ORiON

Vol. 24, No. 2, 2008 ISSN 0529–191–X

ORiON

Journal of the Operations Research Society of South Africa

Aims & scope

ORiON is the official journal of the Operations Research Society of South Africa (ORSSA) and is published biannually. Papers in the following categories are typically published in ORiON:

- *Development of New Theory*, which may be useful to operations research practitioners, or which may lead to the introduction of new methodologies or techniques.
- *OR Success Stories*, which describe demonstrably successful applications of operations research within the Southern African context (at the developing/developed economy interface) or similar environments elsewhere.
- *OR Case Studies*, which might not be "success stories," but which emphasize novel approaches or describe pitfalls in the application of operations research.
- *OR Methodological Reviews*, which survey new and potentially useful methodological developments, aimed at operations research practitioners especially in Southern Africa.

The above list is by no means exhaustive.

Popular ORiON Publication Topics/Subjects

- Arrival processes, queuing theory and applications
- Assignment, allocation and timetabling problems
- Conflict resolution and multi-criteria decision analysis
- Data mining, forecasting, statistical analysis and applications
- Decision support and decision making
- Demand, logistics and supply chain analysis
- Elections, government and development
- Financial investments, risk analysis and portfolio optimization
- Graph & network theory and applications
- Inventory control theory and management
- Knapsack, packing and cutting problems
- Metaheuristics (*e.g.* genetic algorithms, tabu searches, simulated annealing)
- Mathematical (linear, nonlinear, integer, goal, dynamic) programming
- Natural resource management and conservation ecology
- Philosophy, history, marketing and teaching of operations research
- Production management and project scheduling
- Reliability, repairability and availability
- Theoretical and computer simulation
- Transportation networks, vehicle routing and variations of the TSP

Editorial board

Advisory Board

Editorial

This final issue of ORiON Volume 24 contains five interesting papers, again spanning a substantial portion of the wide operational research spectrum, from practical studies to the establishment of theoretical results and the development of new methodology.

In the first paper, titled *Portfolio selection theory and wildlife management*, John Hearne, Truly Santika and Peter Goodman, consider the potential of portfolio selection theory, first suggested by Markowitz [3] in 1952, to determine the optimal mix of species on a game ranch. In this fascinating application, land, or more accurately the food that it sustains, is the resource available to "invest," while the various game species (with their unique return and risk profiles) are the "investment alternatives." The authors solve the problem of deciding what proportion of the available resource is to be invested in each species. They show that if the objective is to minimise risk for a given return, then the problem is analogous to the portfolio selection problem. They also demonstrate the applicability of their approach using typical data for a hypothetical game ranch. The authors conclude that it is necessary to include a third objective in addition to the usual risk and return objectives, so as to ensure sufficient species to maintain the "character" of a game ranch, in terms of the resulting overall quality of the viewing or hunting experience of visitors to the ranch.

The second paper, by Babul Hasan and John Raffensperger titled *Two pricing methods for solving an integrated commercial fishery planning model*, contains two novel pricing methods for solving an integer programming problem. The authors demonstrate these methods by solving an integrated commercial fishery planning model (IFPM) previously published in ORiON [2]. The aim in this model was twofold: (i) to schedule fishing trawlers (*i.e.* to determine when and where the trawlers should go fishing, and when the trawlers should land their catches), and (ii) to decide how to process the landed fish into products at a processing plant so as to maximise profit. Whereas production planning alone would result in an easy linear programming problem, the introduction of a trawler scheduling aspect into the IFPM results in a hard integer program (in the sense that traditional solution methods involve computation times that are far too long to be practical). The two pricing methods developed in this paper are a decomposition-based O'Neill pricing method and a reduced cost-based pricing method. The authors demonstrate the working of these approximate solution methods by means of numerical examples for different planning horizons, considering differently sized problem instances, and concluding that these methods are indeed viable in terms of their execution times when considering a realistic instance of the IFPM.

In the third paper, titled *The identification of possible future provincial boundaries for South Africa based on an intramax analysis of journey-to-work data*, Hannelie Nel, Stephan Krygsman and Tom de Jong use the intramax method and a combination of national census data on place of residence and place of work to identify functional regions in South Africa, based on journey-to-work flows. They describe how these functional regions may be used to demarcate sensible provincial boundaries or provide solutions to disputed areas. The authors briefly review the process that was followed in 1993 to demarcate the current provincial boundaries and go on to propose new boundaries for a four or five province regime, based on the intramax analysis of the journey-to-work flow data mentioned above. They also put forward practical solutions to a number of split-municipality and disputed region problems that have surfaced in the media over the past few years. Their results compare favourably with those from principal component and cluster analyses [1] previously used to demarcate the South African space economy into a hierarchy of development regions.

The topic of the fourth paper, titled *A survey and comparison of heuristics for the 2D oriented on-line strip packing problem* by Nthabiseng Ntene and Jan van Vuuren, is the two-dimensional oriented on-line strip packing problem in which it is required that items be packed, one at a time without rotation or overlap, into a strip of fixed width and infinite height so as to minimise the total height of the packing. The authors review ten heuristics from the literature for the special case where the items are rectangles, propose six modifications to some of these heuristics, and present two entirely new shelf algorithms for this class of strip packing problems. They then go on to compare the performances and efficiencies of all the algorithms in terms of the mean packing height achieved and computation time required when applied to 542 benchmark data sets documented in the literature. They find that two of their proposed algorithmic modifications outperform most of the reviewed algorithms in the literature if the packing data set satisfies certain conditions in terms of the aspect ratios of the rectangles to be packed.

In the final paper, titled *The Steiner ratio for points on a triangular lattice*, Oloff de Wet presents a novel, short proof that the Steiner ratio for points on a triangular lattice in the Euclidean plane is $2/\sqrt{3}$. The Steiner ratio is an efficiency measure of how badly a *minimum* spanning tree performs compared to a Steiner *minimal* tree. This kind of efficiency finds important applications in, for example, the design of integrated circuit boards, communication networks, power networks and pipelines of minimum cost. In the proof, a Steiner tree in two dimensions is "lifted" to become a rectilinear tree in three dimensions, where it is suitably altered. Proof of the result readily follows for the altered rectilinear tree, which is then projected back into the plane. This beautiful proof is a classic example of the seeming contradiction that it may be exceedingly difficult to prove a result in a confined or special case setting, but much easier to prove a more general result (by relaxing the setting confinement), which admits the original result as special case.

I trust that the diversity and quality of the five papers in this issue are such that each reader of ORiON will find something suiting his/her particular tastes and interests. Suggestions and comments on publications in ORiON, in the form of letters to the editor, are welcome and may be published in future issues of the journal.

In January this year an international Advisory Board was appointed for ORiON, comprising fourteen operations researchers from Canada, England, France, India, Italy, New Zealand, the Philippines, South Africa, and the United States of America (these individuals are listed in the front cover of the journal). An electronic copy of ORiON will be sent to these board members every six months and it will be the task of the Advisory Board to advise the Editorial Board on the standard of papers accepted (on a post-publication basis) and also to help settle refereeing disputes in their areas of expertise. The Board will also lend international standing to the journal. An electronic copy of Volume 24(1) went out the Advisory Board in June this year, and the board members are thanked for the invaluable feedback received, including excellent assessments with respect to the standards of papers that appeared in the Volume 24(1), as well as a wealth of practical advice and sensible suggestions with respect to style, typesetting and general strategizing.

I would also like to thank the eleven authors who have contributed their interesting work to Volume 24(2) of ORiON. My sincere thanks also go to the ten anonymous referees who have generously given of their time to evaluate the papers in this issue timeously and in a very professional manner; their excellent work has led to substantial improvements in the quality of papers in virtually all cases.

My thanks also go to Adri van der Merwe, editorial assistant, who assumes much of the administrative duties involved in managing the submission and refereeing processes of manuscripts. I would also like to thank Philip Fourie who has assisted me with meticulous proofreading of the papers contained in this issue — his time and expertise are much appreciated. Thank you also to Associate Editor John Hearne for managing the refereeing process of the penultimate paper in this issue so timeously and professionally on my behalf.

Finally, I would like to thank the business manager, Stephan Visagie, and his typesetting assistant, Lieschen Venter, for their high standards and considerable patience during the nontrivial typesetting process of the manuscripts in LaTeX and for overseeing the time–consuming publication process of this issue.

Jan van Vuuren
December 2008

References

[1] Harmse AC, 2007, *Socia-economic development regions in the South African space economy*, The South African Geographical Journal, **89**, pp. 83–88.

[2] Hasan MB & Raffensperger JF, 2006, *A mixed integer linear program for an integrated fishery*, ORiON, **22(1)**, pp. 19–34.

[3] Markowitz HM, 1952, *Portfolio selection*, Journal of Finance, **7**, pp. 77–91.

Volume 24 (2), pp. 103–113
http://www.orssa.org.za

ORiON
ISSN 0529-191-X
©2008

Portfolio selection theory and wildlife management

JW Hearne* T Santika† P Goodman‡

Received: 12 September 2008; Revised: 17 October 2008; Accepted: 22 October 2008

Abstract

With a strong commercial incentive driving the increase in game ranching in Southern Africa the need has come for more advanced management tools. In this paper the potential of Portfolio Selection Theory to determine the optimal mix of species on game ranches is explored. Land, or the food it produces, is a resource available to invest. We consider species as investment choices. Each species has its own return and risk profile. The question arises as to what proportion of the resource available should be invested in each species. We show that if the objective is to minimise risk for a given return, then the problem is analogous to the Portfolio Selection Problem. The method is then implemented for a typical game ranch. We show that besides risk and return objectives, it is necessary to include an additional objective so as to ensure sufficient species to maintain the character of a game ranch. Some other points of difference from the classical Portfolio Selection problem are also highlighted and discussed.

Key words: Portfolio selection, multi-objective optimisation, game ranching, wildlife management.

1 Introduction

The trend towards transforming livestock production systems into game ranching has increased rapidly since the early 1990s. By the year 2000 it was estimated that there were approximately 5000 fenced game ranches and 4000 mixed game and livestock farms in South Africa covering more than 13% of the country's land area (ABSA Economic Research, 2003). In 2008 some 3000 additional livestock farms were in the process of conversion to integrated game and livestock production. Some concern about the economic sustainability of this activity and the lack of understanding of risk due to market and climatic variability has been expressed (Falkema and Van Hoven, 2000). Strategies to improve the economic returns from game ranches were formulated by Hearne *et al.* (1996), but this work did not deal with risk.

Theron and Van den Honert (2003) dealt with issues of risk and return in an agricultural context. They developed an agricultural investment model based on investment portfolio techniques first proposed by Markowitz (1952). Their objective was to optimise the

*Corresponding author: School of Mathematical and Geospatial Sciences, RMIT University, GPO Box 2476V, Melbourne, 3001, Australia, email: john.hearne@rmit.edu.au

†School of Mathematical and Geospatial Sciences, RMIT University, GPO Box 2476V, Melbourne, 3001, Australia.

‡KZN Wildlife, P.O. BOX 13069, Cascades 3202, KwaZulu-Natal, South Africa.

proportion of land allocated to each of a number of agricultural products. The ideas of Theron and Van den Honert are followed in this paper. Their potential application to game ranches is explored by means of an illustrative case study.

2 The Problem

The portfolio selection problem is the bi-objective problem of choosing a portfolio of investments that minimises risk while maximising returns. As an acceptable trade-off between risk and return is usually required, an efficient frontier of Pareto optimal solutions is generated by repeatedly solving a single objective optimisation problem. Such a problem minimises risk for various given values of return.

Most modern Operations Research textbooks, such as Winston (2003) or Ragsdale (2004), include the formulation of a simple portfolio selection problem similar to the following formulation.

Suppose K is the total capital available to invest in n investment opportunities. Let p_i and r_i denote respectively the capital invested in and the expected return from investment opportunity i, and let $\boldsymbol{p} = (p_1, \ldots, p_n)^T$. Furthermore, suppose V is the portfolio variance and $\boldsymbol{C}$ is the covariance matrix of investment returns. Then the objective is to

$$\left.\begin{array}{rll} \text{minimise } V = \boldsymbol{p}^T\boldsymbol{C}\boldsymbol{p}, & & \\ \text{subject to } \displaystyle\sum_{i\in S} r_i p_i \geq R, & & \text{(acceptable revenue returned)}, \\ \displaystyle\sum_{i\in S} p_i = K, & & \text{(all capital invested)}, \\ p_i \geq 0, & & i \in \{1, \ldots, n\}. \end{array}\right\} \tag{1}$$

By repeatedly solving (1) with different specified values of R an efficient frontier of portfolio variances may be obtained.

Before pursuing the principles of (1) in a game ranch context some background information is necessary. The food requirements of large herbivores are often given in terms of animal units. An *animal unit* (au) is usually defined as the amount of food required to sustain a domestic cow of 455 kg. An impala, for example, only requires 0.16 animal units per head. Therefore six impala require $6 \times 0.16 = 0.96$ au of food resources which is still less than the food resources required by one domestic cow. The *carrying capacity* of a given area of land is defined as the number of animal units the land can sustain.

For a game ranch, a problem analogous to (1) is obtained if species represent investment opportunities and the carrying capacity of the land represents the capital available for investment. Let K be the number of animal units available (*i.e.* the carrying capacity) and denote the set of livestock species by S. Furthermore, suppose p_i animal units are allocated to species $i \in S$. Then the analogous problem is to

$$\left.\begin{aligned} &\text{minimise } V = \boldsymbol{p}^T\boldsymbol{C}\boldsymbol{p}, \\ &\text{subject to } \sum_{i\in S} r_i p_i \geq R, && \text{(acceptable revenue returned)}, \\ &\sum_{i\in S} p_i = K, && \text{(utilizing carrying capacity)}, \\ &p_i \geq 0, && i \in S. \end{aligned}\right\} \tag{2}$$

A shortcoming of the above formulation is that the total food resources represented by the carrying capacity K are assumed to be homogeneous. The formulation may be improved by dividing the carrying capacity into three broad food classes: *bulk graze*, *concentrate graze* and *browse*. The actual utilisation of these food resources depends on both the number of animal units of each species and their respective diets. Let $F = \{\text{bulk graze, concentrate, browse}\}$, and suppose the proportion of food resource j in the diet of species i is denoted by α_{ij}. Then the additional constraint

$$\sum_{i\in S} p_i\alpha_{ij} \leq K_j, \quad j \in F \tag{3}$$

is required, where $\sum_{j\in F} K_j = K$ and $K_j \geq 0$ for all $j \in F$.

The expected returns generated in this model are more complex than those for the ordinary capital investment portfolio. Whilst the return on an investment in shares is mainly a function of changes in price over a certain period, wildlife returns comprise changes in both sales price and population numbers. For example, suppose that there are b buffalo on a ranch at time t, and suppose that the average market price of buffalo at this time is s_b. Then the market value of the buffalo population on the ranch at time t is bs_b. With an annual population growth rate of f_b a ranch owner may expect to own $(1 + f_b)\,b$ buffalo in year $t + 1$. Also, with an annual price growth rate of $\overline{\Delta s_b}$, the sales price of *buffalo* is expected to become $\left(1 + \overline{\Delta s_b}\right) s_b$ after one year. The value of the population after one year would therefore be $bs_b(1 + f_b)\left(1 + \overline{\Delta s_b}\right)$. From this value and the value at time t it is easily shown that the expected annual return on capital invested in the buffalo population is $\overline{\Delta s_b} + f_b + \overline{\Delta s_b} f_b$.

For species $i \in S$, the expected return on capital in livestock is therefore given by

$$R_i = \overline{\Delta s_i} + f_i + \overline{\Delta s_i} f_i, \tag{4}$$

where $\overline{\Delta s_i}$ denotes the average change in the sales price for species i over a certain time period. This is calculated as

$$\overline{\Delta s_i} = \frac{1}{T-1}\sum_{t=1}^{T-1}\left(\frac{s_{i,t+1} - s_{it}}{s_{it}}\right), \quad i \in S \tag{5}$$

where s_{it} is the sales price of species i at time t, and T is the duration of the time under consideration.

The arithmetic mean is calculated in (5) above. This is the classical approach followed in most textbooks. However, there is a large body of literature with alternative formulations of the problem, including for example, the geometric approach suggested by Leippold *et al.* (2004). A thorough review of various methods for calculating $\overline{\Delta s_i}$ is given by Steinbach (2001).

3 Implementation

Consider a hypothetical but typical ranch in southern Africa. Suppose that twelve species are suitable for this ranch. Data relating to these species are given in Table 1. Typical carrying capacities available on such a ranch would be 250 au of bulk graze and 200 au for each of concentrate graze and browse. Previous annual sales prices over the last fifteen years for each species are given in Table 2 and these prices are used to estimate the rate of price change and the covariance matrix required. The model was implemented using the built–in solver of Microsoft® Excel [2].

			Proportional Food Preference		
Species	**au/head**	**Growth Rate**	**Bulk Graze**	**Concentrate Graze**	**Browse**
White Rhino	2.45	7%	0.9	0.1	0.0
Blesbok	0.22	15%	1.0	0.0	0.0
Zebra	0.54	15%	0.7	0.3	0.0
Blue Wildebeest	0.49	16%	0.3	0.7	0.0
Reedbuck	0.19	15%	0.3	0.7	0.0
Red Hartebeest	0.37	15%	0.2	0.8	0.0
Nyala	0.26	20%	0.0	0.4	0.6
Eland	1.01	15%	0.4	0.2	0.4
Impala	0.16	25%	0.0	0.7	0.3
Giraffe	1.45	12%	0.0	0.0	1.0
Kudu	0.40	15%	0.0	0.1	0.9
Springbok	0.16	15%	0.25	0.25	0.5

Table 1: *List of species, animal units per head, growth rates, and the proportions of each food type in their preferred diet.*

The efficient frontier for this problem is shown in Figure 1. In the absence of risk considerations, a return of nearly 31.28% can be obtained. This drops to 26.31% when risk is minimised without any consideration for returns. Normally a decision-maker can use such a graph to choose the preferred trade-off between risk and return. There are other considerations, however, for decision-makers in this problem.

For a quality hunting experience the ranch needs to have a good spread of species. In Figure 2 the populations of each species are shown for the two extremes of the efficient frontier. In the case where "Return" is maximised it may be seen that only three species are maintained at non-zero population levels. In the case where "Risk" is minimised with no constraint on the required return only five species have non-zero populations.

In terms of a "quality wildlife experience" both the solutions shown in Figure 2 would probably be considered undesirable. It is reasonable to argue that a third objective is required, namely to maximise the minimum proportion of the carrying capacity allocated

Year	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004
W/Rhino	50 172	43 800	26 450	27 400	32 767	48 063	43 812	82 051	107 500	117 949	211 429	176 785	237 500	138 325	142 081
Blesbok	250	375	240	281	370	289	425	545	650	491	650	604	580	711	771
Zebra	2 595	2 175	1 320	1 881	1 675	1 457	1 441	1 715	2 300	1 670	2 336	3 093	5 260	4 385	4 404
B/Wildebeest	682	700	443	1 285	1 658	1 322	1 449	1 796	1 900	1 564	2 400	2 503	3 316	1 350	1 333
Reedbuck	1 283	1 450	1 800	800	2 365	1 400	1 886	2 075	2 500	2 338	3 611	4 562	4 088	4 656	3 806
R/Hartebeest	1 000	2 363	979	1 084	1 900	1 560	1 665	2 600	2 850	2 250	3 013	3 247	2 946	3 720	3 906
Nyala	1 487	1 958	1 407	1 224	1 920	1 970	2 348	2 664	4 450	2 726	5 914	7 362	7 538	5 648	5 588
Eland	2 653	2 641	2 550	4 058	4 308	3 136	4 502	4 195	6 750	4 487	6 114	3 904	4 800	4 945	6 802
Impala	288	375	234	247	320	286	416	480	487	421	480	634	590	469	486
Giraffe	9 750	9 000	6 880	8 800	6 725	7 909	6 150	9 769	11 750	10 141	12 333	12 100	13 350	10 931	11 619
Kudu	1 400	1 600	827	1 175	1 463	1 074	1 054	1 866	3 200	1 747	1 933	2 960	2 050	1 900	2 591
Springbok	550	1 000	525	524	500	842	387	476	650	378	587	718	600	514	602

Table 2: *Annual sale prices (in South African Rands) over fifteen years. The data were gathered by the third author over a period of 15 years from the annual KwaZulu-Natal wildlife game auction, South Africa. Some of the missing data (which accounts for less than 10% of the whole data set) were estimated.*

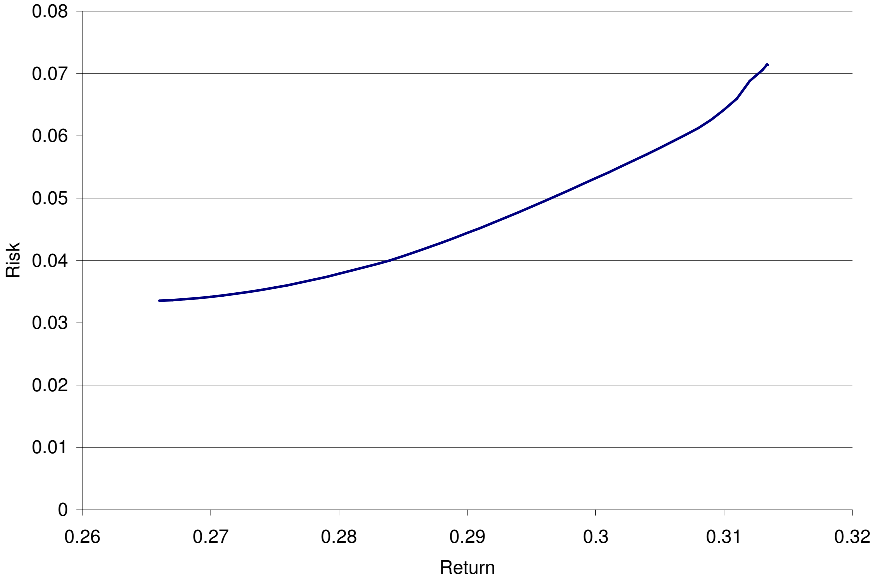


Figure 1: *Efficient frontier of (risk, return) values as solution to (2)–(3).*

to a given species. With three objectives it is best to re-formulate the problem as a multiple objective optimisation problem. "Best" solutions or goals have already been determined for "Returns" and "Risk". Let Q denote the smallest proportion of the carrying capacity allocated to a single species. The following maximin problem determines a goal for Q:

$$\begin{aligned} &\text{Maximise} && Q \\ &\text{subject to} && \sum_{i \in S} p_i = K, && \text{(utilizing carrying capacity)}, \\ &&& p_i \geq Q, && i \in S, \\ &&& Q \geq 0. \end{aligned}$$

The solution to the above problem gives Q as 5.39% of the carrying capacity. This means that each species is allocated *at least* this proportion of the carrying capacity. In terms of individuals this allocates resources sufficient to sustain 35 Eland and greater numbers for other species. Note that due to the constraints relating to the three different types of food resources making up the carrying capacity not all species are allocated equal proportions. So, for example, giraffe are allocated nearly 16% and white rhino just over 30%.

4 Multiple objective optimisation

Having determined goals or best values for the three objectives the multiple objective optimisation problem can now be formulated. Let g_1, g_2 and g_3 be the best values obtained for return, risk, and Q, respectively. Furthermore, let w_1, w_2 and w_3 denote the weights allocated to the objectives of return, risk and Q respectively. Then the objective is to

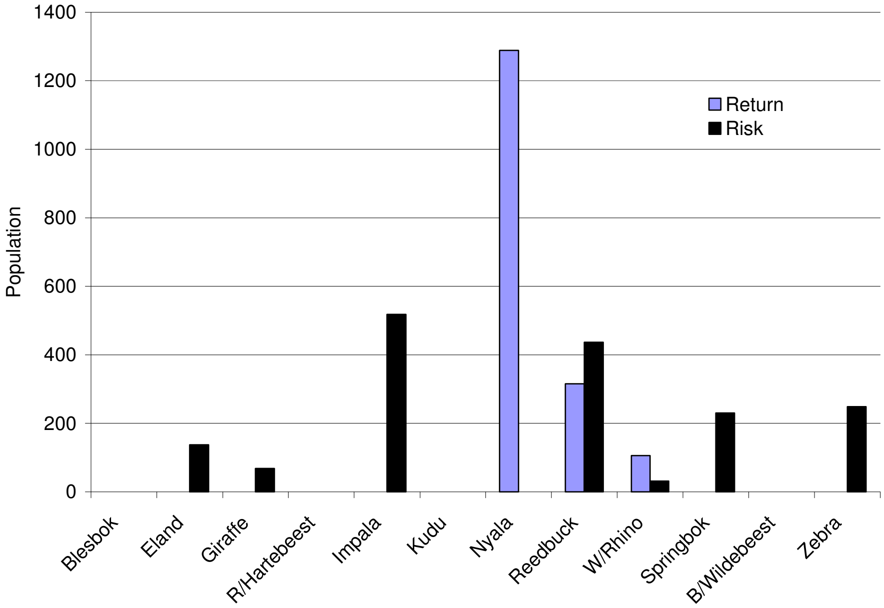


Figure 2: *Populations for the two extreme cases where 'Return' is maximised and where 'Risk' is minimised.*

$$\left.\begin{array}{lll} \text{minimise} & w_1\dfrac{g_1 - \sum\limits_{i\in S} r_i p_i}{g_1} + w_2\dfrac{p^T C p - g_2}{g_2} + (1-w_1-w_2)\dfrac{g_3 - Q}{Q} & \\ \text{subject to} & \sum\limits_{i\in S} p_i\alpha_{ij} \le K_j, \qquad j \in F,\ \text{(enforcing species diversity)}, & \\ & \sum\limits_{i\in S} p_i = K, \qquad \text{(utilising carrying capacity)}, & \\ & p_i \ge Q, \qquad i \in S. & \end{array}\right\} \quad (6)$$

Solving this problem with $w_2 = 0$ and w_1 varying from 0 to 1 the results shown in Figure 3 are obtained. It may be seen that placing more weight on returns reduces the minimum allocation received by a species. Similarly, omitting returns from the objective and varying weights between risk and Q yields the results shown in Figure 4. It is seen that higher risks have to be incurred as Q is increased. It is clear from this analysis that ensuring a "good wildlife experience" comes at the cost of reduced returns and increased risk.

5 Land as capital

We have been dealing with problems that allocate food resources (animal units) rather than capital. Nevertheless, like in the capital investment problem one of the objectives is to maximise the return on capital. Food resources are directly related to the area of land available. In calculating returns on investment, therefore, it would be reasonable that the capital value of the land be taken into account. In §2 we considered the returns that would

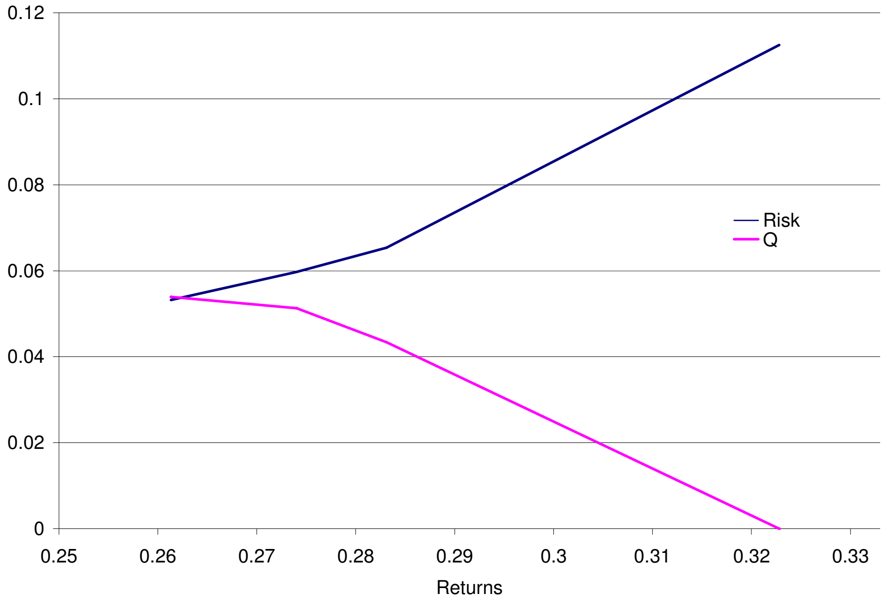


Figure 3: *Solutions to the multiple objective problem (6) with $w_2 = 0$. Here Q is the minimum proportion of carrying capacity (food resources) allocated to any species. The risk associated with each solution is given, but risk was omitted from the objective function.*

be achieved from an initial investment in b buffalo. If L is the value of land utilised by a single buffalo then the return on investment is given by

$$\frac{bs_b\left((1+f_b)\left(1+\overline{\Delta s_b}\right)\right)+bL-(bs_b+bL)}{(bs_b+bL)}.$$

After some simplification the return on investment is given by

$$R_b=\frac{\overline{\Delta s_b}+f_b+\overline{\Delta s_b}f_b}{1+\rho_b},$$

where $\rho_b = \frac{L}{s_b}$ and $L = u_b s_r \pi$. Here u_b denotes the animal unit equivalent for one buffalo (au), s_r denotes the stocking rate (ha.au^{-1}), and π denotes the price per hectare of land (Rand.ha^{-1}). Note that when land value is included, the original return on investment is simply divided by $1 + \rho_i$ for species i.

As an example, using the animal unit equivalent from the second column of Table 1, a stocking rate of 6 hectares per animal unit, and a nominal price of land at R4000 per hectare, the values of ρ can be obtained for each species. For impala and white rhino the calculations yield values of 7.87 and 0.41 respectively. The effect of land price on the returns from these two species may be seen by multiplying the land price by a multiplier. Figure 5 shows the results for land prices from zero through to 1.5 times the nominal land price.

There is an important conclusion to be drawn from Figure 5. Although not true, suppose that impala and white rhino had identical food preferences. In the absence of land costs it would be preferable to stock a ranch with as many impala as possible. As the value

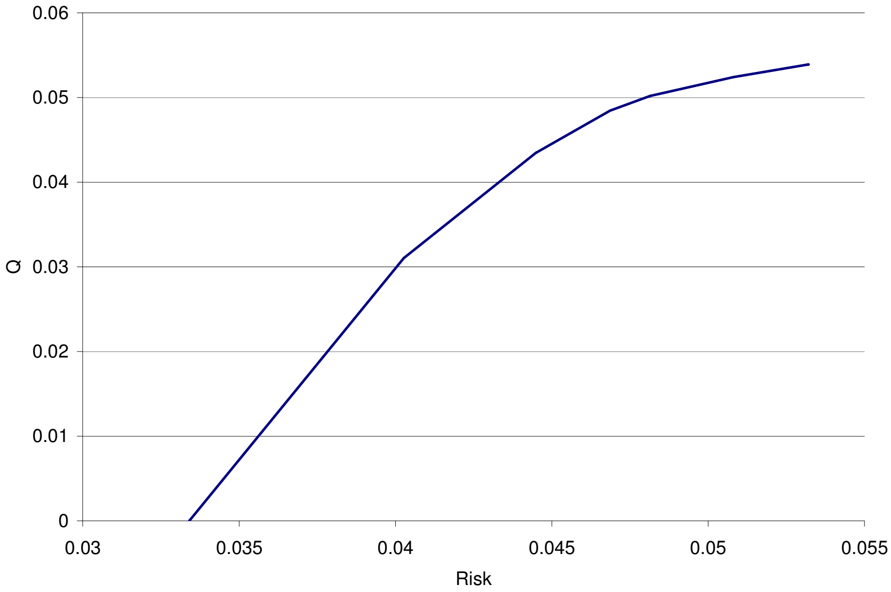


Figure 4: *The relationship between Q and risk in solutions to the multiple objective problem (6) with $w_1 = 0$. Here Q is the minimum proportion of carrying capacity (food resources) allocated to any species.*

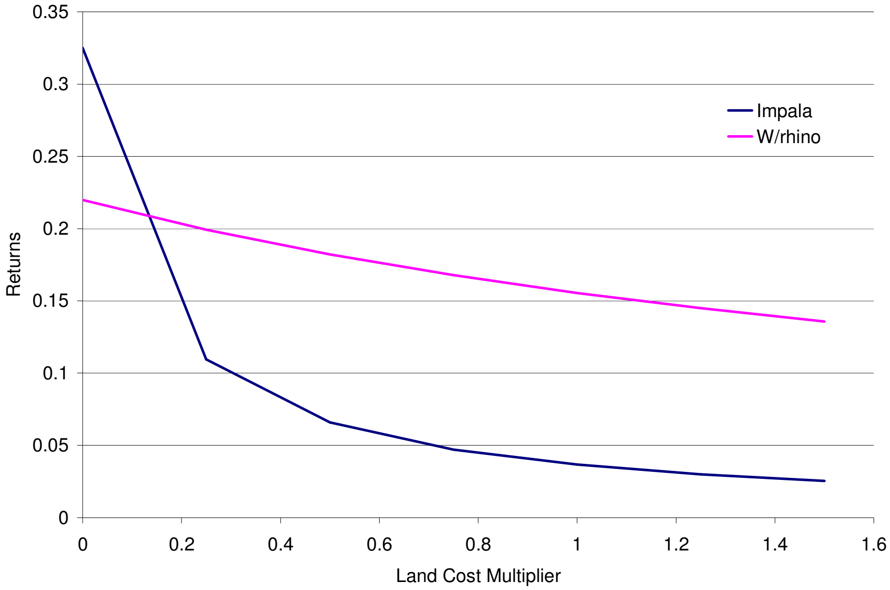


Figure 5: *The effect of land costs on the returns from impala and white rhino.*

of land increases, eventually better returns on investment are obtained from white rhino rather than from impala. It is therefore to be expected that the optimal population of each species will be affected by the value of land.

The effect on return on investment when the cost of land is included in the capital is now further explored. Equal weights were assigned to each of three objectives (returns, risk and

Land cost multiplier	**Blesbok**	**Eland**	**Giraffe**	**R/Hartebeest**	**Impala**	**Kudu**
0	142	146	46	85	194	78
1	139	79	66	84	191	76
2	134	29	81	81	184	73
	Nyala	**Reedbuck**	**W/Rhino**	**Springbok**	**B/Wildebeest**	**Zebra**
0	121	169	63	194	64	58
1	119	273	73	191	63	57
2	115	374	80	184	61	55

Table 3: *The effect of the cost of land on the optimal population numbers of each species. The nominal cost of land is multiplied by 0, 1 and 2 as indicated. For each case the three objectives (returns, risk and Q) in (6) are equally weighted.*

Q). The multiple objective optimisation problem (6) is solved again with three different land costs. This was achieved by multiplying the nominal land costs by 0, 1, and 2. The effect on the optimal populations is shown in Table 3. The two rows commencing with '0' represent the case where land costs are not considered in the calculations. The two rows commencing with '1' use recent or 'nominal' land costs, while the rows commencing with '2' represent the case where land costs are double the nominal value. For each case the three objectives (return, risk and Q) are equally weighted. It can be seen that as land costs increase the optimal balance of species changes: Giraffe, reedbuck and white rhino are allocated a greater proportion of the resources while the population of all other species are decreased. Optimal numbers of Eland, for example, decrease from 146 with no land costs to 79 with nominal land costs.

6 Discussion

The problem of determining population levels for each species on a game ranch so as to maximise returns while minimising risk is essentially analogous to the portfolio selection problem. A difference is that growth in investment value occurs through both natural growth and price change. In our illustration, natural growth was fixed. In practice, however, there will also be some fluctuations in growth rates. It is possible also that changes in price and growth are not independent random variables. There is insufficient data available at present to explore this question further.

A static problem formulation has been used here for illustration purposes. However, these ideas are easily extended to multiperiod problems. In such a case another difference from the standard multiperiod portfolio selection problem emerges. The game ranch problem would not necessarily incur the commission or transaction costs involved in buying and selling shares. Species offering improved returns may simply be allowed to grow to a new level. Of course, this might not always offer an optimal transition path from one 'portfolio' to another.

The purpose of this paper has been to show the connection between portfolio selection problems and the game ranching problem discussed. There have been many advances in Portfolio Selection Theory since the original work by Markowitz (1952). Much of this work can be applied to the game ranch problem in a similar way. The main difficulty is that lack of awareness of this type of approach has meant that the appropriate data has never been collected.

References

[1] Absa Group Economic Research, 2003, *Game ranch profitability in South Africa*, 3rd edition, The SA Financial Sector Forum, Rivonia.

[2] Microsoft Excel, 2008, *Excel homepage*, [Online], [cited 2008, October 29], Available from: `http://office.microsoft.com/en-gb/excel/default.aspx`

[3] Falkema & Van Hoven W, 2000, *Bulls, bears and lions: Game ranch profitability in southern Africa*, SA Financial Sector Forum Publications, Rivonia, p. 69.

[4] Hearne J, Lamberson R & Goodman P, 1996, *Optimising the offtake of large herbivores from a multi-species community*, Ecological Modelling, **92**, pp. 225–233.

[5] Leippold M, Trojani F & Vanini P, 2004, *A geometric approach to multiperiod mean variance optimization of assets and liabilities*, Journal of Economic Dynamics & Control, **28**, pp. 1079–1113.

[6] Markowitz HM, 1952, *Portfolio selection*, Journal of Finance, **7**, pp. 77–91.

[7] Ragsdale CT, 2007, *Spreadsheet modeling and decision analysis*, 5th edition, Thomson South-Western, Mason (OH).

[8] Steinbach MC, 2001, *Markowitz revisited: Mean-variance models in financial portfolio analysis*, SIAM Review, **43**, pp. 31–85.

[9] Theron P & Van den Honert R, 2003, *A mathematical approach to increasing the long-term wealth of agricultural enterprise*, ORiON, **19(1/2)**, pp. 53–74.

[10] Winston WL, 2004, *Operations research: Applications and algorithms*, 4th edition, Duxbury Press, Belmont (CA).

Volume 24 (2), pp. 115–130
http://www.orssa.org.za

ORiON
ISSN 0529-191-X
©2008

Two pricing methods for solving an integrated commercial fishery planning model

MB Hasan[*] JF Raffensperger[†]

Received: 9 April 2007; Revised: 3 August 2007; Accepted: 23 October 2007

Abstract

In this paper, we develop two novel pricing methods for solving an integer program. We demonstrate the methods by solving an integrated commercial fishery planning model (IFPM). In this problem, a fishery manager must schedule fishing trawlers (determine when and where the trawlers should go fishing, and when the trawlers should return the caught fish to the factory). The manager must then decide how to process the fish into products at the factory. The objective is to maximise profit. The problem may be modelled as a single integer program, with both the trawler scheduling and production planning parts integrated. Inventory constraints connect the two parts of the problem. Production planning alone would result in an easy linear program, but due to the trawler scheduling aspect, the IFPM is a hard integer program in the sense that traditional solution methods result in computation times that are far too long to be practical. The two pricing methods developed in this paper are a decomposition–based O'Neill pricing method and a reduced cost–based pricing method. We demonstrate the methods by means of numerical examples for different planning horizons, corresponding to differently sized problems.

Key words: Decomposition, pricing, reduced cost, fishing trawler scheduling.

1 Introduction

In this paper we present recent research on the solution of an integer program for an integrated commercial fishery's activities. Two loosely-connected problems arise in a modern commercial fishery. The first is to schedule trawlers for fishing, including deciding where and when those trawlers should work, and, crucially, when they should return to land the fish. The landed fish generally becomes inventory, which is raw material for a processing plant. The processing plant cleans, processes, and packages the fish for the market. The second problem is to schedule the processing of different types of products.

Based on real data for a commercial fishery in New Zealand, we previously developed a model (Hasan and Raffensperger, 2006) to solve this problem: the *integrated fishery*

[*] Corresponding author: Department of Management, University of Canterbury, Private Bag 4800, Christchurch, 8020, New Zealand, email: b.hasan@mang.canterbury.ac.nz

[†] Department of Management, University of Canterbury, Private Bag 4800, Christchurch, 8020, New Zealand.

planning model (IFPM). The IFPM is designed to co-ordinate trawler scheduling and processing. The model can theoretically be updated and solved periodically to aid in a manager's decision making. Unfortunately, for realistic planning horizons of 20 periods or more, computational times involved in solving the IFPM are quite long, even with methods such as Dantzig-Wolfe decomposition and subgradient optimisation. We have since developed two novel column generation algorithms to solve the IFPM. These algorithms show promise and are based on the decomposition-based pricing algorithm of Mamer and McBride (2000), combined with the integer variable pricing method of O'Neill *et al.* (2005).

1.1 The fishery planning literature

Wide-ranging research has been reported on fisheries. Many papers describe biological models, but only a few consider production planning. Mikalsen and Vassdal (1981) developed a multi-period linear programming (LP) model for one month production planning so as to smooth the seasonal fluctuations of fish supply. Their model is market-driven and incorporates the acquisition of raw material purchased (rather than acquired with their own fishing fleet).

Jensson (1988) developed a product mix LP model to maximize profit of an Icelandic fish processing firm over a five period planning horizon. He addressed both production planning and labour allocation for that processing firm, but did not consider any fleet-specific issues or quota sizes.

Gunn *et al.* (1991) developed a model for calculating the total profit of a Canadian company with respect to integrated fishing and processing. Their model includes a fleet of trawlers, a number of processing plants and market requirements. However, their model ignores the trawler scheduling and labour allocation in the processing firm. Indeed, none of these papers report models that attempt to integrate both trawler scheduling and production.

1.2 The integer programming literature

The literature on integer programming is extensive. We describe only a few papers here that have informed our work.

Martin *et al.* (1985) presented a reduced cost-based branch-and-bound method for solving mixed integer linear programs (MILPs). The authors formulate two candidate problems on the basis of 0-1 integer variables and then optimize both of the candidate problems in order to obtain the MILP solution.

Mamer and McBride (2000) developed a *decomposition-based pricing* (DBP) procedure for linear programs (LPs). Their algorithm works by solving subproblems just as the Dantzig-Wolfe algorithm uses subproblems. However, the DBP master problem exhibits the same form and structure as the original model, but with far fewer variables. Variables that are positive in the subproblem are brought directly into the master problem; all other variables are omitted from the master problem. Our work in this paper builds substantially on the ideas of Mamer and McBride (2000). A DBP approach has also been adopted by De Carvalho (2006) for cutting stock, and by Raffensperger and Schrage (2007) for scheduling training in a tank battalion. We have also previously adopted a DBP approach with respect to an IFPM in Hasan and Raffensperger (2007). In this paper, we describe two improved DBP methods.

The first method we call *decomposition-based O'Neill pricing* (DBONP), because it is based on the work of O'Neill *et al.* (2005). These authors developed a technique for constructing a set of linear prices by solving a MILP and an associated LP, based on a theorem of Gomory and Baumol (1960). They first solve a MILP, set the integer variables to their optimal values, and then remove the integrality constraints to convert the MILP to an LP. They use the dual prices obtained from this LP to form an efficient contract (the dual of the *IFPM*) in the context of an electricity market.

The second method is a *reduced cost-based pricing* (RCBP) method. Unlike Martin *et al.* (1985), we set constraints for both 0-1 integer variables (O'Neill *et al.*, 2005) in the same candidate problem, which is the restricted master problem in the proposed RCBP method. In this method, we do not solve a subproblem at all. Instead, we choose new variables for the restricted master problem based on a reduced cost calculation, and we bring a set of variables into the restricted master problem at each iteration. We show that both of these methods produce better solutions than those reported in our earlier work (Hasan and Raffensperger, 2007).

The remainder of this paper is organized as follows. In §2, we briefly present the IFPM. In §3, we review O'Neill's pricing method and describe the mathematical formulation of the proposed DBONP method. We also present the DBONP algorithm along with numerical examples. In §4, we present the mathematical formulation of the proposed RCBP method, and also present the RCBP algorithm along with numerical examples. The solutions obtained by the DBONP and RCBP methods are compared with that of DBP in §5. Some conclusions follow in §6.

2 The fishery model in matrix notation

In this section we briefly describe our IFPM. The details of the model can be found in Hasan and Raffensperger (2006). We have omitted details of the model in order to focus on the algorithm.

Parameters
Let $V_{t,v}$ be the profit earned by trawler v during period t and let I_t denote the cost per kg of fish landed during period t. Furthermore, suppose $P_{i,j,l}$ is the price per kg of fish type i converted into product j of quality l and let $A^{(0)}_{a,i,t,v}$ denote the amount of raw fish i that will be landed during period t by trawler v from area a. Also, let $A^{(1)}_{i,l,t,v}$ denote the mass transformation for raw fish type i from trawler v of quality l during period t, and let $A^{(2)}_{i,j,l,t}$ denote the mass transformation for raw fish type i into finished product type j of quality l during period t. Finally, let $d^{(1)}_{a,t,v}$ denote the mass balance coefficients on trawler v during period t in area a and let $d^{(2)}_{i,j,l,t}$ denote the mass transformation for raw fish type i into finished product type j of quality l during period t.

Decision variables

Let $w_{p,a,u,t,v}$ be a binary decision variable taking on the value of 1 if a trawler v will go fishing during period u in area a from factory p and lands its catch during period t, or 0 otherwise. Furthermore, let the variable $f_{i,l,t}$ denote the current quantity of fish type i of quality l during period t and let the variable $x_{i,j,l,t}$ denote the amount of fish type i converted into product j of quality l during period t.

The objective in the IFPM is to

$$\text{maximize} \quad \sum_{t,v} V_{t,v} \sum_{p,a,u} (u-t) w_{p,a,u,t,v} - \sum_{t} I_t \sum_{i,l} f_{i,l,t} + \sum_{i,j,l} P_{i,j,l} \sum_{t} x_{i,j,l,t},$$

subject to

$$\sum_{p,u,v} A^{(0)}_{a,i,t,v} w_{p,a,u,t,v} = f_{i,l,t} \qquad \text{for all } i,l,t, \qquad (1)$$

$$\sum_{a,p,u,v} d^{(1)}_{a,t,v} w_{p,a,u,t,v} = b^{(1)}_{t} \qquad \text{for all } t, \qquad (2)$$

$$\sum_{j} d^{(2)}_{i,j,l,t} x_{i,j,l,t} = b^{(2)}_{i,l,t} \qquad \text{for all } i,l,t, \qquad (3)$$

$$\sum_{v} A^{(1)}_{i,l,t,v} f_{i,l,t} + \sum_{j} A^{(2)}_{i,j,l,t} x_{i,j,l,t} = b^{(0)}_{i,l,t} \qquad \text{for all } i,l,t, \qquad (4)$$

$$w_{p,a,u,t,v} \in \{0,1\}, \qquad \text{for all } p,a,u,t,v \qquad (5)$$

$$f_{i,l,t}, x_{i,j,l,t} \geq 0 \qquad \text{for all } i,j,l,t, \qquad (6)$$

where $b^{(0)}_{i,l,t}$ denotes the restriction on the quantity of fish type i of quality l during period t, $b^{(1)}_{t}$ denotes the restriction on the trawler scheduling constraint and $b^{(2)}_{i,j,t}$ denotes the restriction on the quantity of fish type i converted into product j during period t.

Constraint set (1) represents the relationship of the trawler scheduling variables w to landed fish f, as a mass balance in movement of fish from trawlers to the factory, while (2) expresses constraints involving only trawler scheduling, indicating, for example, that a trawler may be in only one place at a time. Constraint set (3) expresses fish processing restrictions, modelling the flow of fish through the factory as raw fish is converted into various products. Constraint set (4) constitutes mass balance constraints, representing the flow of raw landed fish inventory into the fish processing factory. When the integer constraints (5) are relaxed, the model is the usual linear programming relaxation.

The IFPM consists of trawler scheduling and processing, connected by inventory constraints, either (1) or (4). Using Lagrangean relaxation, one can relax either of these side constraints, in which case the model decomposes into an integer program for trawler scheduling, and a linear program for the fish processing. These separate problems are easier to solve, and the sum of their objective values represents an upper bound (since it is a maximization problem) on the optimal objective function value of the IFPM. For example, if we relax (4), we obtain the two subproblems,

$$\left.\begin{array}{lll} \text{maximize} & \displaystyle\sum_{i,j,l} P_{i,j,l} \sum_{t} x_{i,j,l,t} - \sum_{i,l,t} \theta_{i,l,t} \sum_{j} A^{(2)}_{i,j,l,t} x_{i,j,l,t} & \\ \text{subject to} & \displaystyle\sum_{j} d^{(2)}_{i,j,l,t} x_{i,j,l,t} = b^{(2)}_{i,l,t} & \text{for all } i,l,t, \\ & x_{i,j,l,t} \geq 0, & \text{for all } i,j,l,t, \end{array}\right\} P_1(\boldsymbol{\theta}) \qquad (7)$$

and

$$\left.\begin{array}{lll} \text{maximize} & \displaystyle\sum_{t,v} V_{t,v} \sum_{p,a,u} (u-t) w_{p,a,u,t,v} - \sum_{t} I_t \sum_{i,l} f_{i,l,t} & \\ & \displaystyle\qquad - \sum_{i,l,t} \theta_{i,l,t} f_{i,l,t} \sum_{v} A^{(1)}_{i,l,t,v} & \\ \text{subject to} & \displaystyle\sum_{p,u} A^{(0)}_{a,i,t,v} w_{p,a,u,t,v} = f_{i,l,t} & \text{for all } i,l,t, \\ & \displaystyle\sum_{a,p,u,v} d^{(1)}_{a,t,v} w_{p,a,u,t,v} = b^{(1)}_{t} & \text{for all } t, \\ & w_{p,a,u,t,v} \in \{0,1\} & \text{for all } p,a,u,t,v \end{array}\right\} P_2(\boldsymbol{\theta}), \qquad (8)$$

where $\boldsymbol{\theta} = [\theta_{i,l,t}]$ is the matrix of dual prices on (4) assuming that the IFPM is solved as an LP, not as an integer program (IP). It is unfortunate that $P_2(\boldsymbol{\theta})$ cannot be directed by some kind of standard price information on the integer variable $\boldsymbol{w} = [w_{p,a,u,t,v}]$. In fact, the DBONP method actually finds such price information, and uses it.

Following the decomposition-based pricing method for this problem (Hasan and Raffensperger, 2007), the master problem follows from the original problem, assuming its structure and including all its constraints. However, initially only enough variables are included to allow a feasible solution. In the IFPM, the zero matrix is feasible, as the fishery manager can simply choose to do nothing.

At iteration k, the master problem M^k is solved as a linear program, in order to find the necessary dual prices $\boldsymbol{\theta}$. These prices are passed to the subproblems $P_1(\boldsymbol{\theta})$ and $P_2(\boldsymbol{\theta})$, which are then solved. Positive variables from the subproblems are then passed to the master problem, increasing the total number of variables that it contains. The set of variables in the master problem is tracked by the indices of the positive variables found thus far, in an index set I^k. Thus, a variable with its index in I^k has been positive in a subproblem in some previous iteration, and will appear in the master problem. Variables that have always been zero in every subproblem do not have their index in I^k, and thus do not appear in the master problem. In the master problem M^k the objective is to

$$\text{maximize} \quad \sum_{t,v} V_{t,v} \sum_{p,a,u} (u-t) w_{p,a,u,t,v} - \sum_{t} I_t \sum_{i,l} f_{i,l,t} + \sum_{i,j,l} P_{i,j,l} \sum_{t} x_{i,j,l,t},$$

$$
\begin{aligned}
\text{subject to} \quad & \sum_{p,u,v} A^{(0)}_{a,i,t,v} w_{p,a,u,t,v} = f_{i,l,t} && \text{for all } i, l, t, \\
& \sum_{a,p,u,v} d^{(1)}_{a,t,v} w_{p,a,u,t,v} = b^{(1)}_{t} && \text{for all } t, \\
& \sum_{j} d^{(2)}_{i,j,l,t} x_{i,j,l,t} = b^{(2)}_{i,l,t} && \text{for all } i, l, t, \\
\sum_{v} A^{(1)}_{i,l,t,v} f_{i,l,t} + & \sum_{j} A^{(2)}_{i,j,l,t} x_{i,j,l,t} = b^{(0)}_{i,l,t} && \text{for all } i, l, t, \\
& w_{p,a,u,t,v}, f_{i,l,t}, x_{i,j,l,t} \geq 0, && \text{for all } p, a, u, t, v, i, l, t, j \\
& f_{i,l,t}, w_{p,a,u,t,v}, x_{i,j,l,t} \in I^k && \text{for all } p, a, u, t, v, i, l, t, j,
\end{aligned}
$$

where I^k is the index set of positive variables found in the subproblems, and where $f_{i,l,t}$, $w_{p,a,u,t,v}$, $x_{i,j,l,t} = 0$ for $f_{i,l,t}, w_{p,a,u,t,v}, x_{i,j,l,t} \notin I^k$. The index set I^k increases in size with each iteration because each iteration of the subproblems adds new positive variables.

While this decomposition approach is already better than a direct integer programming approach using CPlex, for example, we wished to improve the method further.

3 Decomposition-based O'Neill pricing (DBONP)

In this section, we first discuss the notion of O'Neill pricing in §3.1. In §3.2, we then present the mathematical formulation of the DBONP and present the DBONP algorithm. This is followed by a presentation of numerical examples over different planning horizons §3.3.

3.1 O'Neill's pricing method

O'Neill *et al.* (2005) developed a technique for constructing a set of linear prices from solving an MILP and an associated LP, based on the following theorem of Gomory and Baumol (1960).

Theorem 1 *An MILP with m continuous variables and n integer variables that has a feasible and bounded optimal solution in $(\mathbb{R}^m \times \mathbb{Z}^n)$ can be converted to an LP with at most $(m + n)$ continuous variables and at most n additional linear constraints.* ■

These authors were not interested in a solution as such, nor in the associated computation time, but in finding efficient prices for indivisible objects. To find these prices, they first solved an MILP to optimality. They then added new constraints that fix the integer variables to their optimal values, and removed the integrality constraints to convert the MILP to an LP. Solution of this problem gave dual prices to the new constraints. They showed that the dual variables in the LP have a traditional interpretation as prices. The dual variables explicitly price integral activities, and clear the market in the presence of nonconvexities. They used these dual prices to form an efficient contract in the context of a market for electricity.

3.2 Mathematical formulation for DBONP

To apply the method of O'Neill *et al.* (2005) in the context of DBP, our method

1. finds an optimal solution $\boldsymbol{w} = [w_{p,a,u,t,v}]$ to the restricted master problem as an integer program;
2. fixes the integer variables to their optimal values $\boldsymbol{w}^*$ by means of new constraints of the form $\boldsymbol{w} = \boldsymbol{w}^*$ and solves the restricted master problem as an LP, thus obtaining dual price information $\boldsymbol{\theta_1}$ on the constraints in (9);
3. then uses the resulting dual prices $\boldsymbol{\theta_1}$ to better inform the trawler scheduling subproblem as to which variables should be selected. The trawler subproblem can use this new information through Lagrangean relaxation of the new constraints, that is by solving the following problem, called $P(\boldsymbol{\theta}, \boldsymbol{\theta}_1)$, in which the objective is to

$$\left.\begin{array}{lll}
\text{maximize} & \displaystyle\sum_{t,v} V_{t,v}\sum_{p,a,u}(u-t)w_{p,a,u,t,v} - \sum_{t}\sum_{i,l} f_{i,l,t} & \\
& \displaystyle -\sum_{i,l,t}\theta_{i,l,t}\sum_{v} A^{(1)}_{i,l,t,v} & \\
& \displaystyle -\sum_{i,l,t} {}^{1}\theta_{i,l,t}(u-t)\sum_{p,a,u}(w_{p,a,u,t,v}-w^{*}_{p,a,u,t,v}) & \\
\text{subject to} & \displaystyle\sum_{p,u} A^{(0)}_{a,i,t,v} w_{p,a,u,t,v} = f_{i,l,t} & \text{for all } i,l,t,\\
& \displaystyle\sum_{a,p,u,v} d^{(1)}_{a,t,v} w_{p,a,u,t,v} \le {}^{1}b_{t} & \text{for all } t,\\
& w_{p,a,u,t,v}\in\{0,1\}, & \text{for all } p,a,u,t,v\\
& f_{i,l,t}\ge 0, & \text{for all } i,l,t.
\end{array}\right\} P(\boldsymbol{\theta},\boldsymbol{\theta_1})$$

4. Positive variables from both subproblems are brought into the restricted master problem. Two stopping criteria are enforced, namely when no new positive variables are produced, or when the objective values of the subproblems and master problem are equal. By explicitly pricing the integer variables, and using that price information in the subproblem, we bring better variables into the restricted master problem, and return to step 1.

Note, however, that this approach requires solving the restricted master problem as an integer program at every iteration. This is computationally expensive. We therefore employ ordinary DBP, solving the restricted master problem and subproblems as LPs, until we find an LP optimum. We then switch to the formal DBONP method, and continue iterating. This approach creates two separate loops. The first loop does not utilize the constraints (9), whereas the second loop does.

Loop 1. Relax the inventory balance constraint (4), and then apply the DBP method, to obtain the final restricted master problem as an LP.

Step 0: Initialize. Set iteration number $k \leftarrow 1$ and the initial prices $\boldsymbol{\theta}_1 \leftarrow \boldsymbol{\theta}$.

Step 1: Solve subproblems $P_1(\boldsymbol{\theta})$ and $P_2(\boldsymbol{\theta})$, treating $P_2(\boldsymbol{\theta})$ as an IP. For $\boldsymbol{w}^i > \mathbf{0}$ put i in I^k, where $I^k = \{i : \boldsymbol{w}^i > \mathbf{0}$ in $P_1(\boldsymbol{\theta})$, and $P_2(\boldsymbol{\theta})$ for any iteration $1, 2, \ldots, k\}$.

Step 2: Solve M^k as an LP to obtain dual prices $\boldsymbol{\theta}^k$ and pass them to the subproblems.

Step 3: If $v\left(P_1(\boldsymbol{\theta}) + P_2(\boldsymbol{\theta})\right) = v\left(M^{k+1}\right)$, then go to Loop 2. Else $k \leftarrow k + 1$ and go to step 1. Here $v\left(P_1(\boldsymbol{\theta}) + P_2(\boldsymbol{\theta})\right)$ represents the objective function value of the subproblems and $v\left(M^{k+1}\right)$ represents the objective function value of the restricted master problem.

Loop 2. Solve the current restricted master problem as an IP, and add constraints which fix the integer variables to their optimal values. Solve the master problem as an LP and obtain the dual prices on the inventory balance constraint (4), and the equations associated with the integer variables. We have the dual prices $\boldsymbol{\theta}^k$ as before, but now we also have new dual prices $\boldsymbol{\theta}_1$ from the new constraints.

Step 4: Solve the restricted master problem as an IP.

Step 5: For integer variables, fix $\boldsymbol{w}^i = \boldsymbol{w}^{i*}$.

Step 6 : Solve the master problem as an LP with $\boldsymbol{w}^i$ fixed. Obtain dual prices $\boldsymbol{\theta}^k$ and $\boldsymbol{\theta}_1$, and pass them to the subproblems.

Step 7: Solve the subproblems $P_1(\boldsymbol{\theta})$ and $P_2(\boldsymbol{\theta}, \boldsymbol{\theta}_1)$ with the dual prices obtained from step 6. If no new variables enter into the restricted master problem, then stop. Else go back to step 4.

We present the logic of the DBONP algorithm in the form of a flowchart Figure 1.

3.3 Numerical results

We compare the solutions of the DBONP approach with those obtained from the original IFPM, LP relaxation problem, and DBP algorithm. The results are presented in Table 1. We observe no duality gap for the 5, 10 and 25-period models, thus confirming optimality. However, the 15, 20, and 30-period models exhibit small gaps. For example, a 30-period model exhibits a 0.02% duality gap. The average duality gap is only 0.04% computed over the six different planning horizon models. These gaps may be considered negligible. Notice that for the results described above we started with dual prices of $\boldsymbol{\theta} = \mathbf{0}$. Instead, we also attempted creating the initial dual prices naively. Results are reported in Table 2. Solutions obtained from DBONP are close to the true optima. The average duality gap is only 0.06%, but sometimes worse than in Table 1.

Tables 1 and 2 show that the solutions obtained via the DBONP approach are either equal to or very close to the optimal solutions (15-period, 20-period and 30-period models). To see why a small difference in profit remains, we compared the true optimal solution with that obtained by the DBONP algorithm for a 30-period planning horizon problem. The total number of trawler trips in the DBONP solution coincides with that in the exact solution, but the schedule is slightly different, as shown in Figures 2 and 3. As a result, there is a slight change in the processing and holding costs.

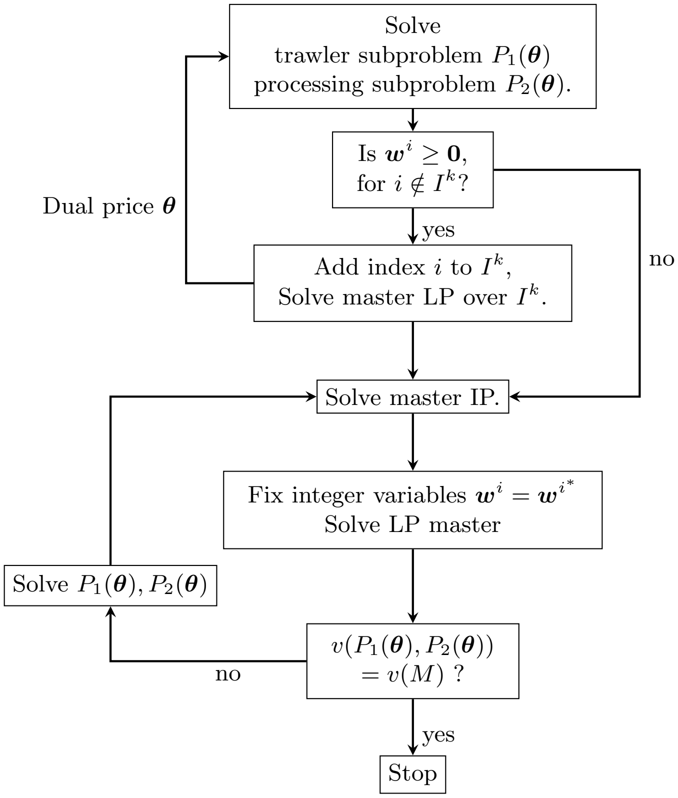

Solve
trawler subproblem $P_1(\boldsymbol{\theta})$
processing subproblem $P_2(\boldsymbol{\theta})$.
Is $\boldsymbol{w}^i \geq \mathbf{0}$,
for $i \notin I^k$?
yes
no
Dual price $\boldsymbol{\theta}$
Add index i to I^k,
Solve master LP over I^k.
Solve master IP.
Fix integer variables $\boldsymbol{w}^i = \boldsymbol{w}^{i^*}$
Solve LP master
Solve $P_1(\boldsymbol{\theta}), P_2(\boldsymbol{\theta})$
$v(P_1(\boldsymbol{\theta}), P_2(\boldsymbol{\theta}))$
$= v(M)$?
no
yes
Stop

Figure 1: *Flowchart of the DBONP procedure.*

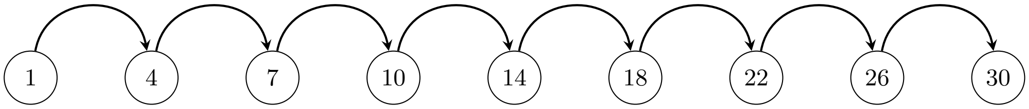


Figure 2: *Trawler 1 schedule in the optimal solution. Here the edges represent periods and vertices represent the required number of periods for a trip.*

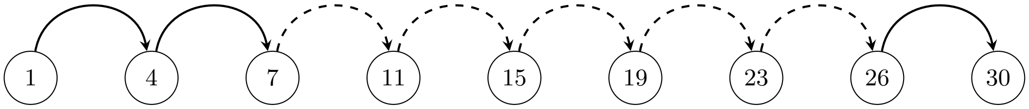


Figure 3: *Trawler 1 schedule in the DBONP solution. Here the normal edges represent the trawler trips which coincide with the schedule obtained by the exact (IP) solution and the dashed edges represent the trips which are slightly different from the schedule obtained by the exact (IP) solution.*

Figures 4, 5, and 6 show the solution times, duality gap, and number of iterations, for different planning horizon models respectively, when solved by the DBP and DBONP algorithms. The DBONP approach requires a larger number of iterations and more computation time, but produces better solutions than the DBP approach.

In this section we employed both DBP and O'Neill pricing to develop the DBONP tech-

Length of planning Horizon	Number of variables	Number of iterations	Solution time (s)	DBP solution	DBONP solution	Duality gap
5	489	29	217	\$522 764	\$522 764	0.00%
10	1 284	27	216	\$1 065 540	\$1 065 775	0.00%
15	2 229	33	345	\$1 579 309	\$1 579 570	0.15%
20	3 324	48	912	\$1 874 097	\$1 878 580	0.08%
25	6 440	45	796	\$2 120 282	\$2 121 887	0.00%
30	6 938	44	3 562	\$2 293 803	\$2 300 230	0.02%

Table 1: *Comparison of the solutions obtained by the DBP and DBONP methods. All computations performed on a Pentium III processor with a clock speed of 665 MHz and 384 MB RAM.*

Length of planning Horizon	Number of variables	Number of iterations	Solution time(s)	DBP solution	DBONP solution	Duality gap
5	1 264	29	208	\$522 764	\$522 764	0.00%
10	2 601	30	266	\$1 065 540	\$1 065 540	0.02%
15	4 087	36	387	\$1 579 309	\$1 580 670	0.08%
20	4 926	50	1045	\$1 874 097	\$1 873 950	0.30%
25	6 259	43	710	\$2 120 282	\$2 121 887	0.00%
30	8 277	50	3129	\$2 293 803	\$2 300 460	0.01%

Table 2: *Comparison of the number of iterations, computation times and solutions obtained by the DBP and DBONP methods. All computations performed on a Pentium III processor with a clock speed of 665 MHz and 384 MB RAM.*

nique. We found that the DBONP algorithm requires slightly longer computation times, but produces better solutions than our earlier DBP procedure. To improve further on the computation times, we also developed a reduced cost–based pricing method.

4 The Reduced cost–based pricing for IFPM

One reason why the DBONP algorithm took a relatively long computation time was due to solution of the trawler scheduling subproblem as an IP. We therefore attempted to eliminate this step. Instead, we use the O'Neill price information to find the reduced cost for each integer variable. Under this approach we are moving away from the DBP philosophy for the trawler scheduling aspect of the problem, but we continue to use DBP for the fish processing subproblem. So the processing subproblem, and the restricted master problem, are the same as with the DBP approach. Instead of employing the trawler scheduling subproblem, we merely calculate the reduced cost of the variables of that subproblem, which is extremely fast.

4.1 The Reduced cost of a variable

The reduced cost of a variable w_j with associated objective function coefficient c_j is the net change in the objective function when generating one unit of w_j, and is defined as

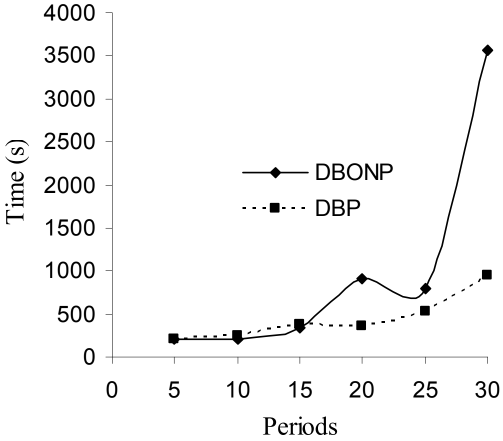


Figure 4: *Solution times required by DBP and DBONP for different planning horizons. All computations performed on a Pentium III processor with a clock speed of 665 MHz and 384 MB RAM.*

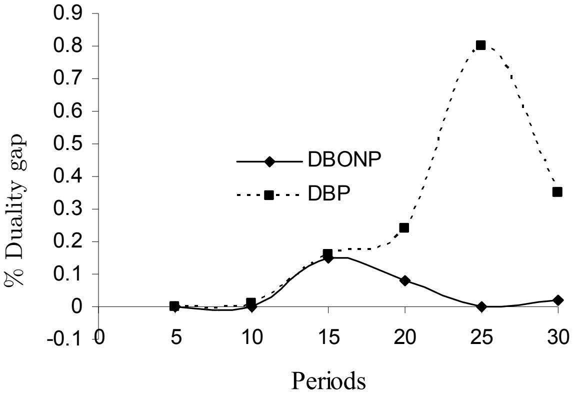


Figure 5: *% Duality gap of DBP and DBONP.*

$\bar{c}_j = c_j - z_j$, where z_j denotes $\boldsymbol{c}_{BV}B^{-1}\boldsymbol{a}_j$. Here $\boldsymbol{c}_{BV}$ are the cost coefficients of the basic variables, B^{-1} is the inverse of the basis matrix, and a_j is the corresponding column of the basic variables. The reduced cost gives the marginal value of a variable on the objective function related to the current basic solution. For a maximization problem, the variable with largest positive reduced cost will be the incoming variable. Following the notation in AMPL (Fourer *et al.*, 1993), we denote the reduced cost of variable $\boldsymbol{w}$ as $\boldsymbol{w}.rc$. Denote $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ as the dual prices on (1) and (2) respectively, with $\boldsymbol{a}_0$ and $\boldsymbol{d}_1$ as the relevant columns of A_0 and D_1 respectively. Then

$$\boldsymbol{w}.rc = \boldsymbol{c}_1 - \boldsymbol{\lambda}_1\boldsymbol{a}_0 - \boldsymbol{\lambda}_2\boldsymbol{d}_1 - \boldsymbol{\theta}' \tag{9}$$

This reduced cost calculation has an explicit term for the integrality constraint. In the next section, we show how to use this reduced cost calculation.

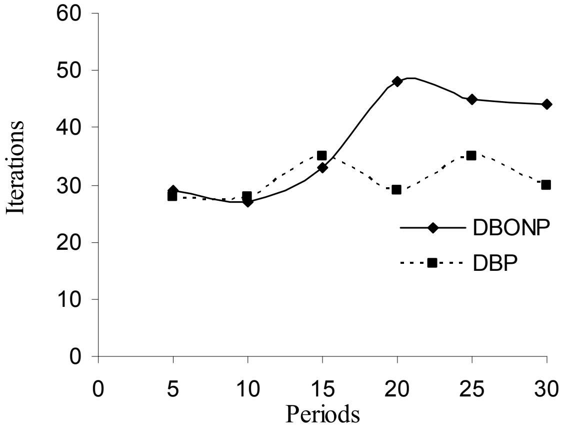


Figure 6: *Number of iterations required to solve different planning horizons by DBP and DBONP.*

4.2 The RCBP algorithm

The RCBP algorithm proceeds as follows:

Step 0. Set $k \leftarrow 1$.

Step 1. Solve M^k as an IP.

Step 2. Add constraints of the form (9) for the integer variables. Solve the restricted master problem as an LP. Obtain dual prices for the trawler scheduling constraints (1), (2), and (9).

Step 3. Calculate $\boldsymbol{w}.rc$ in (11). Scan the reduced costs for all integer variables. Include integer variables with positive reduced cost to the restricted master problem. For the continuous variables from the fish processing part of the problem, there are two options:

> **Option 1:** All continuous variables appear in every restricted master problem.
>
> **Option 2:** Continuous variables with positive reduced cost are added to the restricted master problem at each iteration.

Step 4. For the processing subproblem, solve the processing LP subproblem, and add all positive variables to the restricted master problem as in the DBP approach.

Step 5. If no new variable enters the restricted master problem, then stop. Else $k \leftarrow k+1$ and go back to step 1.

We present the logic of the RCBP algorithm in the form of a flowchart in Figure 7.

4.3 Numerical results

We solved IFPM with different planning horizon models using each option in Step 3. Option 2 takes fewer iterations and less time to solve the fishery model than does Option 1. Results are reported in Table 3.

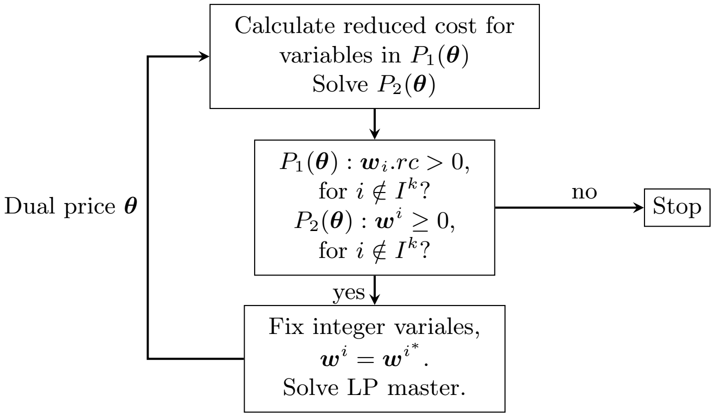

Calculate reduced cost for variables in $P_1(\boldsymbol{\theta})$ Solve $P_2(\boldsymbol{\theta})$
$P_1(\boldsymbol{\theta}) : \boldsymbol{w}_i.rc > 0$, for $i \notin I^k$? $P_2(\boldsymbol{\theta}) : \boldsymbol{w}^i \geq 0$, for $i \notin I^k$?
no
Stop
yes
Fix integer variales, $\boldsymbol{w}^i = \boldsymbol{w}^{i^*}$. Solve LP master.
Dual price $\boldsymbol{\theta}$

Figure 7: *Flowchart of the RCBP procedure.*

Planning Horizon	Description of entering variables	Number of iterations	Solution time (sec.)	RCBP optimal value	Duality gap
5	Option 1	5	39	\$522 764	0%
	Option 2	3	5	\$522 764	0%
10	Option 1	10	142	\$1 065 538	0.02%
	Option 2	5	15	\$1 065 538	0.02%
15	Option 1	11	113	\$1 582 006	0%
	Option 2	5	53	\$1 582 008	0%
20	Option 1	7	109	\$1 877 275	0.15%
	Option 2	4	71	\$1 879 928	0.01%
25	Option 1	6	74	\$2 107 736	0.66%
	Option 2	8	111	\$2 121 887	0%
30	Option 1	8	262	\$2 284 545	0.71%
	Option 2	10	901	\$2 299 648	0.05%

Table 3: *Total profit, number of iterations, and solution times for the RCBP procedure.*

5 Comparison of DBP, DBONP and RCBP solutions

In this section we compare the solutions obtained as well as the number of iterations and solution times required by the DBP, DBONP and RCBP methods in Figures 8–10. The RCBP algorithm is the best among the methods we developed. It requires the smallest solution time to solve, requires fewer iterations and yields better solutions. We further investigated these methods using three different problem instances under many different catch rate scenarios. The numerical results reported here are consistent with those otained for the other problem instances.

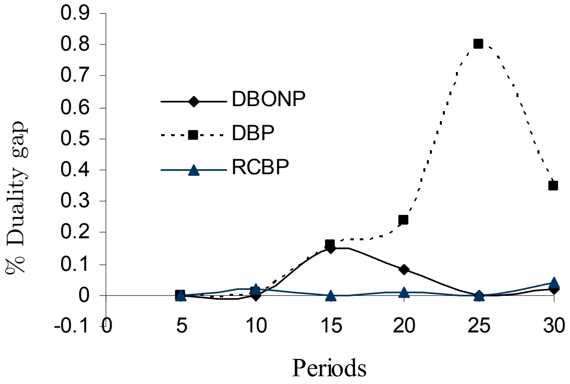


Figure 8: *% Duality gap of DBP, DBONP, and RCBP.*

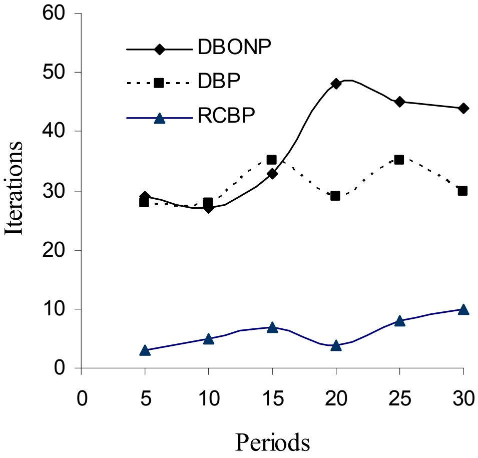


Figure 9: *Number of iterations required to solve DBP, DBONP, and RCBP.*

6 Conclusion

In this paper we developed two different column generation algorithms for faster solution of an IFPM. The first is the DBONP method and the second is the RCBP method — both these approaches are based on O'Neill pricing.

In the RCBP method, we solved only easy LP subproblems and avoided the need to solve IP subproblems. Instead of solving the IP trawler scheduling subproblem, we calculated the reduced cost for each variable, choosing variables with positive reduced cost to bring into the restricted master problem.

Compared to the DBP method alone, we found that the DBONP algorithm took slightly longer, but tended to produce better solutions. However, the RCBP method is both faster and gives better solutions than the DBP approach, and in some cases than the DBONP method.

Note that we never employed a specialized branch-and-bound technique, except for that native to CPlex in the restricted master problem and subproblems. It therefore appears that the combination of DBP and O'Neill pricing approaches may facilitate the develop-

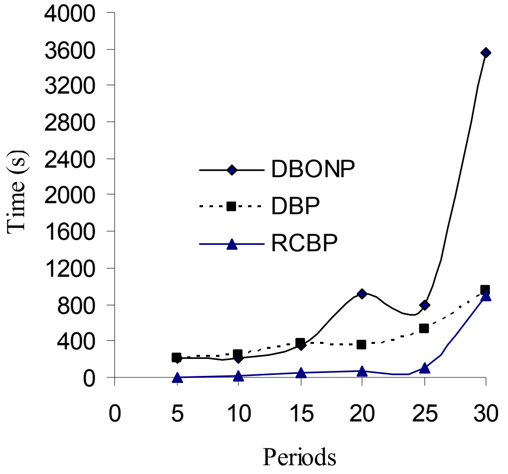


Figure 10: *Solution times required to solve DBP, DBONP, and RCBP.*

ment of a range of new column generation algorithms that may prove effective for integer programs.

References

[1] De Carvalho JMV, 1998, *Exact solution of cutting stock problems using column generation and branch-and-bound*, International Transactions in Operational Research, **5(1)**, pp. 35–44.

[2] Fourer R, Gay DM & Kernighan BW, 1993, *AMPL: A modelling language for mathematical programming,* Curt Hinrichs Publishing, Pacific Grove (CA).

[3] Geoffrion AM, 1974, *Lagrangean relaxation for integer programming*, Mathematical Programming Study, **2**, pp. 82–114.

[4] Gomory RE & Baumol WJ, 1960, *Integer programming and pricing,* Econometrica, **28(3)**, pp. 521–550.

[5] Gunn EA, Millar HH & Newbold SM, 1991, *A model for planning harvesting and marketing activities for integrated fishing firms under an enterprise allocation scheme*, European Journal of Operational Research, **55(2)**, pp. 243–259.

[6] Hasan MB and Raffensperger JF, 2006, *A mixed integer linear program for an integrated fishery*, ORiON, **22(1)**, pp. 19–34.

[7] Hasan MB & Raffensperger JF, 2007, *A decomposition based pricing method for solving a large-scale MILP model for an integrated fishery*, Journal of Applied Mathematics and Decision Science, **2007**, Article ID 56404, Available from: `http://www.hindawi.com/journals/jamds/volume-2007/regular.1.html`.

[8] Jensson P, 1988, *Daily production planning in fish processing firms*, European Journal of Operational Research, **36(3)**, pp. 410–415.

[9] Mamer JW & McBride RD, 2000, *A decomposition-based pricing procedure for large-scale linear programs: An application to the linear multi-commodity flow problem*, Management Science, **46(5)**, pp. 693–709.

[10] Martin K & Sweeney DJ, 1983, *An ideal column algorithm for integer programs with special ordered sets of variables*, Mathematical Programming, **26(1)**, pp. 48–63.

[11] Martin KR, Sweeney DJ & Doherty ME, 1985, *The reduced cost branch-and-bound algorithm for mixed integer programming*, Computers & Operations Research, **12(2)**, pp. 139–149.

[12] Mikalsen B & Vassdal T, 1981, *A short term production planning model in fish processing*, pp. 223–233, in Haley KB (Ed.), *Applied operations research in fishing*, Plenum Press, New York (NY).

[13] O'Neill RP, Sotkiewicz PM, Hobbs BF, Rothkopf MH & Stewart WR, 2005, *Efficient market-clearing prices in markets with non-convexities,* European Journal of Operational Research, **164(1)**, pp. 269–285.

[14] Raffensperger JF & Schrage L, 2008, *Scheduling training for a tank battalion: How to measure readiness*, Computers and Operations Research, **35(6)**, pp. 1844–1864.

Volume 24 (2), pp. 131–156
http://www.orssa.org.za

ORiON
ISSN 0529-191-X
©2008

The identification of possible future provincial boundaries for South Africa based on an intramax analysis of journey-to-work data

JH Nel[*] SC Krygsman[†] T de Jong[‡]

Received: 12 August 2008; Revised: 24 October 2008; Accepted: 27 October 2008

Abstract

National census data contain information on place of residence and place of work. It is possible to combine this information and create journey-to-work flows. The process of establishing these flows are presented in this paper. The intramax method is explained and used to identify functional regions based upon these flows. Interesting applications, such as the demarcation of regions in South Africa are considered and solutions to disputed areas are put forward. The process of the creation of the current provincial boundaries are discussed. New boundaries, based on the intramax analysis of the journey-to-work data are proposed for four or five new provinces. Results compare favourably with those from a principal component and cluster analysis, which has previously been used to demarcate the South African space economy into a hierarchy of development regions.

Key words: Journey-to-work flows, provincial boundaries, intramax method, principal component analysis, functional regions, demarcation of regions.

1 Introduction

On 28 May 1993, the Negotiating Council of the Multiparty Negotiating Process established a fifteen-person commission to make proposals for new internal boundaries in South Africa [7]. The resulting Commission on the Demarcation/Delimitation of Regions (the CDDR) held its first meeting on 8 June 1993 and reached a decision by 31 July 1993. After six weeks, the commission more than doubled the number of provinces, from the initial four to the current nine provinces [7]. No meaningful time was allotted for public consultation, and the commissioners took as the initial draft the nine planning regions established by the Development Bank of Southern Africa between 1982 and 1988 [7]. Only one month of the CDDR's itinerary was devoted to gathering of testimony, and in reaction to broad

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

[†] Department of Logistics, University of Stellenbosch, Private Bag X1, 7602, Matieland, South Africa.

[‡] Department of Human Geography and Urban and Regional Planning, Faculty of Geosciences, Utrecht University, PO Box 80115, 3508 TC, Utrecht, The Netherlands.

public criticism, a further three months were allocated, beginning in August 1993. After the commission submitted its report, politicians hacked away and swapped magisterial districts in order to reach a final party agreement. From a party-political point of view, the negotiations resulted in demarcations that offered important minority parties a future base for provincial power [9].

Griggs [9] noted that these political party compromises resulted in two main problems: too many non-viable provinces, and boundary conflicts. Only Gauteng and the Western Cape provinces had thriving metropolitan regions, no former 'homelands' and had the potential in 1994 to generate enough income to finance their own administrations. He noted that most of the other provinces lack resources, infrastructure and capacity, and require central government support. Furthermore, more than fourteen years after the final provincial map was produced by multiparty negotiations, there were still eight or more active disputes. Griggs [9] proposed increased public participation by referenda as a way of resolving many of the issues.

Boundaries should be drawn so as to minimise the splitting of communities. South Africa's current spatial organisation and delineation are characterised by internal conflicts. Figure 1 shows, on a national level, the disputed areas after the 1993 delineation of provincial boundaries. Ramutsindela and Simon [19] described the process of negotiating between the provinces in the time period after 1993 as "horse-trading." Northern Province (currently Limpopo Province), for example, demanded that the towns of Groblersdal and Marble Hall, which are part of Mpumalanga, be transferred to the Northern Province to compensate for relinquishing Bushbuckridge. On the other hand, the people of Bushbuckridge have been campaigning for years to be incorporated into Mpumalanga and not Limpopo Province. While belonging to Limpopo Province, research has shown that many (95–98%) of the residents prefer incorporation into Mpumalanga, with their reasons advanced being geographical proximity and economic ties. Residents argue that this is where they work and undertake their shopping [19].

According to Smith [21], the former chairperson of the ANC, Mosiuoa Lekota, became the most senior member of the party to date to suggest that a reduction in number from the current nine provinces should be considered seriously. According to Ngalwa [18] a discussion document, which moots a four or five province option, was drafted and circulated in government during 2007. Some ministers in the previous cabinet, including Finance Minister Trevor Manuel, Defence Minister Mosiuoa Lekota and Minister Sydney Mufamadi have publicly suggested that the number of provinces should be reduced. They also requested that proper research should be conducted to review the performance of the provincial system before deciding on their future.

It is clear that the process of demarcation cannot be examined without taking political motives into consideration, whilst the needs of people living and working in the provinces should also be considered. Functional regions based on activities of households and businesses are the people's way of deciding to which areas they belong.

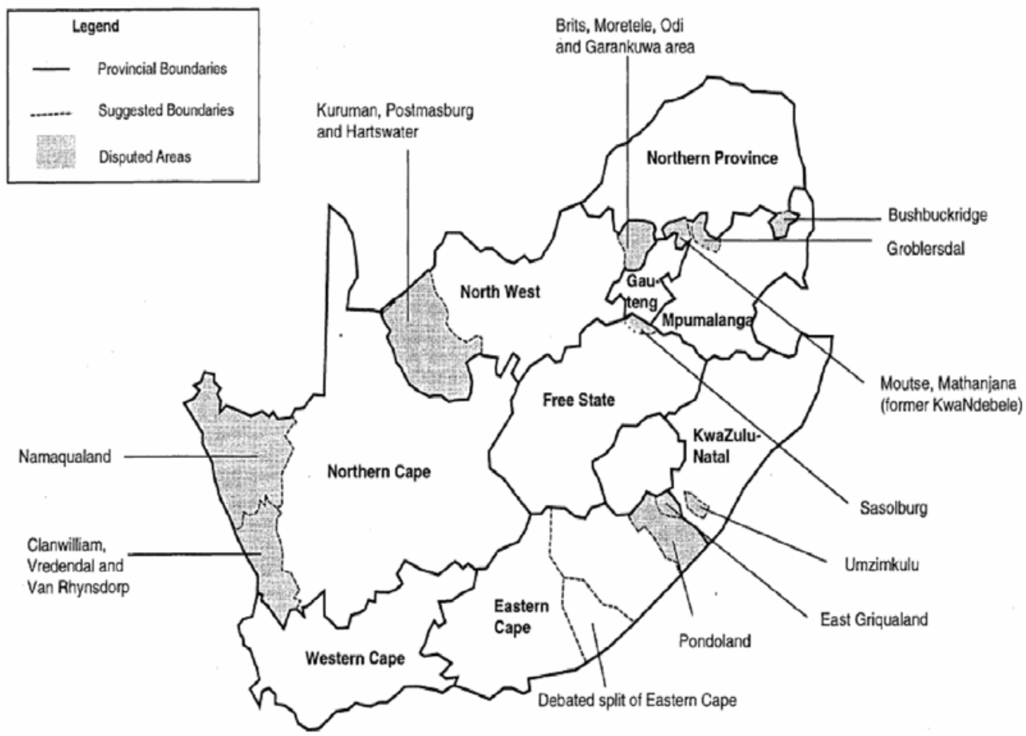


Figure 1: *Disputed areas after the 1993 delimitation of provincial boundaries [19]. (Original source: Saturday Weekend Argus, 13–14 January, 1996, p20.)*

2 Functional regions

The concept of a functional region or functional area may be described in many ways. Feldman *et al.* [4] described it as an area defined by business and economic activities rather than by administrative or historic boundaries. A functional region was also defined by Brown and Holmes [1] as an area or locational entity which enjoys more interaction or connection within its boundaries than with outside areas.

Functional regions may also be seen as areas in which the businesses concerned recruit most of their labour force. The quality of functional region demarcation has a strong influence on both productivity and prosperity. The functional region is a phenomenon arising exclusively from human activity, and is best described as a community of interests. In respect of human activity, specific reference is paid to transport, work and residential choice and therefore functional regions are a spatial manifestation of social organisation. Functional regions represent the day-to-day regions in people's lives, *i.e.* they are created by the various choices and decisions of individual people and enterprises.

Feldman *et al.* [4] noted that the best-established technique for a functional approach to area grouping is to identify boundaries across which relatively few people commute. Mitchell *et al.* [17] reasoned that journey-to-work data provide information about the interaction between spatial units and are a useful basis for defining functional regions. A commuting area is conceived as a geographical area within which there is a high degree of interactivity and may be seen as an appropriate spatial region to capture the interplay between labour supply and demand. Mitchell *et al.* [17] concluded that aggregations of journey-to-work data reflect economic behaviour rather than administrative structures.

The objective of this paper is to analyse journey-to-work flow data and to use intramax analysis to establish functional regions in South Africa in general, but specifically at provincial level. The purpose is to demonstrate how functional regions differ from administrative regions (which are more than likely demarcated in terms of political or ideological philosophy). A further objective is to test whether the functional regions or provinces identified by the intramax analysis are economically viable regions.

3 Literature review on analysis of flow data

Journey-to-work data may be captured in a network flow problem, which consists of a collection of transhipment nodes connected by directed arcs in both directions. Figure 2 contains an example of journey-to-work data between four regions.

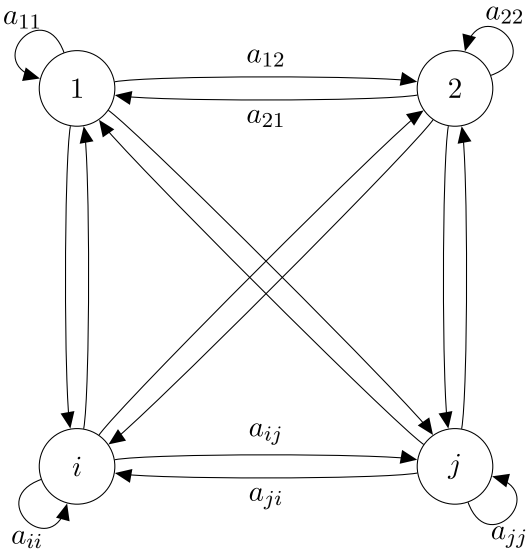


Figure 2: *Example of a network of flows between 4 regions.*

A schematic representation of a so-called interaction matrix is provided in Table 1, where rows are designated as origins and columns are destinations. Marginal totals may be interpreted as follows: $O_i = \sum_j a_{ij}$ and $D_j = \sum_i a_{ij}$ represent the total outflow from region i and total inflow into region j respectively.

Ward [26] developed a hierarchical aggregation procedure which is a routine for searching through groups of data to find which pair of basic data units shows the greatest mutual similarity with respect to specified characteristics. Given k subsets, this method permits their reduction to $k-1$ mutually exclusive subsets by considering the union of all possible $\binom{k}{2} = k(k-1)/2$ pairs that can be formed and accepting the union with which an optimal value of the objective function is associated. The process may be repeated until all subsets

	Region 1	Region 2	$\ldots$	Region j	$\ldots$	Total
Region 1	a_{11}	a_{12}	$\ldots$	a_{1j}	$\ldots$	$\sum_j a_{1j} = O_1$
Region 2	a_{21}	a_{22}	$\ldots$	a_{2j}	$\ldots$	$\sum_j a_{2j} = O_2$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
Region i	a_{i1}	a_{i2}	$\ldots$	a_{ij}	$\ldots$	$\sum_j a_{ij} = O_i$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
Total	$\sum_i a_{i1} = D_1$	$\sum_i a_{i2} = D_2$	$\ldots$	$\sum_i a_{ij} = D_j$	$\ldots$	$\sum_i \sum_j a_{ij} = n$

Table 1: *Journey-to-work interaction matrix.*

are in one group.

Ward [26] defines a functional relation that provides a "value reflecting" number as an objective function. It is common practice to use the mean value to represent all scores. The loss in information that results from treating scores as one group may be indicated by a "value-reflecting" number such as the *Error Sum of Squares* (ESS). The ESS is given by

$$ESS = \sum_{i=1}^{m} (x_i - \overline{x})^2 = \sum_{i=1}^{m} x_i^2 - \frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)^2,$$

where x_i is the score of the i-th individual and where m denotes the number of individuals. If scores are classified in groups, the grouping can be evaluated as the sum of the ESS values, that is

$$ESS_{\text{Groups}} = ESS_{(\text{Group 1})} + ESS_{(\text{Group 2})} + \ldots$$

The same procedure can be used for aggregation of flow data if the objective function is respecified in terms of the two-directional flow between two regions. It will be necessary to consider two entries for this purpose, namely a_{ij} and a_{ji}, for all $i \neq j$.

Masser and Brown [14] formulated as objective the maximisation, at each stage of the grouping process, of the difference between the observed values, a_{ij}, and "expected values" a_{ij}^*, which are derived similarly to the expected frequency of the cell in row i and column j in a contingency table for the Chi-square test, namely

$$a_{ij}^* = \frac{O_i D_j}{n}, \quad \text{where } n = \sum_i \sum_j a_{ij}.$$

The objective is therefore to

$$\underset{i \neq j}{\text{maximise}} \quad \left\{(a_{ij} - a_{ij}^*) + (a_{ji} - a_{ji}^*)\right\}.$$

The entries a_{ij} are standardised so that

$$\sum_i \sum_j a'_{ij} = 1,$$

where $a'_{ij} = a_{ij}/n$. It can be shown that the standardised objective is to

$$\underset{i \neq j}{\text{maximise}} \quad \left\{ (a'_{ij} - a'^{*}_{ij}) + (a'_{ji} - a'^{*}_{ji}) \right\}.$$

Contiguity constraints may be introduced to restrict the search for potential pairings. These constraints may take the form $c_{ij} = 1$, if movement of a basic data unit from i to j is allowed, and $c_{ij} = 0$ otherwise.

The intramax analysis is a stepwise analysis. During each step two areas are grouped together and the interaction between the two areas becomes internal (or intrazonal) interaction for the new resulting area. This new area now takes the place of the two parent areas at the next step of the analysis. So with N areas, all areas are grouped together into one area after $N - 1$ steps and all interaction is intrazonal. The outcome of an intramax analysis may be presented in dendrogram form.

According to Tyree [24], the alternative concept of *mobility ratios* was developed by three sociologists, Natalie Rogoff, David Glass and Gösta Carlsson [24], working independently on the problem of intergenerational occupational mobility. A matrix of frequencies of occupations of respondents by occupations of fathers may be converted into matrices of inflow and outflow percentages. The mobility ratio M_{ij} is simply the ratio

$$M_{ij} = \frac{a_{ij}n}{O_i D_j}, \quad i \neq j, \tag{1}$$

of the frequency observed in a cell to the frequency expected under the assumption of statistical independence. Hollingworth [13] studied migration between Scottish executive areas and also defined the mobility index as (1). The value of the objective function in this case is then $M_{ij} + M_{ji}$, which was used as a symmetric measure of the mutual association of areas i and j.

Hirst [12] noted that both the objective functions defined by Masser and Brown [14] and Hollingworth [13] is inappropriate, because of the influence of unequal marginal distributions which define the expected frequencies. For example, the ratio or difference between the observed and expected values will tend to increase for cells in those rows and columns with large sums. Since the objective function is recalculated after each step in the grouping procedure, this bias will be cumulative.

Tyree [24] suggested that the interaction matrix should first be adjusted to achieve an arbitrary origin-destination distribution. This may be accomplished iteratively by standard matrix operations: rows are scaled initially to sum to a given total, and then columns are scaled to sum to the same total. This procedure is repeated until sufficient convergence occurs to a matrix in which all row and column sums are simultaneously equal. Hirst [12] claimed that it can be proved that this matrix exists, is unique, and that the iterative procedure is convergent. He suggested that a possible solution would be to divide $a'_{ij} - a'^{*}_{ij}$ by a'^{*}_{ij}, with a'^{*}_{ij} corrected for blank entries in the interaction matrix as proposed by Goodman [7], but noted that results will still tend to favour small zones, because of the differences between the values obtained for small as opposed to large zones. Hirst also remarked that an increasing number of heuristic techniques has become available, and that a need for comparative evaluation of their respective merits and areas of application has arisen.

Masser and Scheurwater [15] evaluated three methods for functional regionalisation, namely the *functional distance method* [1] (not discussed in this paper), the *iterative proportional fitting based procedure* (IPFP) [20] (not discussed in this paper) and the intramax procedure [14]. Their conclusion was that the intramax procedure is the only one of the three procedures which explicitly identifies regions that have more (direct) interaction with each other than with other areas at each stage of the grouping process. It has a practical advantage over the other two methods, because it only involves a series of direct comparisons between the observed and expected values that are calculated by the multiplication of the respective row and column totals. This avoids the complex set of matrix manipulations that are required for the other two methods. The intramax procedure may be more readily applied to large data sets and may be adapted more easily to deal with large, sparse matrices. Masser and Scheurwater [15] also noted that stronger connections would appear between pairs of smaller zones containing a relatively low proportion of intrazonal interaction than between pairs of larger zones containing a relatively high proportion of intrazonal interaction and that the former would tend to fuse together before the latter. They reason that this bias noted by Hirst [12], far from being a disadvantage, is in fact advantageous and that it is a reflection of the inherent characteristics of the structure of spatial interaction in the matrix.

Fischer *et al.* [5] compared the intramax procedure with the IPFP-based graph approach (not discussed in this paper) and came to the conclusion that the intramax approach is superior to the IPFP-based graph-theoretical one, because the results are easily interpretable in terms of functional regions. The intramax approach also leads to spatial groupings which show more interaction with each other than with other regions.

Brown and Pitfield [2] noted that the objective function was reformulated in literature appearing after the comment of Hirst [12] to

$$\underset{i\neq j}{\text{maximise}} \quad \left\{ \frac{a'_{ij} - a'^{*}_{ij}}{a'^{*}_{ij}} + \frac{a'_{ji} - a'^{*}_{ji}}{a'^{*}_{ji}} \right\}.$$

They remarked that this revised form of the objective function was employed in all subsequent applications of the procedure, and may be re-expressed a little more simply as

$$\underset{i\neq j}{\text{maximise}} \quad \left\{ \frac{a'_{ij}}{a'^{*}_{ij}} + \frac{a'_{ji}}{a'^{*}_{ji}} \right\}. \tag{2}$$

The reason for this is that the part that is subtracted in each term is constant and may thus be ignored. This objective function is also discussed by Brown and Pitfield [2]. The resulting formula is strikingly similar to the mobility ratios employed by Hollingworth [13], where

$$\frac{a'_{ij}}{a'^{*}_{ij}} = \frac{\frac{1}{n}a_{ij}}{\frac{1}{n}\sum_j a_{ij}\frac{1}{n}\sum_i a_{ij}} = \frac{na_{ij}}{\sum_j a_{ij}\sum_i a_{ij}}.$$

4 The software suite *Flowmap*

Flowmap [25] is a software suite developed at the University of Utrecht, the Netherlands (in conjunction with the CSIR, South Africa). The suite performs geographical analyses and specialises in displaying *interaction data* (such as commuting and migration flows), *interaction analysis* (such as accessibility analysis), *network analysis*, and *interaction modelling.* The program uses several kinds of data, which may be grouped into three classes: maps, flow data and distance tables.

Flowmap uses intramax analysis to identify functional regions from an interaction matrix. "*The objective of the intramax procedure is to maximise the proportion within the group interaction at each stage of the grouping process, while taking account of the variations in the row and column totals of the matrix*" [22]. This implies that in this particular case two areas are grouped together for which the objective function

$$\frac{T_{ij}}{O_i D_j} + \frac{T_{ji}}{O_j D_i} \tag{3}$$

is maximised where T_{ij} is the interaction between origin location i and destination location j, and where

$$O_i = \sum_j T_{ij} \text{ and } D_j = \sum_i T_{ij}.$$

This is similar to (2) and the method of Hollingsworth [11], but the constant n is omitted. The objective function in (3) can only be calculated for all $D_j > 0$ and for all $O_i > 0$. In Flowmap actual flow values are used, hence T_{ij} instead of a'_{ij}, but that should not have any effect on the results as no comparisons are made; the maximum relationship is merely sought at each aggregation step. The use of the above objective function is also substantiated in a thesis by Floor and de Jong [6].

5 Methodology and data

The methodology employed and the data used in this paper are described in this section.

5.1 Journey-to-work data and intramax analysis

The data used in this paper all derive from the 2001 South African Census [22]. The question was asked "In the seven days before 10 October did (the person) do any work for pay (in cash or in kind), profit or family gain, for one hour or more? If "Yes," does (the person) work in the same sub-place in which s/he usually lives?" If "No," the main place of work was recorded. The definition of work includes formal, informal and seasonal work. The database of all persons between the age of 15 to 65 represented 28 427 129 individuals. A subdatabase was prepared at the request of the authors containing amongst others, the following fields: main place code and main place of work code. For reasons of confidentiality, records were totalled and frequencies in each category, defined by the field names, were calculated. The resulting subdatabase contained a total of 1 890 827 records. Part of the confidentialising process was to change frequencies of 1 and 2 according to an algorithm, as follows:

- Change a frequency of 1 to 0 in two thirds of the cases;
- Change a frequency of 1 to 3 in one third of the cases;

- Change a frequency of 2 to 0 in one third of the cases and
- Change a frequency of 2 to 3 in two thirds of the cases.

Certain records were not considered for the intramax analysis[1]. The records not considered included 198 758 records (18 792 972 individuals) for which the main place of work were marked as not applicable, due to the fact that these records represent persons unemployed or not economically active. A further 65 556 records (156 899 individuals) were deleted, because the main place of work was "unspecified." Of the remaining records, a further 107 818 records (182 237 individuals) were removed due to the fact that they replied "No" to the question "Is this your usual place of stay?" A further 3 997 records (9 679 individuals) were deleted because their economic activity was marked "Not economically active." Some further 32 290 records (59 933 individuals) were deleted because the main place names could not be matched (the province code was given instead of the code of a specific main place).

The following data cleanup was also performed and the interaction data were adjusted accordingly:

- 7 islands were removed,
- 638 fully embedded regions were dissolved,
- 24 main places without interaction were dissolved,
- 46 main places with only intrazonal interaction were dissolved.

Intramax analysis was therefore applied to a total of 861 939 records involving 2 393 extended main places.

5.2 Principal component and cluster analysis

It is important to validate the results, *e.g.* to use different methods with different variables to establish whether boundaries and regions defined by the intramax analysis may be viewed as socio-economic functional regions. Harmse [10], using mainly 1996 Census data, demarcated the South African space economy into a hierarchy of five development regions, *i.e.* a highly developed metropolitan core region, an upward transitional region, a downward transitional region, a resource frontier region and special problem regions. Harmse *et al.* [11] reapplied this technique on 2001 Census data, using the following socio-economic variables:

- Population density,
- Birth rate,
- Youthful dependency ratio,
- Per capita income,
- Number of persons per 10 000 earning more than R51 201 per month,
- Percentage of people employed,
- Number of people per 1 000 working in agriculture,
- Number of people per 1 000 working in secondary sector,
- Number of people per 1 000 working in financial services,

[1] Only employed persons for whom journey-to-work data per main place could be calculated are included.

- Percentage of people living in urban areas,
- Number of people per 1 000 with more than 12 years of education,
- Percentage of households whose refuse is removed by local authority,
- Percentage of households living in formal housing,
- Percentage of households using electricity for cooking, and
- Percentage of households with piped water in the house.

A data matrix consisting of variables and municipalities as spatial units was compiled as input for the multivariate analysis. Using principal component analysis, the large number of correlated variables was reduced to fewer variables that captured most of the variation in the original variables. Cluster analysis was then used to identify groups of similar main places in order to reduce the number of spatial units to a more manageable number, using the scores of the different principal components. By applying Ward's cluster analysis, the semi-partial R^2 values generated was used to identify a significant grouping. The mean score on principal component I for these different groups was calculated in order to determine how the groups may be assigned to the different regional types [10]. The results are reported in the following section.

The Community Profile database [23] of Census 2001 was accessed in SuperCross format at main place level. The weighted mean, median and inter quartile range of some socio-economic variables were calculated for a proposed five-province scenario and were compared using Bonferroni multiple comparisons.

6 Results

6.1 Intramax analysis

A total of 2 392 iterations were required in the intramax process. At each stage of the clustering process, two regions with the strongest possible commuting ties were aggregated. These two regions were then seen as one region, and commuting between these two regions become intrazonal. The total number of regions was thus reduced by one region and the interaction matrix was reduced by one row and one column. This process was repeated until only one region remained (theoretically), in which all commuting is intrazonal.

During this process, there were 18 minor areas exhibiting unusually large flows, which were not clustered — they remained original main places. For example, the Kgalagadi Park (main place 39 302) in the Northern Cape has only outside commuter links and comprises a total of 7 persons all residing/working in the Saldanha area over 800 km away. The flows to/from the 18 problem main places were removed. Other surviving unlinked main places were also removed or dissolved, yet ensuring that this process did not impact on the boundaries of the remaining clusters.

The clustering process continued until 80% of the interzonal interaction internalised with 70 functional areas (blocks) remained. The results are shown in the dendrograms in Figures 3–7 and the map in Figure 8.

In Figure 3, the Nama Khoi region includes the town of Springbok and the Richtersveld National Park. This fuses with the Matzikama region, which includes Van Rhynsdorp,

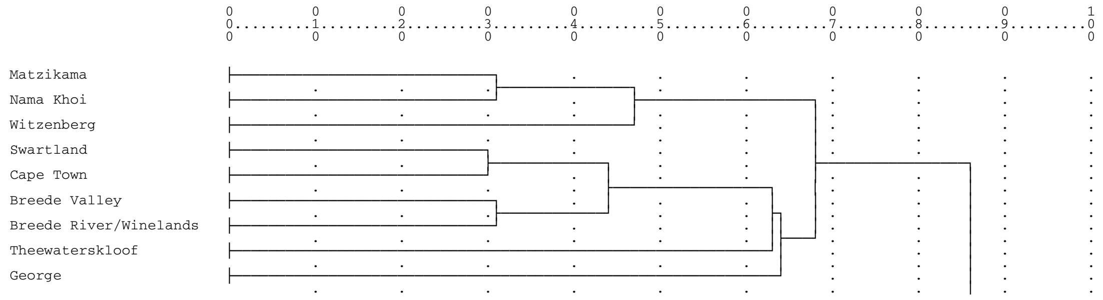


Figure 3: *Dendrogram of the last nine regions in the Western Cape.*

Vredendal, Calvinia, Sutherland, Carnarvon and others. Approximately 31% of all the journey-to-work flows in and out of these regions are intrazonal for this new aggregation. In the next step, this region fuses with the Witzenberg region, which includes places such as Ceres, Tulbach and Clanwilliam (47% intrazonal). The Cape Town region (including Stellenbosch, Strand, Paarl, *etc.*) fuses with the Swartland region, which includes Moorreesburg, Malmesbury, Saldanha and others (30% intrazonal). The Breede River/Winelands area (Montagu, Swellendam, *etc.*) fuses with the Breede Valley area (Worcester, Robertson, *etc.*) (31% intrazonal), which then fuses with the Cape Town / Swartland cluster (44% intrazonal). This cluster then fuses with the Theewaterskloof cluster (63% intrazonal), which includes the Overberg region. The George cluster (which includes most of the Garden Route) fuses with the larger Cape Town cluster (64% intrazonal), and finally this fuses with the Witzenberg / Nama Khoi / Matzikama cluster (68% intrazonal). ('First Province' of the nine last clusters shown in Figure 8.)

In Figure 4, the Paradise Beach and Kouga areas (Jeffreys Bay, Tsitsikamma National park, Stormsriver area) merges with the Port Elizabeth area (30% intrazonal), and fuses in the next step with the Ubuntu area (including Victoria West, Richmond, *etc.*) and the Inxuba Yethemba region (Cradock, Middelburg, *etc.*) (31% intrazonal). This region then fuses with the Graaff Reinet area (47% intrazonal). The Grahamstown and East London regions (31% intrazonal) fuse with the Lusikisiki (including Flagstaff), Queenstown, Kokstad and Marburg (Port Shepstone and others) regions (63% intrazonal). This region then fuses with the greater Port Elizabeth cluster (67% intrazonal). ('Second Province' of the nine last clusters shown in Figure 8.)

In the second part of Figure 4, the Durban and Pietermaritzburg regions (34% intrazonal) merge with the Umvoti (Greytown, Kranskop, *etc.*) and Stanger regions (43% intrazonal). The Myeni/Ntsinde area (Jozini, *etc.*) fuses with the Richards Bay area (31% intrazonal), and this region fuses next into the greater Durban region, followed by the Mkhambathini region, which looks like a region on its own (Camperdown, *etc.*) (69% intrazonal). ('Third Province' of the nine last clusters shown in Figure 8.)

The third part of Figure 4 consists of the Ladysmith region (including Escourt, *etc.*) and the Newcastle region (including the Volksrust and Standerton areas in the current Mpumalanga Province) (45% intrazonal). ('Fourth Province' of the nine last clusters shown in Figure 8.)

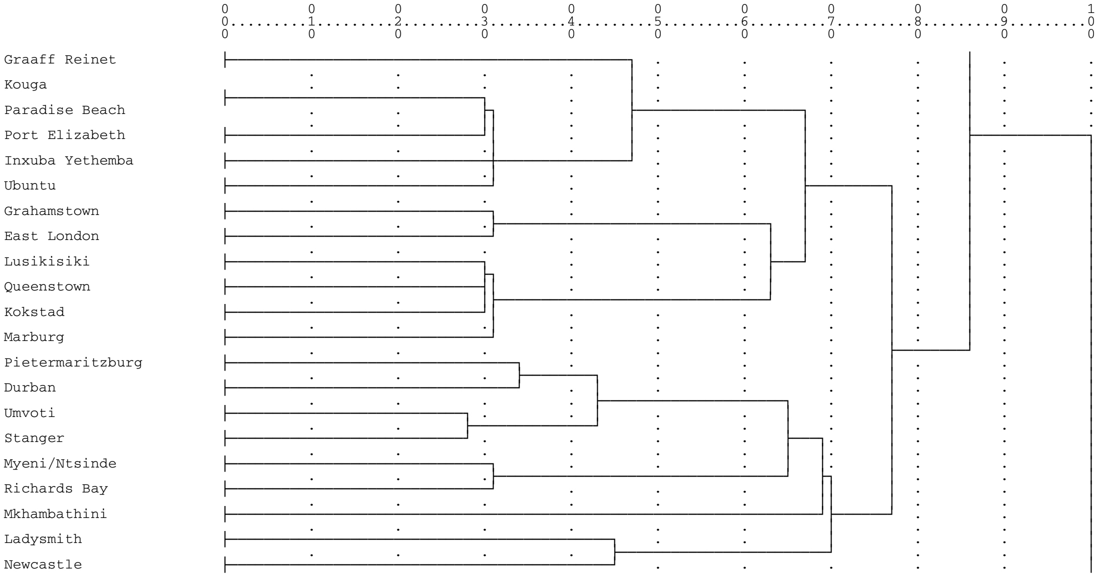


Figure 4: *Dendrogram of the last twenty-one regions in the Eastern Coastal region. Kouga and Paradise Beach merge at the very start of the procedure resulting in less than 0.5% intrazonal interaction.*

In Figure 5 the Dukathole (including Jamestown in the current Eastern Cape and Aliwal North) and Kopanong (Bethulie, Philippolis, *etc.*) areas fuse (31% intrazonal). This region then fuses with the Naledi (Van Stadensrus, Wepener, *etc.*) and Bloemfontein regions and the resulting region results in 47% intrazonal flows. The Setsoto (Clocolan, Ficksburg, Senekal), Nketoana (Lindley, Reitz, Petrus Steyn), Phuthaditjhaba and Phumelela (Memel, Vrede and Warden) regions merge (47% intrazonal flow) and this region fuses with the greater Bloemfontein region (68% intrazonal). The Tswelopele (Bultfontein and Hoopstad), Maquassi Hills (Leeudoringstad and Makwassie regions in the current North West Province), Thabong (Odendaalsrus and Welkom), Nala (Bothaville regions), Moqhaka (Kroonstad and Steynsrus) and Klerksdorp region in the current North West Province fuse (48% intrazonal) which then fuse with the previous region, including Bloemfontein, (69% intrazonal) to form the ‘Fifth Province’ of the nine last clusters shown in Figure 8.

In Figure 6, the regions of Kai !Garib (Augrabies, Kakamas and other regions in the Northern Cape) and !Kheis (Groblershoop, Grootdrink, *etc.* in the Northern Cape) merge with the Kimberley, Letsemeng (Petrusburg, Jacobsdal, *etc.* in the Free State) and Vryburg (also Schweizer-Reneke and other regions in the North West Province) regions (47% intrazonal). This region merges with the Rustenburg and Mafikeng fusion (69% intrazonal), resulting in the ‘Sixth Province’ of the nine last clusters shown in Figure 8.

The Modderfontein region merges with the Boksburg, Johannesburg fusion (28% intrazonal), and the Evaton (Vaal Triangle, including Sasolburg in the Free State) and Lesedi (Heidelberg, Nigel Springs) regions then fuse into the Johannesburg region (46% intrazonal), then follow the Pretoria region, the Randfontein region and lastly the Merafong (Carltonville, Khutsong and others) region (66% intrazonal). This results in the ‘Seventh

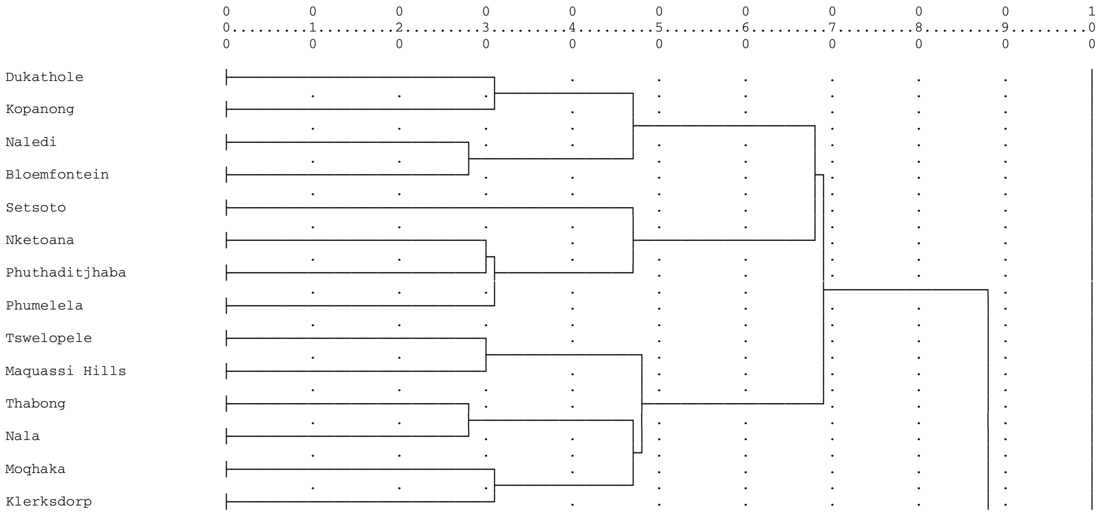


Figure 5: *Dendrogram of the last fourteen regions in the Central region.*

Province' of the nine last clusters shown in Figure 8.

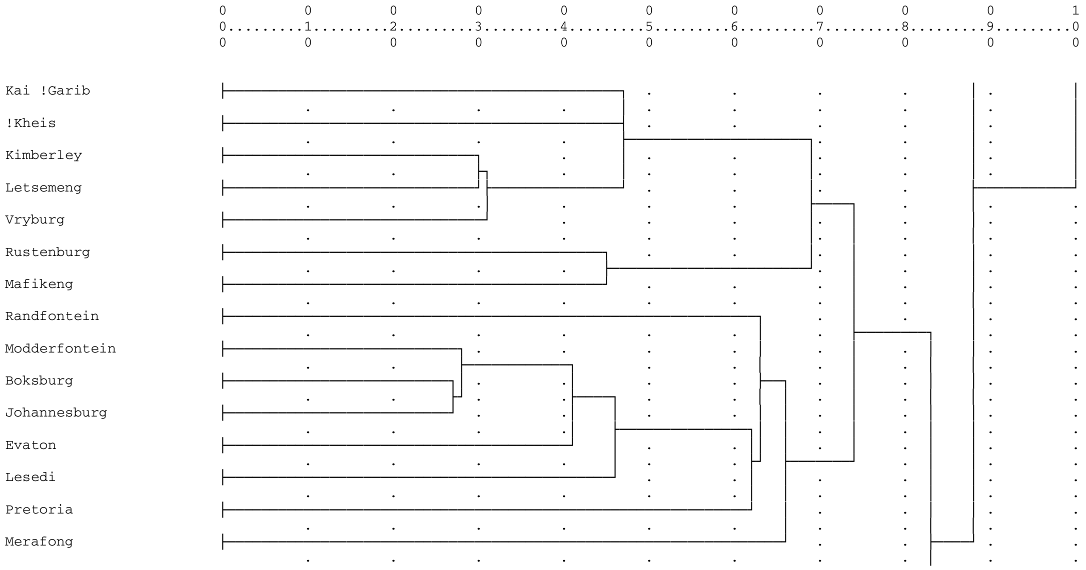


Figure 6: *Dendrogram of the last fifteen regions in the Gauteng region.*

Figure 7 is a fusion of the remaining regions of the Limpopo Province and the Mpumalanga Province. Msukaligwa (Ermelo region) and Mkhondo (Piet Retief region) merge with Embalenhle (Kinross, Leslie, Evander regions) and Witbank region (48% intrazonal). The Highlands (Dullstroom, Machadodorp regions) and Mbombela (Nelspruit region) regions merge with the greater Witbank region (67% intrazonal) which completes the 'Eighth Province' of the nine last clusters shown in Figure 8.

Finally the Greater Tzaneen (including Haenertsburg, Letsitele, *etc.*) and Phalaborwa (including Gravelotte, Die Eiland, *etc.*) regions merge (28% intrazonal). The Pietersburg region (Polokwane) fuses with the Tzaneen region (45% intrazonal), followed by a fusion with the Tshivhase region (Thohoyandou, Gijana, *etc.*) and lastly the Bela-Bela region (Warmbaths, Nylstroom, *etc.*), with a total of 67% intrazonal flow, resulting in the 'Ninth Province' in Figure 8.

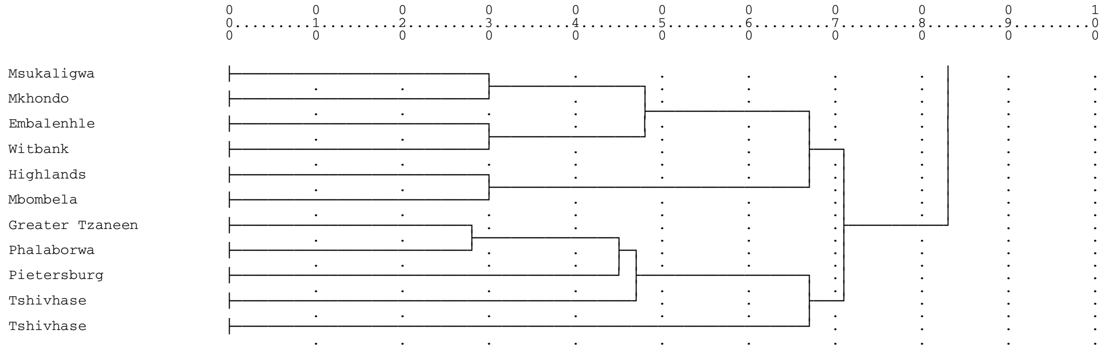


Figure 7: *Dendrogram of the last eleven regions in the Northern region.*

The dots in Figure 8 are proportional to the volume or level of intrazonal interaction per new functional block and the nine-province division shown in the figure has been constructed by means of the intramax method from the interaction between the remaining 70 blocks.

Tables 2 and 3 show the commuter flows crossing provincial boundaries in the current context and the proposed new situation with nine provinces. The number of boundary-crossing commuters is reduced in the intramax solution by over 45% from 287 000 to approximately 157 000. The total workforce is approximately 9.4 million, but only some 2.7 million workers commute daily between different main places. The difference between the total workforce and the part of the workforce that actually commutes explains the difference between the numbers in Tables 2 and 3 and the numbers given in §5.1.

6.2 Reducing the number of provinces to four or five

The dendrograms in Figures 3 to 7 show that 'Province 4' (Newcastle region) fuses with 'Province 3' (Durban region) (69% intrazonal). Next, the remainder of the Mpumalanga region ('Province 8') fuses with the remainder of the Limpopo Province ('Province 9') (71% intrazonal). Next follows the remainder of the North West region ('Province 6'), which fuses with the greater Gauteng area ('Province 7') (73% intrazonal). The Eastern Cape region ('Province 2') clusters together with the KwaZulu-Natal region ('Province 3') (77% intrazonal) leaving a remainder of five 'provinces', *i.e.* the Western Cape with part of the Northern Cape; an amalgamation of the Eastern Cape and KwaZulu-Natal; an amalgamation of the North West and Free State; an amalgamation of the remainder of the Northern Cape, North West and Gauteng; and an amalgamation of Limpopo Province and the remainder of Mpumalanga (five provinces). In the next step, the newly formed Limpopo Province would fuse with the Gauteng region (four possible provinces),

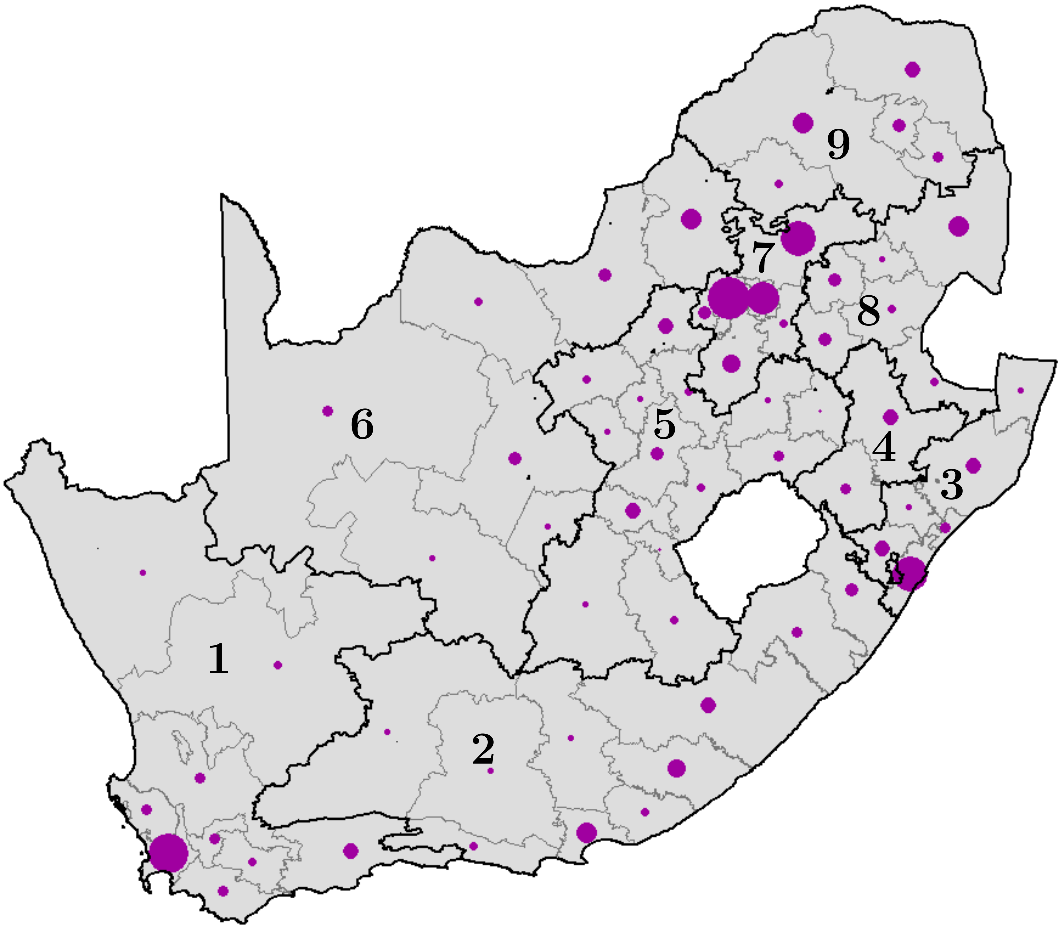


Figure 8: *The remaining nine clusters with dots indicating the relative size of the intrazonal interaction per functional region.*

the Eastern Cape and KwaZulu-Natal region would fuse with the Western Cape region (three provinces) and the Free State region would amalgamate with the Gauteng region (two provinces). The country thus becomes consolidated into a final North-South division.

The boundaries between the Western Cape region and the Eastern Cape / KwaZulu-Natal region are mountainous regions, but it seems that rivers, such as the Orange River and the Vaal River, which were historical boundaries, do not impact as much on the boundaries any longer, because of the accessibility via roads to the nearest major centres.

Figure 9 shows the reduction from nine provinces to four provinces.

6.3 Disputed areas

The disputed area of Bushbuckridge is used as an example to demonstrate how the intramax analysis may be used to resolve similar contentious situations. The current provincial boundary crosses straight through the Buschbuckridge functional area and generates five times more cross-boundary commuting than the alternative suggested by intramax.

Pro- vince	Total flow	EC	FS	GP	KZN	LP	MP	NW	NC	WC
EC	163 998	149 004	1 064	2 870	4 556	665	966	1 279	368	3 226
FS	92 398	880	81 951	5 679	772	350	419	1 059	530	758
GP	1 007 615	5 325	6 781	957 885	6 314	3 483	7 234	14 313	1 351	4 929
KZ	415 992	7 593	992	5826	392 881	1893	2 306	1 174	590	2 737
LP	129 898	1 438	545	5 926	1 399	108 316	10 215	1 098	436	525
MP	132 701	1 023	893	33 515	1 666	2 093	91 377	1 196	415	523
NW	240 823	1 100	1 640	86 974	755	6 467	872	137 311	5 180	524
NC	27 245	312	349	607	138	166	574	467	24 065	567
WC	479 774	3 812	1 478	4 165	5 374	1 024	566	792	2 143	460 420

Table 2: *Commuter flow within and between the current nine provinces. The following abbreviations are used: EC = Eastern Cape, FS = Free State, GP = Gauteng, KZN = KwaZulu-Natal, LP = Limpopo, MP = Mpumalanga, NW = North West, NC = Northern Cape and WC = Western Cape. Total flow value within provinces: 2 403 210 (89.32%). Total flow value between provinces: 287 234 (10.68%).*

The map in Figure 10(a) shows "major" commuter flows, many crossing the current provincial boundary. The map in Figure 10(b) shows the intramax analysis results after a cleanup into eleven functional areas just before the Buskbuckridge area fuses with the South Kruger Park. The map in Figure 10(c) shows several larger commuter flows into / out of Bushbuckridge across the current provincial boundary.

The intramax results shown in Figure 10(d) allocate the whole of the Bushbuckridge functional area to the southern province and the proposed boundary follows the boundary of the building block instead of cutting through it. Of the 60 420 commuters in the area 3 771 (6.24%) currently cross the provincial boundary. This number would be reduced to 679 (1.12%) in the proposed provincial split.

The disputed regions of Groblersdal and Marble Hall (Shown in Figure 1) were allocated to Mpumalanga, but transferred to Limpopo province in December 2005 [8]. The intramax analysis indicates that these regions will actually fuse with the Gauteng region. Sasolburg will also fuse into the Gauteng region, and not with the Free State, where it is currently situated.

Kuruman, Postmasburg and Hartswater (currently in the Northern Cape) will be allocated to the North West region, but the boundaries of the North West region will move further south, and include more regions of the Northern Cape, even regions such as Upington, Prieska and De Aar. This is because of the accessibility to Kimberley, which will also be located in the North West region.

The Namaqualand (currently in the Northern Cape), Clanwilliam and Van Rhynsdorp (currently in the Western Cape) regions will be allocated to the Western Cape, and again, here, the N7 route ensures accessibility to the Cape Metropole.

The Pondoland, East Griqualand (currently in the Eastern Cape) and Umzimkulu (currently in the KwaZulu Natal) regions will fuse initially with the Eastern Cape region, but in a four and five province scenario, the Eastern Cape region will fuse with the KwaZulu-Natal region, leaving these disputed areas in the middle of the new province.

Pro- vince	Total flow	CT	QT	DU	NC	Kl	lB	JO	WB	PB
CT	481 244	461 782	3 930	4 583	546	1395	2 485	4 658	1 086	779
QT	184 433	2773	169 864	3 482	454	1 489	1 343	3 264	981	783
DU	365 583	2 694	7 529	343 668	2 300	848	1 121	4 955	1 701	767
NC	28 159	97	438	1 440	23 573	147	79	1 151	505	729
Kl	118 382	972	1 267	610	181	108 484	2 363	3 749	398	358
lB	134 115	942	968	360	170	1 591	119 701	9 086	333	964
JO	1 160 833	5 473	6 436	4 412	1 296	3 996	10 862	1 117 538	6 246	4574
WB	98 406	594	1 037	753	537	729	594	4 930	87 510	1 722
PB	119 194	499	1 343	879	171	603	1 148	11 021	1 974	101 556

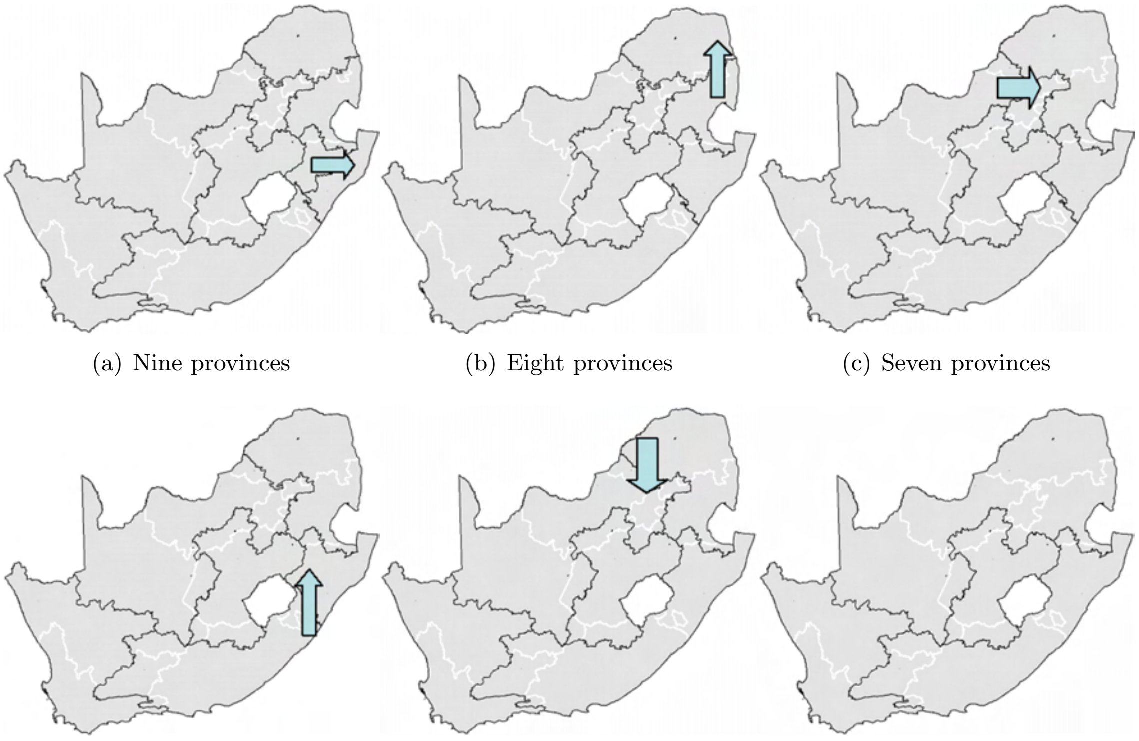
Table 3: *Commuter flow within and between the last nine clusters. The following abbreviations are used: CT = Cape Town (combination of Western Cape and Northern Cape), QT = Queenstown (mostly Eastern Cape), DU = Durban (mostly KwaZulu-Natal), NC = Newcastle (combination of KwaZulu-Natal and Mpumalanga), KL = Klerksdorp (combination of Free State and North West), RB = Rustenburg (combination of North West and Northern Cape), JO = Johannesburg (mostly Gauteng, with parts of surrounding provinces included), WB = Witbank (remainder of Mpumalanga) and PB = Pietersburg (remainder of Limpopo Province). Total flow value within provinces: 2 533 676 (94.18%). Total flow value between provinces: 156 673 (5.82%).*

The Brits and Garankuwa areas (currently in the Northwest Province) will also fuse with the Gauteng region.

Since 2001, numerous administrative problems and service delivery constraints associated with cross boundary municipalities prompted a special Presidential Coordinating Council to recommend the scrapping of this municipal category in 2001 [16]. The process of eliminating cross boundary municipalities was completed in December 2005 with the adoption by the National Assembly of the Constitution's Twelfth Amendment Act and the Cross-boundary Municipalities Laws Repeal and Related Matters Act, 2005 [3]. Both pieces of legislation effectively eliminated the reality of cross boundary municipalities and demarcated affected municipalities to one province or another. As a result, amongst others, aBushbuckridge, Khutsong, and Matatiele have been incorporated into the Mpumalanga, North West and Eastern Cape provinces respectively. The last two communities have violently resisted the new provincial locations.

Khutsong is part of Merafong municipality which was not indicated as a disputed area in Figure 1. This municipality was partly in the Northwest Province and partly in the Gauteng Province. It was allocated in 2005 to the Northwest Province. According to Figures 6 and 8 the area merges with the 'Seventh Province', which is mostly part of Gauteng.

Matatiele is part of Pondoland in Figure 1, and according to intramax analysis will be incorporated into the Queenstown area, which will, according to Figures 4 and Figure 8, merge with the 'Second Province', which will mostly be the Eastern Cape.



(a) Nine provinces (b) Eight provinces (c) Seven provinces

(d) Six provinces (e) Five provinces (f) Four provinces

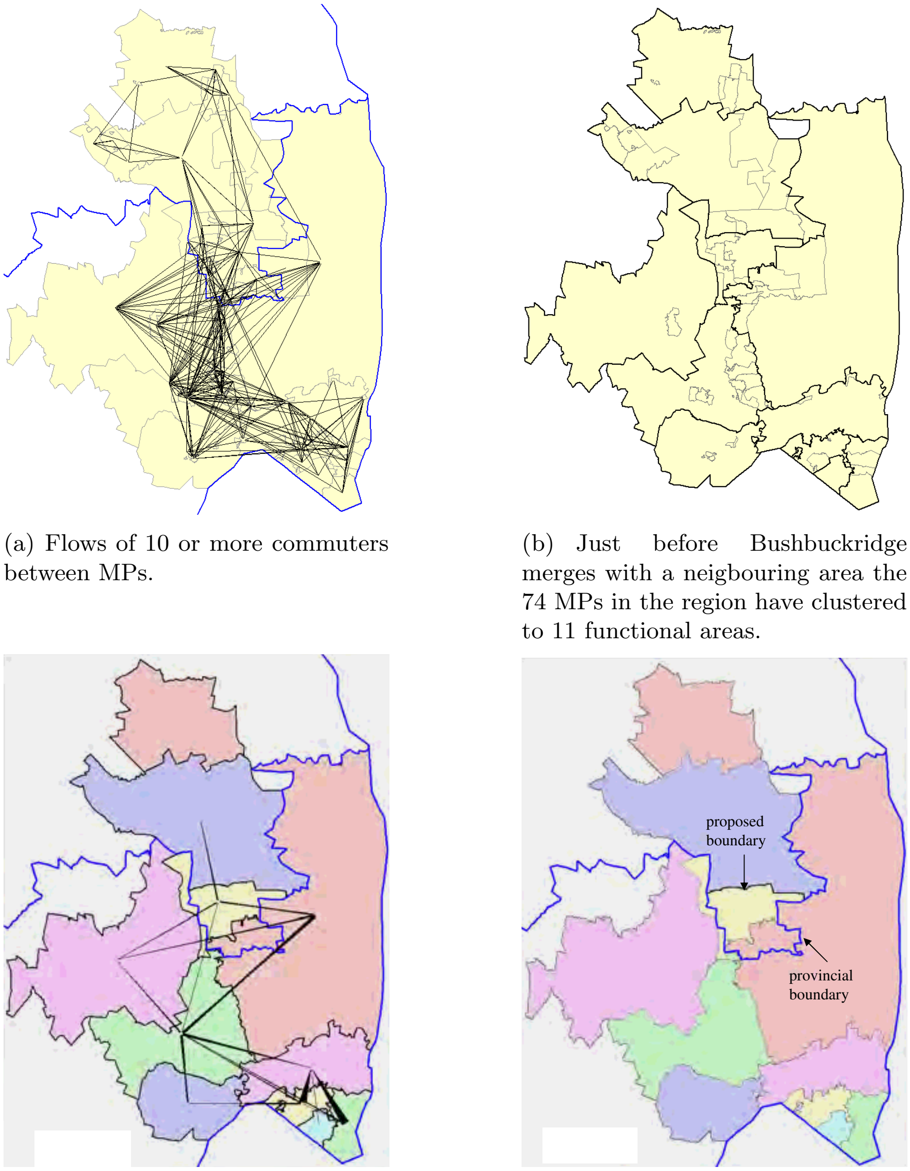
Figure 9: *Reduction from nine to four provinces using intramax analysis.*

6.4 Principal component/cluster analysis compared with intramax

According to Harmse *et al.* [11] the data matrix, consisting of 16 variables and 249 spatial units, was subjected to a principal component analysis. Three principal components had eigenvalues larger than 1, and together they were responsible for 77.9% of the variation in the original data set. The first principal component represented most of the socio-economic variables, and 10 out of 16 variables had scores of more than 0.75 on principal component I (PC I).

The calculated PC I scores for each of the 249 spatial units comprised a new data set. Cluster analysis was performed on this data set and the most effective grouping of the 249 spatial units resulted in 18 groups, which were then assigned to four regional types. Discriminant analysis was conducted to determine the effectiveness of the groupings.

Figure 11 shows the results of the demarcation of socio-economic development regions in the South African space economy. The 2001 development regions in South Africa ranged from the highly developed core region, through the upward-transitional and downward-transitional regions, to the special problem regions. According to Harmse *et al.* [11], the core region has the highest level of development and, in 2001, 69.2% of the country's total income was earned by people living in the core region. The core region housed 38% of the country's population on only 5.45% of the land area. The non-contiguous core region consists of the following regions (in descending order per province): City of Johannesburg



(a) Flows of 10 or more commuters between MPs.

(b) Just before Bushbuckridge merges with a neigbouring area the 74 MPs in the region have clustered to 11 functional areas.

(c) Flows of 100 or more commuters between the functional areas Bushbuckridge has stronger ties to the east (Kruger Park South).

(d) An intramax two way split of the area results in a new provincial boundary.

Figure 10: *In-depth analysis of Bushbuckridge as example of a disputed area.*

Metro in Gauteng, the Kruger Park in Mpumalanga, City of Tswane Metro in Gauteng, City of Cape Town Metro and Stellenbosch in the Western Cape, Ekurhuleni Metro in Gauteng, Gamagara in the Northern Cape, Midvaal in Gauteng, Mossel Bay in the Western Cape, Ethekwini in KwaZulu-Natal, Mogale City in Gauteng, Overstrand, Cape Agulhas, Saldanha Bay, George and Drakenstein in the Western Cape, Nelson Mandela Metro in the Eastern Cape, uMngeni in KwaZulu-Natal, Kungwini and Randfontein in Gauteng, Potchefstroom in the North West, Emfuleni in Gauteng, Knysna in the Western Cape, Nokeng tsa Taemane in Gauteng and the Swartland municipalities in the Western Cape. The Sol Plaatjie municipality in the Northern Cape would be the next on the list.

The levels of the socio-economic development in the 45 districts comprising the upward transitional region were not as high as in the core region, but although the region contains only 13.3% of the total population, it contributes a further 13.7% of the total income and 16.7% of the total number of people employed. These regions are usually adjacent to the core regions.

The 133 districts in the downward transitional region comprise the largest part of the system (61.1%). These are usually relatively poorly developed and unintegrated regions. These regions usually make a relatively small contribution towards the economy. In this case, the 33.3% of the total population only contributes 25.5% of total employed and 15.0% of total income.

The 45 districts in the special problem region have the lowest level of development in the space economy. These regions are characterised by very low levels of income and very low levels of employment. In this case, 15.4% of the population contributes 3.9% of total employment and 2.1% of total income. These regions pose a challenge to development [11].

	Special problem region	Downward transitional region	Upward transitional region	Core region	13 District municipalities excluded
Number of districts	45	133	45	26	13
% of total area	9.9	61.1	21.2	5.5	2.4
% of total population	15.4	33.3	13.3	38.0	0.01
% of total employed	3.9	25.5	16.7	53.9	0.02
% of total income	2.1	15.0	13.7	69.2	0.03

Table 4: *Contribution of each regional type to selected variables [11].*

The last four regions obtained by the intramax analysis were superimposed in Figure 11 on the development regions of the South African space economy, to establish visually some measure of validity of the intramax analysis.

Clearly, the Western Cape region consists of a strong core region and most of the surrounding regions are upward transitional. There are no special problem regions in this province, and one can come to the conclusion that the level of socio-economic development is high, *i.e.* this province can exist as a unit.

The combination of Gauteng, North West and Limpopo Province also has a strong core region in the Gauteng province, with smaller core regions in the mining areas of the Northern Cape. It has smaller upward transitional areas and larger downward transitional areas with a few problem regions. The strong core should be able to carry these problem regions economically. If this region is sub-divided and the Northern region (Limpopo province) is separated from this region, it might lead to a province (the northern part) with no core region, very little upward transition, large downward transition and problem regions, resulting in the region exhibiting low socio-economic development. This province might be dependent on the government for support.

In the central province, which is a combination of the Free State and North West provinces, the Potchefstroom region is reflected as a core region, but the Bloemfontein and Kroonstad regions are upward transitional regions. There are no problem regions in this province.

The last region is the combination of the major parts of the Eastern Cape and KwaZulu-Natal. Three core areas are identified: the Durban, Pietermaritzburg and Port Elizabeth regions. This region has large problem areas and downward transition regions, compared to upward transition regions. For this region, these three cores can combine their economic power in the combined province, but this province will experience a challenge to survive economically, based on these results.

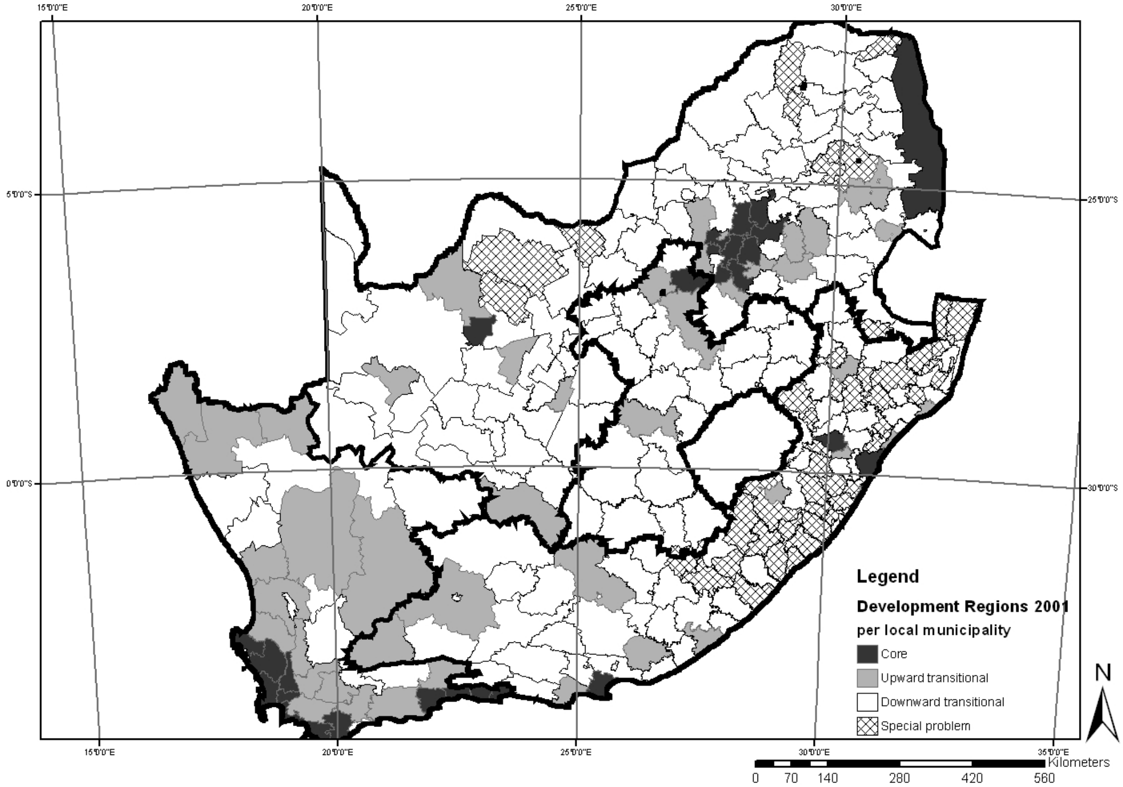


Figure 11: *Development regions in the South African space economy, with the four proposed provinces superimposed on the development regions.*

6.5 Socio-economic results, using 2001 Census data

The last five provinces are finally compared using certain socio-economic variables of data extracted from the Census 2001 Community Profile Databases [23]. The Limpopo Province region is kept separate from the Gauteng/North West region, because it was noticed that the combined region comprised 47.7% of the country, and might be too large to be considered a province. The results in Tables 5, 6 and 7 were calculated by weighing the data in the main places appropriately.

The weighted mean and median values for each region might differ substantially, because of the uneven distribution of the variables amongst the different main places within each region. For this reason, both statistics are reflected.

The statistics in Table 5 were weighted for the total number of people in each main place. The Western Cape region has the highest level of urbanisation (795 per 1 000 persons), followed by the Free State / North West and Gauteng / North West regions (approximately 630 per 1 000 persons). The Limpopo region has the lowest number (165 per 1 000 persons), the lowest number of informal persons, too, but the highest level of tribal / farm / small holding persons (785 per 1 000 persons) and the highest level of youthful dependency (children 0–14 years of age). The other regions have approximately the same level of informal persons (between 77 and 93 per 1 000 persons). The Western Cape has the lowest level of tribal / farm / small holding persons per 1 000 persons.

The statistics in Table 6 were weighted for the number of persons between 15 and 65 years of age. The Western Cape region has the highest level of agricultural and manufacturing activity and the highest employment level per 1 000 people aged 15 to 65 years, but the lowest level of mining activities. This region has the lowest number of people with education level of grade 7 and lower per 1 000 people aged 15 to 65 years.

The Free State / North West region has a high level of agricultural and mining activity, but low on the manufacturing level per 1 000 people aged 15 to 65 years of age. Employment levels are average.

Province (number of main places)	% of the total number of people in SA		Urban Person	Informal Person	Tribal + farm small holding	Youthful dependency
Western Cape		WMN:	795 [**A**]	78 [**A**]	98 [**D**]	273 [**D**]
region		WMD:	929	5	0	279
(350)	10.2	WQ:	733–968	0–112	0–12	246–300
Free State/North		WMN:	632 [**B**]	77 [**A**]	258 [**C**]	305 [**C**]
West region		WMD:	810	0	0	314
(267)	7.20	WQ:	0–963	0–171	0–843	282–338
Eastern Cape/		WMN:	333 [**C**]	92 [**A**]	554 [**B**]	355 [**B**]
KwaZulu–Natal		WMD:	0	0	964	351
region (1288)	34.9	WQ:	0–809	0–103	0–999	284–434
Gauteng/North		WMN:	635 [**B**]	93 [**A**]	239 [**C**]	269 [**D**]
West region		WMD:	846	34	0	257
(627)	31.0	WQ:	338–899	0–93	0–137	216–310
Limpopo		WMN:	165 [**D**]	30 [**B**]	785 [**A**]	379 [**A**]
region		WMD:	0	0	992	407
(496)	16.7	WQ:	0–0	0–0	957–999	341–424
South		WMN:	467	80	427	320
Africa		WMD:	659	0	12	250
(3028)	100	WQ:	0–893	0–92	0–994	397–317

Table 5: *Comparing the five intramax regions with respect to area of residence and youthful dependency per 1 000 persons. [A] to [D]: different symbols indicate which means of these variables (comparing different regions in descending order from [A] to [D]) are significantly different, Bonferroni multiple comparison test, $p < 0.05$. The following abbreviations are used: WMN for weighted mean per 1 000 persons, WMD for weighted median per 1 000 persons and WQ for weighted inter-quartile range per 1 000 persons.*

The Eastern Cape / KwaZulu-Natal region features significantly less agricultural activity, less mining activities, the lowest level of employment, and comparatively a high level of people with an education level of grade 7 and lower. However, some manufacturing activities take place in this region.

The Gauteng / North West region has a low level of agricultural activity, a high level of mining activity and a relatively high level of manufacturing activity and employment compared to the other regions.

The Limpopo region has a high level of agricultural activity, a higher than average level of mining activity, but a low level of manufacturing activity and a low level of employment. If combined with the Gauteng / North West region, the two regions can augment each other.

Province (number of main places)		Industry agriculture	Industry mining	Industry manufacturing	Employed	Grade 7 and less
Western	WMN:	67 [**A**]	3.4 [**C**]	67 [**A**]	484 [**A**]	272 [**C**]
Cape region	WMD:	9	1.0	75	489	254
(348)	WQ:	7–30	0–2	34–91	9 425–526	178–312
Free State/	WMN:	55 [**A**]	34 [**A**]	24 [**D**]	337 [**C**]	420 [**A**]
North West	WMD:	8	2	21	290	428
region (267)	WQ:	5–18	1–18	10–30	0 226–483	360–502
Eastern Cape/	WMN:	23 [**B**]	1 [**C**]	38 [**C**]	251 [**D**]	418 [**A**]
KwaZulu–Natal	WMD:	4	1	27	228	424
region (1287)	WQ:	3–9	0–2	4–60	93–378	269–570
Gauteng/	WMN:	22 [**B**]	23 [**AB**]	51 [**B**]	404 [**B**]	303 [**B**]
North West	WMD:	6	3	50	352	283
region (627)	WQ:	4–10	2–7	30–66	287–541	169–403
Limpopo	WMN:	51 [**A**]	15 [**B**]	24 [**D**]	270 [**D**]	445 [**A**]
region	WMD:	11	2	14	214	462
(496)	WQ:	6–21	1–9	6–31	123–354	394–521
South	WMN:	34	13	43	337	367
Africa	WMD:	7	2	38	322	360
(3025)	WQ:	4–13	1–4	11–63	189–497	228–500

Table 6: *Comparing the five intramax regions with respect to industries, employment and education per 1 000 of persons aged 15–65. [A] to [D]: different symbols indicate which means of these variables (comparing different regions in descending order from [A] to [D]) are significantly different, Bonferroni multiple comparison test, $p < 0.05$. The following abbreviations are used: WMN for weighted mean per 1 000 persons (aged 15–65), WMD for weighted median per 1 000 persons (aged 15–65) and WQ for weighted inter-quartile range per 1 000 persons (aged 15–65).*

The statistics in Table 7 were weighted for the total number of households. The Western Cape region has the highest level of annual household income, and the highest mean number of households living in brick houses, equipped with electricity and piped water inside the house per 1 000 households. The Free State / North West region has the highest level of informal houses per 1 000 households. The Eastern Cape / KwaZulu-Natal region has the lowest level of brick housing, and the lowest level of electricity and piped water in the house per 1 000 households. The Limpopo region has the lowest level of annual household income.

Province (number of main places)		Ann hh income (R)	Brick house	Informal house	Electricity	Piped water in house
Western Cape region (344)	WMN:	75 615 [**A**]	631 [**A**]	156 [**B**]	878 [**A**]	851 [**A**]
	WMD:	63 850	606	53	929	931
	WQ:	408–116	558–773	37–217	814–984	802–948
Free State/ North West region (266)	WMN:	30 853 [**C**]	572 [**B**]	232 [**A**]	747 [**B**]	699 [**C**]
	WMD:	18 610	594	231	748	663
	WQ:	13 926–28 160	465–669	43–356	638–879	585–914
Eastern Cape/ KwaZulu–Natal region (1282)	WMN:	34 123 [**C**]	413 [**C**]	106 [**C**]	573 [**D**]	458 [**D**]
	WMD:	19 896	429	41	645	516
	WQ:	13 955–36 424	204–568	13–131	5 299–859	46–820
Gauteng/ North West region (626)	WMN:	60 780 [**B**]	555 [**B**]	211 [**A**]	787 [**B**]	754 [**B**]
	WMD:	29 916	558	147	853	856
	WQ:	20 738–101 675	437–676	51–327	698–916	684–911
Limpopo region (491)	WMN:	26 135 [**C**]	624 [**A**]	89 [**C**]	629 [**C**]	467 [**D**]
	WMD:	17 642	673	42	649	415
	WQ:	13 787–22 741	479–777	17–89	510–773	249–689
South Africa (3009)	WMN:	46 361	530	155	702	623
	WMD:	23 954	558	82	763	731
	WQ:	16 190–56 298	411–682	23–243	581–903	384–886

Table 7: *Comparing the five intramax regions with regards to household income, type of housing and services per 1 000 of households. [A] to [D]: different symbols indicate which means of these variables (comparing different regions in descending order from [A] to [D]) are significantly different, Bonferroni multiple comparison test, $p < 0.05$. The following abbreviations are used: WMN for weighted mean per 1 000 households, WMD for weighted median per 1 000 households and WQ for weighted inter-quartile range per 1 000 households.*

7 Conclusion

Based on journey-to-work flows extracted from Census 2001 data, the intramax procedure was used to aggregate the 3 109 (with some minor modifications) main places in South Africa into four or possibly five provinces. The provinces thus identified are:

- A 'Western Cape' province, which includes most of the current Western Cape and some regions of the previous Northern Cape region;
- a coastal province which is the amalgamation of most regions in the Eastern Cape and KwaZulu-Natal provinces;
- a central province consisting of most of the Free State and a small part of the North West province;
- a combination of the Gauteng province, the remainder of the Northern Cape and North West province; and
- a combination of the Limpopo province and the Northern parts of Mpumalanga.

It is interesting to note that provinces with relatively low commuting figures, as reflected in Table 3, also have low employment figures, as reflected in Table 6.

Disputed areas were highlighted and intramax solutions were provided for these disputes. These solutions are based on economic activities of people living in the areas, which might be of interest to policy makers in future.

The results of a recent paper on the demarcation of the socio-economic development regions in the South African space economy were discussed with the purpose of applying them to the newly formed provinces. It is clear that the Western Cape region, with a strong core and mostly upward transitional regions also reflects high socio-development, according to Tables 5–7. The Eastern Cape / KwaZulu-Natal region reflects three minor core regions and large downward transitional and special problem regions. This is reflected in the fact that the socio-economic variables in Tables 5–7 clearly indicate that few industrial activities take place (apart from manufacturing). This region also suffers from low education, employment and income levels and poorly developed services in comparison with the other provinces. The development of this region poses a challenge to the government, but it has a true potential to improve, especially given its manufacturing base and access to harbours.

The central region, comprising of a combination of most of the Free State and parts of the North West province is almost a perfect match. The agricultural industry in the Free State combined with some mining activities from the regions in the North West province ensures that this region can be economically viable. Additionally, the fact that there are no problem regions according to the South African space economy model, indicates that this region can be economically independent.

The last two regions — Gauteng, parts of the Northern Cape, the North West province and a combination of Limpopo and Mpumalanga — may be too large for one province, but the northern region has no core (the Kruger Park should be seen independently), large problem areas and downward transitional regions, which correspond to low income, poor education and employment levels compared to the other regions with large rural areas. It has a good agricultural industry in place, as well as some mining activities, which might be further explored. If this province is combined with the highly developed Gauteng region, which lacks a significant agricultural industry, it can be a powerful region.

Based on commuting flows, the intramax method is a useful tool for demarcating regions using daily activity systems. From a management point of view provincial/adminstrative boundaries should take these daily activity systems into account in some form or another.

References

[1] Brown LA & Holmes J, 1971, *The delimitation of functional regions, nodal regions, and hierarchies by functional distance approaches*, Journal of Regional Science, **11(1)**, pp. 57–72.

[2] Brown PJB & Pitfield DE , 1990, *An intramax derivation of commodity market areas from freight flow data*, Transportation Planning and Technology, **15**, pp. 59–81.

[3] Cross-boundary Municipalities Laws Repeal and Related Matters Act of 2005, South Africa, [Statute], Government Gazette, **486**, Pretoria, pp. 1–20.

[4] Feldman O, Simmonds D, Troll N & Tsang F, 2005, *Creation of a system of functional areas for England and Wales and for Scotland.* Proