acknowledges that opinions, findings and conclusions or recommendations expressed in any publication generated by the NRF supported research are that of the author(s), and that the NRF accepts no liability whatsoever in this regard

with the aim of developing predictive models on each of the groups separately in order to improve the overall predictive power.

Currently two main streams of statistical segmentation exist in the industry, namely *unsupervised segmentation* which maximises the dissimilarity of the segments [4], or *supervised segmentation* which maximises the target variable separation or impurity between segments [5]. An example of the first is k-means clustering, while an example of the second is a decision tree.

Both these streams make intuitive sense depending on the application and the requirements of the models developed and many examples exist where the use of either technique improved model performance. However, both these streams focus on a single aspect, i.e. either target separation or independent variable distribution and combining both aspects might deliver better results in some instances.

We propose a segmentation methodology that is based on current research into *semi supervised clustering* and which aims to balance both the use of the target variable as well as the distribution of the independent variables during segmentation. In order to proceed with details on the proposed semi supervised approach an overview of unsupervised k-means clustering method is provided next.

2 Unsupervised segmentation: K-means clustering

K-means clustering is one of the simplest and most common clustering techniques used in data analysis today [4] It follows a very simple iterative process that continuously cycles through the entire dataset until convergence is achieved.

Let $x_i = \{x_{i1}, x_{i2}, \cdots, x_{im}\}$ denote the i-th observation for an m-variable data set. On completion of the clustering algorithm each observation x_i, with $i = 1,2, \ldots n$, will be assigned to one of the segments $S_1, S_2, \ldots, S_K$ where each S_j denotes an index set containing the observation indices of all the variables assigned to it. That is, if observation x_i is assigned to segment S_j, then $i \in S_j$. Denoting $u_j = \{u_{j1}, u_{j2}, \ldots u_{jm}\}$ as the mean (centroid) of segment S_j, the distance from each observation x_i to the segment mean u_j is given by a distance function $d(x_i, u_j)$. If a Euclidian distance measure is used, then

$$d(S_j, x_i) = \|x_i - u_j\|^2 \tag{1}$$

The objective of the ordinary k-means clustering algorithm is to make segment assignments in order to minimise the inter-segment distances. For notational purposes we introduce $c \in \mathbb{C}$ as an index of an assignment of all the observations to different segments with $\mathbb{C}$ the set of all combinations of possible assignments. **Table 1** illustrates the concept for a data set with 4 observations and 2 segments. There are a total of 7 different combinations of assigning the 4 observations to the 2 segments and the cell entries in the table give the specific observations assigned to each segment:

c	S_1	S_2
1	1	2,3,4
2	2	1,3,4
⋮	⋮	⋮
7	1,4	2,3

$\mathbb{C}$

Table 1

The notation S_{cj} is introduced to reference all the observations for a given assignment $c \in \mathbb{C}$ and for a given segment index j. For example, from the table above $S_{22} = \{1,3,4\}$. The objective function of the ordinary k-means clustering algorithm can now be stated in generic form as the following:

$$\min_{c \in \mathbb{C}} \sum_{j=1}^{K} \sum_{i \in S_{cj}} d(S_{cj}, x_i) \tag{2}$$

3 Semi supervised segmentation within predictive modelling

In comparison to traditional clustering, semi supervised clustering is a relatively new concept [5]. In most applications of semi supervised clustering the aim is to assist or guide the unsupervised clustering algorithm in finding the correct clusters by providing pairwise constraints or class labels on observations where cluster relationships are known. In these applications the class or cluster assignment cannot be fully supervised as the quantity of information known is too little to be representative of the entire data set [5, 6, 7].

The *semi supervised segmentation* methodology we propose is similar to semi supervised clustering in the sense that some a-priori knowledge of the data is imposed on cluster (or segment) formation. It uses similar techniques such as similarity adaptation and search modification which forms the basis of most semi supervised clustering applications [5, 6].

Semi supervised segmentation as proposed here differs from standard semi supervised clustering in that the knowledge imposed has nothing to do with pre-determined cluster affinities of specific observations, but rather the combined and scalable optimisation of an additional outcome. In that sense this application of semi supervised segmentation can also be viewed as a multi target optimisation problem that utilises clustering techniques as its core vehicle of execution. The ultimate goal of semi supervised segmentation, as opposed to semi supervised clustering, is not final object classification, but rather an informed separation of observations into groups on which supervised classification can be performed, i.e. segmentation for predictive modelling.

The most prevalent principle of semi supervised clustering that is utilised in this application is similarity adaptation. That is, a distortion factor is used to adjust the dissimilarity (distance) measure of a clustering algorithm so that the final outcome is not solely influenced by maximum dissimilarity [5].

The formulation given in (2) is generic enough to represent the objective function of any clustering algorithm that aims to minimise inter-segment distances using a predefined distance measure. Furthermore it can easily be extended to include "supervision". Let $\varphi(c)$ denote a distortion function that measures the separation of the target variable y between the segments $S_{c1}, S_{c2}, \dots, S_{ck}$.for a given assignment $c \in \mathbb{C}$. Examples of such a distortion function will be provided below. Let w be a weight of how much the objective function of the clustering algorithm is penalised by the distortion function. The proposed optimisation problem for the

semi supervised segmentation problem taking inter-segment distances into account is the following:

$$\min_{c\in\mathbb{C}}\left[w\,\varphi(c)+(1-w)\sum_{j=1}^{K}\sum_{i\in S_{cj}}d\left(S_{cj},x_i\right)\right] \tag{3}$$

For purposes of describing a heuristic approach for solving the semi-supervised segmentation below, the following partial objective function is introduced:

$$\zeta\left(c,S_{cj},x_i\right)=w\varphi(c)+(1-w)d(S_{cj},x_i) \tag{4}$$

Note that for equation (4) the scale of $\varphi(c)$ and $d(s_{cj},x_i)$ could differ substantially. As part of the solution approach the two terms in (4) are normalised for every assignment $c\in\mathbb{C}$ by subtracting the mean and dividing by the standard deviation.

4 Semi supervised segmentation with k-means clustering

Equation (3) gives a generic formulation of the semi supervised segmentation problem for a user defined distance measure and distortion function. An example of a distortion function is the information value (IV) of a specific population split; see [10]. The IV is a measure of the separation or impurity of the target variable between segments if the target variable is binary. Consider a specific segment assignment $c\in\mathbb{C}$ and let $P_{cj}^{T}=\sum_{i\in S_{cj}}y_i\,/\sum_{l=1}^{K}\sum_{i\in Sc_l}y_i$ be the proportion of events $(y=1)$ of segment S_{cj} relative to the total population. Let $P_{cj}^{F}=\sum_{i\in Sc_j}(1-y_i)\,/\sum_{l=1}^{K}\sum_{i\in Sc_l}(1-y_i)$ be the proportion of non-events $(y=0)$ of segment S_{cj} relative to the total population.

For each observation x_i and each segment S_{cj}, the resulting IV distortion function is defined as:

$$\varphi(\mathrm{c})=\begin{cases}-\sum_{j=1}^{k}\left[\left(P_{cj}^{T}-P_{cj}^{F}\right)\times\ln\left(\frac{P_{cj}^{T}}{P_{cj}^{F}}\right)\right] & \text{if } 0<P_{cj}^{T}<1\\ 0 & \text{otherwise}\end{cases} \tag{5}$$

4.1 A Heuristic approach

The approach for solving the optimisation problem (3) is described by means of an example. Assume a k-means semi supervised segmentation algorithm with IV distortion is in the assignment step with all observations assigned to one of the three clusters $\mathrm{S}_1-\mathrm{S}_3$ except for the last observation x_N. Table 2 gives the combinations of possible assignments that need to be evaluated in order to make the final assignment:

c	S_1	S_2	S_3
c_1	$S_1\cup\{N\}$	S_2	S_3
c_2	S_1	$S_1\cup\{N\}$	S_3
c_3	S_1	S_2	$S_3\cup\{N\}$

Table 2

Table 3 provides details on the values of $\varphi(c)$, $d\left(s_{cj},x_i\right)$ and the partial objective function $\zeta\left(c,S_{cj},x_N\right)$, with weight w at 0.3.

From the results it is evident that the assignment of observation x_N to segment S_1 would yield the minimum partial objective of -0.6403.

c	$d(s_{cj}, x_N)$	$\varphi(c)$	$\zeta(c, S_{cj}, x_N)$
c_1	-0.5361	**-0.8835**	**-0.6403**
c_2	**-0.6177**	-0.2021	-0.4930
c_3	1.1537	1.0856	1.1333

Table 3

The semi supervised segmentation algorithm thus groups observations together that are similar both from an independent variable perspective and a target separation rate perspective. The degree to which the target rate influences the cluster assignment can be adjusted by adjusting the value of w.

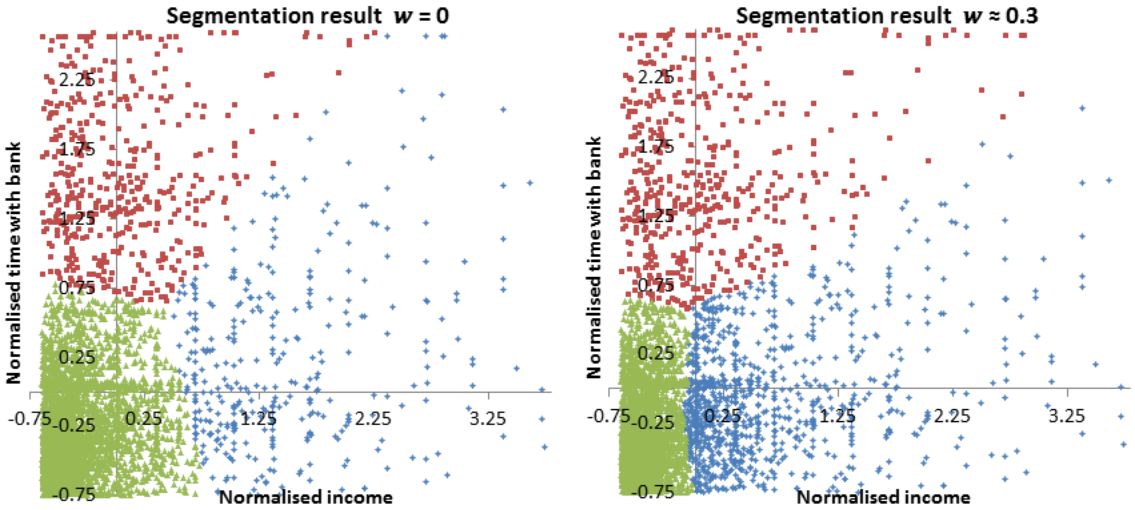


Figure 1

To illustrate segmentation results visually, we will use a dataset which contains information about customers of a bank that was targeted through a direct marketing campaign. The target variable for prediction is binary and indicates whether the client responded to the campaign.

The two characteristics that will be investigated visually is the client's tenure with the bank and the client's income.

Figure 1 visually contrasts the final cluster results of both a value of 0 for w and a high value for w with 3 clusters. The effect is a reduction of the average distance between clusters, but an increase in the value of φ.

Figure 2 shows the impact on the outcomes using different values of the distortion weight w. In theory therefore, w can be adjusted to find an optimal mix between distance-based dissimilarity and target variable separation. However, some analysis and results revealed that choosing a high weight will cause k-means to over fit. This causes the IV value to actually decrease for higher values of the distortion weight when implemented on a validation dataset.

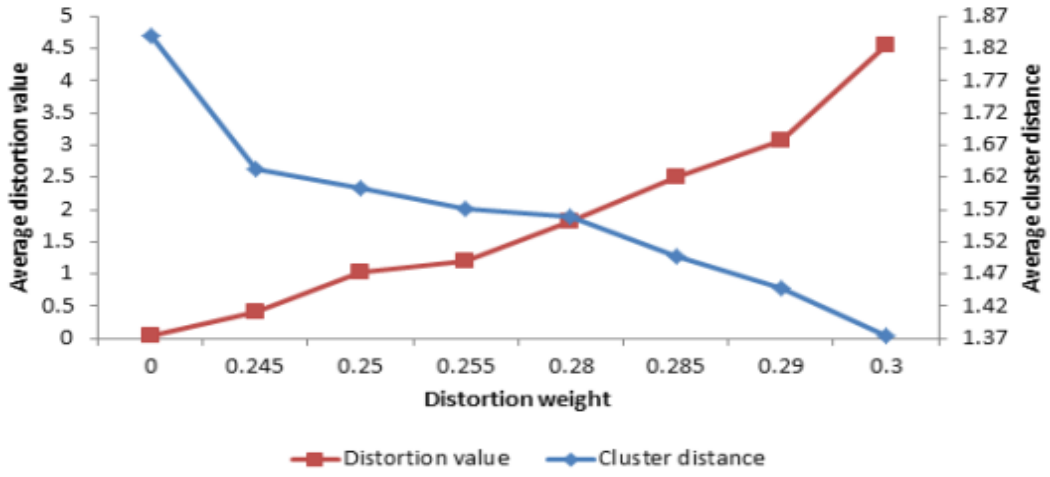


Figure 2

5 Semi supervised segmentation utilising Wong's hybrid clustering method

Although the k-means algorithm has the advantage that it is computationally tractable it unfortunately has the tendency to over fit when applied in the semi supervised context. In an attempt to improve the situation Wong's hybrid clustering method is considered.

Density based clustering techniques have been researched by Hartigan [4], Wong and Lane [8] and many others, and refers to clustering techniques that use probability density estimates to define dissimilarity as well as cluster adjacency. In contrast to the k-means algorithm, these clustering techniques do not start off with a pre-defined number of clusters, but starts off with each observation in its own cluster, which is systematically combined depending on the dissimilarity measure used. Computationally, these techniques are significantly more complex than k-means but possess the ability to form clusters of any form and size [9].

Since many predictive models can now be developed on relatively large datasets (>1000 observations and 20 or more characteristics), the semi supervised segmentation algorithm needs to be viable for such large datasets. This virtually eliminates the more common kernel based density methods like K-nearest neighbourhood [8] and Uniform kernel due to its complexity. Wong [10] proposed a methodology that combines the speed of k-means with the advantages of density based clustering techniques.

Wong's method consists of two stages [10]:

1. A preliminary clustering analysis is done using a k-means algorithm with k significantly larger than the number of final clusters required.
2. The k clusters formed in step 1 is analysed and combined based on density clustering dissimilarities until the required number of clusters is formed or only a single cluster remain.

Preliminary cluster S_r and S_t are considered adjacent if the midpoint between the centroids u_r and u_t is closer to either u_r or u_t than to any other preliminary cluster mean based on Euclidean distance. Each preliminary cluster therefore has only one potential cluster that it can be combined with. The cluster pair which is combined at each step is the pair for which the following density based dissimilarity measure is the minimum:

$$d(s_r, s_t) = \begin{cases} \dfrac{\left(\sum_{i\in S_r} d(S_r, x_i) + \sum_{i\in S_t} d(S_t, x_i) + \frac{1}{4}(|S_r| + |S_t|)\|u_r - u_t\|^2\right)^{\frac{m}{2}}}{(|S_r| + |S_t|)^{1+\frac{m}{2}}} & \text{if } s_r \text{ is adjacent to } s_t \\ \infty & \text{otherwise} \end{cases} \tag{6}$$

The use of (6) requires a reformulation of the semi-supervised segmentation optimisation problem. Keeping with the notation from Section 4 we let $c \in \mathbb{C}$ denote an index of an assignment of all the preliminary segments $S_1, S_2, \dots, S_k$ to the final segments $N_1, N_2, \dots, N_p$ with $p < k$ and with $\mathbb{C}$ the set of all combinations of possible assignments. The aggregation of the initial segments into the final set of segments is done in a binary fashion as illustrated by the tree in Figure 3.

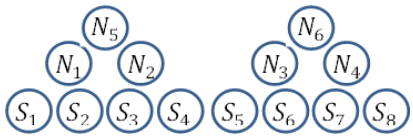


Figure 3

The final segments for the tree, also referred to as the nodes of the tree, are $N_5 = \{N_1, N_2\} = \{S_1, S_2, S_3, S_4\}$ and $N_6 = \{N_3, N_4\} = \{S_5, S_6, S_7, S_8\}$. This is an example of only one combination of assignments. In order to enumerate all possible combinations we use the notation N_{cj} to represent the set of nodes/segments assigned to it for a given combination $c \in \mathbb{C}$. In order to evaluate the density dissimilarity between two segments or nodes that will potentially be joined to form a new node in the tree for a given combination $c \in \mathbb{C}$, we make use of the notation $d(N_{cj})$. For example, to calculate the dissimilarity between nodes N_1 and N_2 by using (6) we calculate $d(N_{c1}) = d(s_1, s_2)$.

The proposed optimisation problem for the semi supervised segmentation problem with density based dissimilarity is the following:

$$\min_{c\in\mathbb{C}} \left[w\,\varphi(c) + (1-w)\sum_{j=1}^{p} d(N_{cj}) \right] \tag{7}$$

For purposes of describing a heuristic approach for solving the density based semi-supervised segmentation approach below, the following partial objective function is introduced:

$$\zeta(N_{cj}, c) = w\varphi(c) + (1-w)d(S_r, S_t) \tag{8}$$

With $N_{cj} = \{S_r, S_t\}$

The semi supervised segmentation algorithm using Wong's method was adjusted to follow three steps:

Step 1 – preliminary cluster analysis

The first step (as with Wong's method) is to do a preliminary cluster analysis by forming a relatively large number of initial clusters. The step, however, differs from Wong's first step in two important ways:

1. Instead of limiting the initial cluster algorithm to only k-means, any clustering methodology which includes kernel based density algorithms (where the size of the dataset is not too large), or even a semi supervised segmentation methodology can be used.
2. Instead of specifying the number of initial clusters, emphasis is placed on the size of the initial clusters, ensuring the initial clusters are large enough to limit the effect of over fitting as observed with the k-means application. The way in which the cluster sizes are controlled depends on the cluster algorithm used, but all algorithms require an input from the user to specify the initial sizes.

Step 2 – target rate clustering

The second step is optional since it depends on the distortion measure used. Some distortion measures, like the IV function requires at least 1 event and 1 non-event in each of the clusters in order to be defined as a non-zero value. Step 2 therefore ensures that this is the case by combining all clusters with either a 0% or 100% target rate with its closest clusters using Wong's standard method.

Step 3 – Semi supervised segmentation

Steps 1 and 2 are preparation steps for the semi supervised segmentation performed in step 3. The data is now organised in such a manner that the partial objective function in (8) can be applied.

The adjacency definition as defined by Wong is not ideal for a semi supervised approach. Therefore cluster adjacency is defined through the use of k-nearest-neighbour kernel based principles. The algorithm requires an input, k, and is broken into two parts. It first identifies the neighbours of each of the initial clusters formed in the first two steps through a k-nearest neighbour analysis on all data, and subsequently assumes that whenever segment S_r is combined with S_t, the new cluster's neighbours are the unique combination of the neighbours of S_r and S_t.

6 Results

IV results were obtained by performing numerous segmentation exercises on the data discussed in Section 4 for both the k-means as well as the density based approaches. Measurements were taken for a range of distortion weights and for different segment sizes. The density based approach continually produced segment combinations with higher IV values as the distortion weight increased where with higher distortion weights the k-means method started to show IV values that either decreased or stabilized (due to over fit). For the k-means method, the optimal IV point differed depending on the number of segments required.

Because the distance measures used in the k-means and density segmentation approaches are different by design, it was not possible to compare the two approaches using their specific objective function distances. However, Calinski and Harabasz [11] proposed a measurement (widely known as the CH measure) that can be universally applied to most forms of cluster analysis to measure cluster pertinence. This CH measure was used to compare the two

methodologies on a like-for-like basis. The CH measure is the ratio of the separation (or "between cluster sum-of-squares") to the cohesion or ("within cluster sum-of-squares"):

$$\frac{\sum_{j=1}^{k} |S_j|(u - u_j)^2}{\sum_{j=1}^{k} \sum_{i \in S_j} (x_i - u_j)} \tag{9}$$

Where u represents the centroid of the entire dataset.

The results showed that the CH measure reduced at very high target weights for k-means, but remained relatively stable overall. What was surprising to see was that the CH measure actually increased for some higher target weights in the density segmentation methodology. This may be due to the fact that segments were formed taking into account more information about the population, securing a more favourable result. Overall, however, the k-means algorithm seemed to form segments with higher CH values for this dataset.

In order to further evaluate the segmentation methodologies proposed, a comparison was done on logistic regression results. The independent variables used contain information on a loan applicant's bureau exposures at application stage and the binary target for these models indicates whether a customer defaulted on a loan granted or not. The two semi supervised segmentation approaches discussed in this paper were applied to the data along with similar unsupervised segmentation techniques and supervised segmentation in the form of a decision tree. In each case the dataset was divided into the segments indicated by the methodology, and logistic regression models were fitted on the segments independently using standard stepwise logistic regression. These results were then verified on a validation sample to ensure no over fit occurred. The combined accuracy of the segmented models was compared to the accuracy of a model fitted on the unsegmented dataset. The results are summarised in Table 4.

It can be seen that the methodology achieving the best results for this dataset was the semi supervised k-means methodology. As expected, the decision tree obtained the best IV value, since it is fully supervised. Of interest is also that the density methodologies failed to propose segments that significantly improved model accuracy even though it was most successful at optimising both the target separation and the segment dissimilarity as measured by the CH measure. This shows that the underlying characteristics of the data used determines the best segmentation methodology, and a single methodology will not necessarily always deliver the best results.

7 Summary and conclusions

The semi supervised segmentation methodologies proposed in this paper are able to identify segments for predictive model development that optimise both the independent variable distribution and target separation. The density based approach is less susceptible to over fit, but this does not guarantee that it will propose segments that will result in more accurate predictive models. The best methodology to use for segmentation will always depend on the characteristics of the underlying data which is used. On the datasets considered in this paper the k-means semi supervised approach outperforms both standard clustering and decision trees as a segmentation methodology based on predictive model accuracy.

Clustering methodology	Nr Segments	Optimal weight (w)	IV Value	CH Measure	Gini	Gini improvement (over LR with no segmentation)	Misclassification rate improvement (over LR with no segmentation)
Unsupervised k-means	10	0	0.119	0.298	34.1%	1.8%	0.52%
Semi supervised k-means	10	0.4	0.140	0.213	34.5%	2.2%	0.66%
Unsupervised density	10	0	0.030	0.339	33.0%	0.7%	0.25%
Semi supervised density	10	0.8	0.237	0.382	33.9%	1.6%	0.50%
Supervised density (decision tree)	10	N/A	0.306	0.138	34.3%	2.0%	0.61%
Logistic regression with no segmentation	N/A	N/A	N/A	N/A	32.3%	N/A	N/A

Table 4: Results on model accuracy

References

[1] R. Anderson, The Credit Scoring Toolkit, New York: Oxford University Press, 2007.

[2] L. C. Thomas, Consumer Credit Models, New York: Oxford University Press, 2009.

[3] N. Siddiqi, Credit Risk Scorecards, New Jersey: John Wiley & Sons, 2006.

[4] T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning, 1 ed., Springer, 2001.

[5] D. J. Hand, Construction and Assessment of Classification Rules, West Sussex: John Wiley & Sons Ltd, 1997.

[6] J. A. Hartigan, Clustering algorithms, 1 ed., New York: John Wiley & Sons, 1975.

[7] N. Grira, M. Crucianu and N. Boujemaa, "Unsupervised and Semi-supervised Clustering: a Brief Survey," Le Chesnay, 2005.

[8] M. B. S. Bilenko, "Integrating constraints and metric learing in semi-supervised clustering," *Proceedings of ICML,* 2004.

[9] Apostolis, "A Review of Machine Learning Techniques for Processing Multimedia Content," Multimedia Understanding through Semantics, Computation and LEarning, Dublin, 2004.

[10] S. Kullback, Information Theory and Statistics, 1 ed., New York: Joh Wiley & Sons, 1958.

[11] M. A. L. T. Wong, "A Kth Nearest Neighbor Clustering Procedure," *Journal of the Royal Statistical Society,* 1983.

[12] M. A. Wong, "A Hybrid Clustering Method for Identifying High-Density Clusters," *Journal of the American Statistical Association,* vol. 77, no. 380, p. 841, 1982.

[13] T. Calinski and J. Harabasz, "A Dendrite Method for Cluster Analysis," *Communications in Statistics – Theory and Methods,* vol. 3, no. 1, pp. 1-27, 1974.

[14] D. MacKay, Information Theory, Inference and Learning Algorithms, 1 ed., Cambridge: Cambridge University Press, 2003.

[15] E. Mays, Credit Scoring for Risk Managers: The Handbook for Lenders, Mason: South-Western Publishing, 2004.

Proceedings of the 2013 ORSSA Annual Conference
pp. 133–140
www.orssa.org.za/wiki/uploads/Conf/2013ORSSAConferenceProceedings.pdf

ORSSA Proceedings
ISBN
XXX-XXXXX-XXXX-X

The use of machine learning techniques to optimize admission requirements at a higher educational institution.

David Hutton* Johan Van Niekerk†

Abstract

Education is commonly seen as an escape from poverty and a critical path to securing a better standard of living. This is especially relevant in the South African context. The need is so great that we have, in one instance, witnessed people trampled to death at the gates of a higher educational institution whilst attempting to register for this opportunity. Tertiary educational institutions are faced with ever increasing applications for a limited amount of available positions. It is therefore imperative to optimally select candidates who will go on to successfully complete their studies and graduate. This study will attempt to use machine learning techniques to determine key indicators to help optimize selection criteria.

Key words: Machine learning, decision tree induction, education

1 Introduction

Tertiary educational institutions are faced with ever increasing applications for a limited amount of available positions. The root cause of this great need is a limited capacity and a demand which outstrips supply. This is not a problem specific to South Africa. It is however exaggerated in the South African context due to the legacy of apartheid which left the country's infrastructure insufficient for the opening of facilities to all people.

It is therefore imperative to optimally select candidates who will go on to successfully complete their studies and graduate. Every student that drops out has effectively taken the place of someone else who may have been successful. The big picture speaks of time, resources and ultimately money that is at stake, be it from government, industry sponsorship or tertiary education. Improving the selection process of candidates is thus a vital cog in delivering a more prosperous and productive future for all South Africans.

At the NMMU School of ICT the current selection mechanism is the Admission Point Score (APS). This scoring system assigns a point score derived from the candidate's grade 12 results. Along with the APS there may be additional requirements that together determine if the student is to be accepted.

Unfortunately this points system is not without error and the NMMU School of ICT still experiences a high drop-out and failure rate resulting in a throughput ratio which is lower than desired.

The remainder of this paper will firstly look at the current admission system in more detail. It will then look at the available data on which the admissions decisions are made. This data will be analysed, comparing admission data with graduation data to form banded ranges of APS scoring. These banded ranges will then be analysed to determine bands most likely to benefit from optimization. Once these

*Department of Information Technology Nelson Mandela Metropolitan University, South Africa, email: david@thehut.co.za
†Department of Information Technology Nelson Mandela Metropolitan University, South Africa, email: johan.vanniekerk@nmmu.ac.za

Grade	APS Contribution
A	7
B	6
C	5
D	4
E	3
F	2
G	1

Table 1: APS points allocation.

bands have been identified, intelligent optimization techniques will be used on this targeted data to determine if more accurate success rates can be achieved.

2 Admission criteria

At the NMMU different faculties set their own specific entrance requirements but all of these are based on an APS scoring system. The scoring system allocates a point for the top six grade 12 subjects taken by the student. The points allocation can be seen in Table 1. A maximum APS of 42 is therefore possible if a student achieves an A in all six of their subjects.

There has been previous work comparing university success to aptitude tests [1] as well as comparing success to academic ability and personal qualities [7]. As far as could be determined none have used machine learning techniques to optimize the selection process.

This study is going to focus on admission requirements for the NMMU School of ICT and more specifically the NDip IT: Soft Dev and the NDip IT: Comm Net. The admission requirements for both of these diplomas is a minimum APS score of 32 points. In addition to this they both require at least a level 3 for home language or first additional language, and at least a level 2 for Mathematics or level 4 for Mathematical Literacy. Applicants who do not meet the requirements for direct admission and have an APS of 26 to 31 are referred for access assessment.

These two diplomas were selected because they are considered representative of the courses on offer at the School of ICT. There is also access to sufficient data which is a requirement to make this study possible.

3 Data scope and delineation

3.1 The scope

The data for this study comes exclusively from the NMMU School of ICT, is specific to the two diplomas mentioned in section 2 and spans several years. It consists of students that successfully registered from 2007 through 2009. The accompanying graduation data is for students who graduated in 2010 through 2013. To be clear, the graduated year falls in the year after the actual study was completed. Thus if a student registered in 2007 and completed the diploma in three years their first year of study would be 2007. Their second year of study would be 2008, third and last year 2009 and their graduation year would be 2010.

3.2 The data

The raw data for this paper came in two forms before it was modified to meet the analytical requirements of this study. This comprised a master list of student registration information along with graduation data. There are two main data representations, firstly the main listing of all students discribed in Table 2.

Secondly a data listing containing all the subject information discribed in Table 3.

Determining when a student dropped out or failed cannot be determined with 100% accuracy as there is no data field indicating this fact. Instead the decision was made to allow a reasonable length of time

Student Key:	A surrogate to anonymize the data.
Academic Year:	The year the student registered for the particular course.
Qual Code:	The qualification code that the student registered to study.
Score:	The calculated APS for the student.
School:	The school attended by the student.
Graduated Year	The year the student graduated.

Table 2: Student registration data.

Student Key:	A surrogate to anonymize the data.
Subject Code:	The code indicating the subject taken.
Mark:	The mark obtained for the given subject.

Table 3: Student subject data.

for students to graduate. For three year diplomas, students were allowed up to four years to graduate. If there is graduation data past the student's fourth year and there is still no graduation date, then that student is deemed to not have completed the course. It is possible for students to complete the course after 5 years, or even longer, however this is not desirable since it represents an inefficient use of resources.

4 Decision tree induction

Decision tree induction is a very small aspect of the greater field of machine learning. Machine learning is made up of three main parts: Induction, Evolutionary Computing and Connectionism [2].

The reason why decision tree induction was chosen over evolutionary computing and connectionism was due to its deterministic nature. It involves forming specific rules which are applied to the problem area in order to determine the solution. This allows for a clear understanding of which attributes play a part in the decision making process so that the process can be reverse engineered and fully defined. The available data attributes will thus be analysed and a classification rule will be developed in the form of a decision tree [3]. Two important concepts need to be defined, namely Entropy and Information gain:

4.1 Entropy

Entropy details the degree of disorder in a dataset. Shannon [4] proposed his theorem for calculating entropy as defined in equation 1 where s denotes a given dataset and n the number of discreet values in the dataset.

$$Entropy(s) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{1}$$

4.2 Information gain

Information Gain is the reduction in entropy by splitting the attribute and can be expressed by equation 2 [8].

$$Gain(S, Subset) = Entropy(S) - \sum \frac{NumberofSubsetValues}{TotalNumberofDatasetValues} Entropy(S_{subset}) \tag{2}$$

When the decision tree is created the information gained from each attribute in the dataset is evaluated. The attribute with the most information gain (strongest attribute) forms the root node. The next node will have the next strongest attribute and in this manner entropy is reduced in the system resulting in an efficient decision tree [3]. The goal is thus to get to a point where all the data is classified to the target output as cleanly as possible.

Group Name	Subjects included
Sciences	Physical Science Biology
Other languages	All other languages.
Business	Business Economics Economics Accountancy
Technical	Technica Electronics
Other	Everything else

Table 4: APS points allocation.

Subject Mark: The mark attainted for the subject.
APS: The calculated APS score.
Successful: A flag indicating if the student graduated.

5 Research

5.1 Initial analysis

During the initial analysis it was found that the data was extremely sparse. No meaningful decision trees could be generated from it. Only a few subjects, like English and Mathematics, were able to influence the decision tree due to the high number of students who took these subjects. Most of the other subjects did not have enough results over the range of students to help construct a meaningful tree.

It was then decided to group the data based on broad subject categories in an attempt to allow the decision tree to perform better. Certain subjects were left untouched because they had sufficient student numbers. These were English, Afrikaans and Mathematics. Computer Science was also selected to remain on its own due to the specific relevance it has to the field of study. Once that was decided upon, the following groups were created as shown in Table 4.

It is important to note that Table 4 is a complete listing of the groups but not of the subjects included in each group. It mentions a few of the subjects to give an idea of what was included in the group. It is worth mentioning that the language split was made purely based on the amount of data available. English and Afrikaans had very high numbers of data points compared to African and other European languages.

The methodology used for grouping was to take the highest grade for the student in the given group. Thus if the student took both Physical Science and Biology, only their highest mark in either of the two subjects would be used.

For subjects that fell into the 'Other' category it was found that the attributes on their own were not significant primarily because the data was so sparse. This was in contrast to English and Afrikaans which were very prominent in the dataset.

Once the data had been collected and analysed the following attributes were chosen for the purpose of generating the decision tree:

The original dataset comprised of 410 rows of data. This data was then split into two datasets. A training dataset consisting of 214 rows and a test dataset consisting of 196 rows which will be discussed in subsequent subsections.

5.2 Decision tree induction in Matlab

For the purposes of this research Matlab was used to generate the decision tree. It has built in classes that allow for both classification and regression trees. Classification trees deal with the predicted value being binary in nature [5] allowing for distinct 'yes' and 'no' type scenarios. Regression trees on the other hand cater for the predicted value being continous [6]. The purpose of this research was to determine success rates. This is a binary value, pass or fail, and as such the classification tree class was used in

Matlab to generate the decision tree.

5.3 Training data

The training dataset consisted of students registering in 2007 and 2008. A value of 5 plus APS was chosen as a reasonable starting point as students with an APS >= 37 were found to have a good success rate based just off of their APS scoring. It should be noted that many of the students in the sample had APS scores below 32 and were thus admitted based on their performance in secondary admissions tests.

5.4 Testing data

The second dataset is the testing dataset which is used for testing and verification of the generated decision tree. This dataset consisted of the same data as in the training dataset i.e. APS < 37, except that it was for students registering in 2009, who were excluded from the training set. This is a very important aspect of the research as without it one could not be sure that the resulting tree is equally effective as a predictor for new data.

5.5 Initial results

Matlab was used to generate a decision tree based on the training data and the resultant decision tree as can be see in Figure 1.

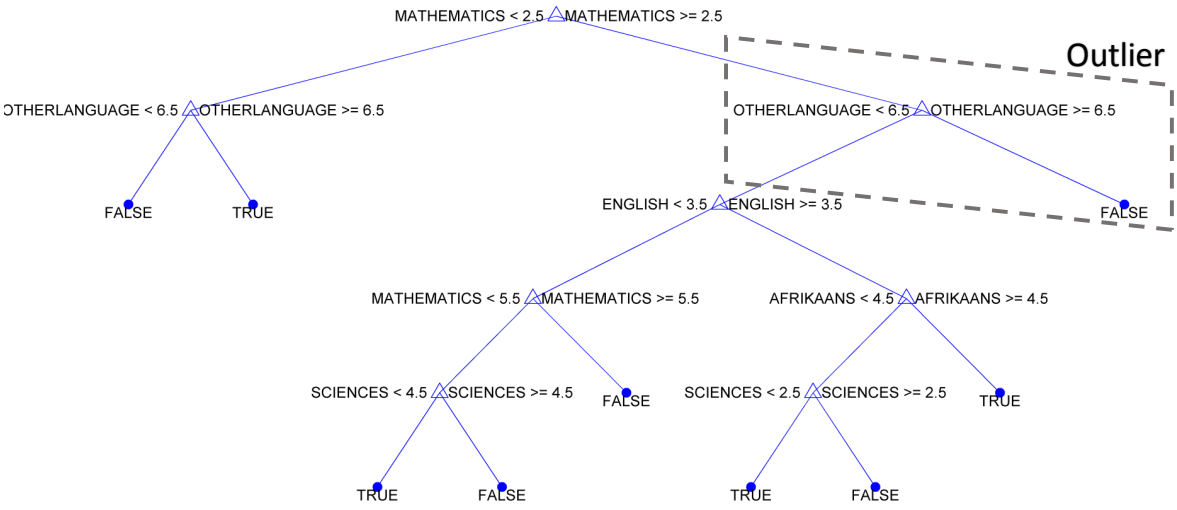


Figure 1: Decision tree

This decision tree details the names of the subjects as well as a data point, which correlates to the mark acheived for the subject, as outlined in Table 1. At the root node a split of the dataset occurs with a mathematics value >= 2.5 with further splits occuring as indicated. The values 'TRUE' and 'FALSE' indicate if the student is likely to be successful. It is clear to see that MATHEMATICS, along with any form of language, play a very important roll in determining future success. The tree still contains some anomolies as can be seen in the area denoted with the caption 'Outlier' in Figure 1. It suggests that a student with MATHEMATICS >= 2.5 and OTHERLANGUAGE >= 6.5 would be unsuccessful. On closer examination of the data, it was found that this case was an outlier in this specific dataset and could therefore be ignored.

5.6 Verification

This decision tree was then encoded into a decision structure which was supplied with the 2009 based test data. The following should be noted:

- The unmodified success rate of the entire test dataset was 33%.
- If the 32 APS point threshold had been implemented as a hard cut-off and no one under that threshold had been considered, the actual success rate would have risen to 58%.
- Applying the decision tree logic to the same band produced a predicted success rate of 59%. This does not point to any clear gain if compared to the APS cutoff of 32.

The next range analysed was for students below 32 APS points. For this range the APS had no bearing on student selection. Instead they were selected via secondary access assessment. The following was noted:

- The success rate for these students below 32 was 32%.
- If the decision tree was used instead of secondary admission testing criteria the selected students below 32 APS would have had a success rate of 58%. This shows a significant difference from the access assessment.

Thus, if one uses the decision tree instead of secondary admissions testing for all students with an APS < 32, and uses the APS score for those above 32 APS points, the overall success rate of the test sample of students in 0 - 36 APS band improves from 58 % to 64%.

6 Discussion

From the results we have obtained we certainly see merit in the APS system and its use. The intention of this study was never to get rid of it completely, and we can confirm that it does serve a useful purpose. The results from section 5.6 have been graphically represented in Figure 2.

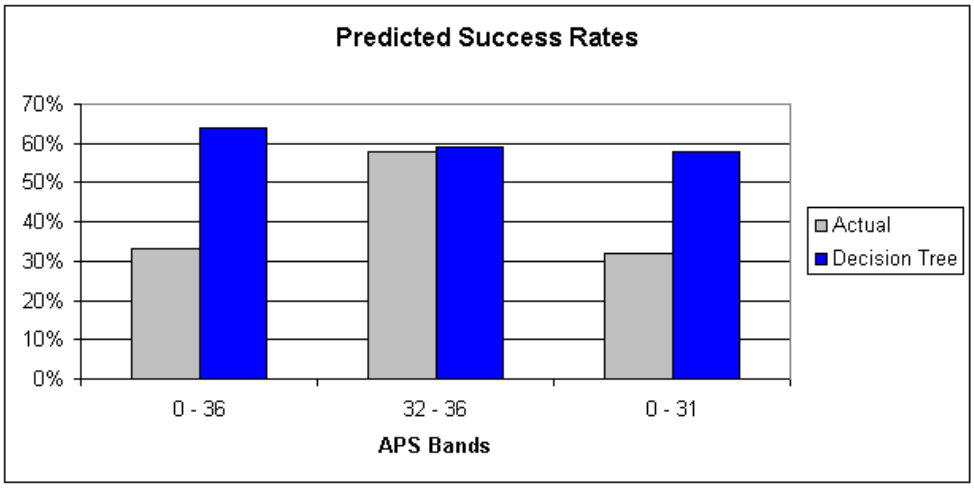


Figure 2: Decision tree

A very clear picture is presented showing how the APS relates to the decision tree, and how the decision tree is able to help identify students outside the bounds of the APS threshold. In the 32 - 36 band the APS and decision tree are almost identical and no significant gain is recorded. As soon as the students below 32 are included the decision tree makes a significant impact.

6.1 Decision tree analysis

Looking at the decision tree in Figure 1 it can be broken down and analysed. Mathematics is clearly a very strong attribute showing up on two levels in the tree including the all important root node. Along

Method	Training data accuracy	Test data accuracy
Decision tree	64%	58%
Rules combined APS	66%	56%

Table 5: Using rules with APS.

with this other languages, English and Afrikaans are all seen as important attributes. It should also be noted that decision trees often result in seemingly non-sensible data, often as the result of outliers in the data. The second listing of other languages would be a case of that, as well as the sciences right at the bottom of the tree. The sciences in particular will be ignored as the results are further examined.

6.1.1 Rule creation

The intent of the study is to provide a more optimal method of predicting success. The decision tree has shown that it can optimize this process. The last step is to interpret the tree and create a set of rules that will combine the APS with the knowledge gained from the decision tree creating a more accurate selection mechanism.

The decision tree was studied and the following rule set was created:

$$\text{APS} = APS_{(current)}$$

$$\text{APS} \mathrel{+}= \text{if(MathematicsScore} > 5.5) \text{ then } \frac{MathematicsScore}{2} \text{ else } 0$$

$$\text{APS} \mathrel{+}= \text{if(OtherLangScore} > 6.5) \text{ then } 3 \text{ else } 0$$

$$\text{APS} \mathrel{+}= \text{if(English} > 3.5 \text{ then } 2 \text{ else } 0$$

$$\text{APS} \mathrel{+}= \text{if(Afrikaans} > 4.5 \text{ then } 1 \text{ else } 0$$

$$\text{Accept} = \text{if(APS} \geq 32) \text{ true else false}$$

When this rule set was applied to both the test and the training data it was within 2% of the decision tree when looking at students who fall below an initial 32 APS. The intent of this rule set is to add weight to specific subjects as identified by the decision tree. Mathematics is the root node so it has the potential to add the most weight by allowing an additional half the Mathematics score to be added if the student is identified by the rule set. The rest of the methodology follows a similar paradigm where additional points are added per level on the decision tree. Other languages get 3 additional points, English 2 additional points and Afrikaans gets 1 additional point. The results gained from applying these rules together with the APS can be seen in Table 5.

7 Conclusion

At this stage of the study it should be noted that the researchers are aware of other factors, one of them being the high school attended, which may also play an important role in success rates. For the purposes of this study other attributes were considered outside the scope of the current research problem and future research might include a broader attribute set.

As can be seen, when adding these additional rules to the APS system, value is added and it is possible to more accurately select successful candidates when compared to secondary admissions testing. The lesson learnt is that the current access assessments are not that accurate and appear to simply add additional overhead to the application process.

The use of machine learning techniques to develop a modification to the APS system that could serve as better predictors for eventual success shows promise. This research was limited to a small data set

and focused on two courses requiring similar skill sets in the same school. Future work would have to explore the use of similar techniques for a larger, more encompassing data set before true generalizations regarding these techniques to improve admissions requirements could be made.

References

[1] Atkinson, R. (2001). Achievement versus aptitude tests in college admissions. *Issues in Science and Technology*, 18(2).

[2] Bhattacharyya, S. and Pendharkar, P. C. (1998). Inductive, evolutionary, and neural computing techniques for discrimination: A comparative study*. *Decision Sciences*, 29(4):871–899.

[3] Quinlan, J. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

[4] Shannon, C. and Weaver, W. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 623–656.

[5] Strobl, C., Malley, J., and Tutz, G. (2009). An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological methods*, 14(4):323.

[6] Wilkinson, L. (2004). Classification and regression trees. *Systat*, 11:35–56.

[7] Willingham, W. W. et al. (1985). *Success in college: The role of personal qualities and academic ability*. ERIC.

[8] Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420. Morgan Kaufmann Publishers, Inc.

Appendix A: Review Criteria

ORSSA 2013 Annual Conference: Peer-review stream
Review guidelines

Compiled by Elias J. Willemse

Email: ejwillemse@gmail.com

Tel: +27 (0) 71 890 2714

Instructions

This document contains general guidelines to assist referees in reviewing manuscripts submitted for possible inclusion in the proceedings of the 2013 Operations Research Society of South Africa (ORSSA) Annual Conference. The document specifies a set of twelve criteria. For each criterion, the paper can be adjudged to be good, fair or poor. A fourth option is also provided (Unsure/not applicable) if the criterion does not apply. Using the criteria as a guideline, referees are required to make a final recommendation as to whether or not the paper should be accepted for publication in the conference proceedings. The review deadline for referees is **Thursday 8 August 2013.** The review process is double blind, so please DO NOT include personal information, such as your name and contact details, in the referee report.

Referees can directly complete this document and return it via email to:

- ejwillemse@gmail.com (email).

Alternatively, referees can compile a separate referee report listing the criteria and their chosen appraisal options. For example:

3. Quality/correct use of mathematics: Poor.
4. Justification of arguments: Fair.

With this option, please make sure that the manuscript number and title is also included. The document should also contain the final recommendation, and notes and comments.

1. Manuscript details

Manuscript number:
Title:

2. Reviewer disclosures

Please answer the two questions below:

	Yes	**No**
Does the article you are being asked to review match your expertise?		
Are there any potential conflicts of interest if you review this article?		

3. Technical quality of paper

Please select the most appropriate appraisal (good/fair/poor/unsure) of the paper for each of the four criteria listed below:

	Good	Fair	Poor	Unsure/not applicable
1. Correct use of language				
2. Clarity of expression/exposition				
3. Quality/correct use of mathematics				
4. Justification of arguments				

4. Contribution of paper to Operations Research

Please select the most appropriate appraisal (good/fair/poor/unsure) of the paper for each of the four criteria listed below:

	Good	Fair	Poor	Unsure/not applicable
5. Knowledge of field				
6. Quality and consistency of referencing				
7. Significance of contribution				
8. Paper suitability for conference proceedings				

5. General

Please select the most appropriate appraisal (good/fair/poor/unsure) of the paper for each of the four criteria listed below:

	Good	Fair	Poor	Unsure/not applicable
9. Clarity and quality of illustrations				
10. Usefulness of paper to OR practitioners				
11. Suitability and length of title and abstract				
12. Suitability of overall paper length				

The submission requirements stipulated that the total paper length (including appendices and references) should not exceed eight pages. The editorial board will, however, allow some leniency, to a maximum of ten pages, in applying this criterion.

6. Final recommendations

Based on the evaluation of the manuscript on the twelve criteria listed above, please select one of the following recommendations:

a) Accept paper as it stands.
b) Provisionally accept paper subject to corrections.
c) Reject paper.
d) Not sure.

Please provide general comments and motivations in the space provided below for making the above recommendation, referring to the evaluation criteria where appropriate. In addition, if you recommended

option (b), please provide a separate list of desired revisions or alterations. If you recommended option (c) please provide a constructive summary of reasons for rejecting the paper.

Notes and comments: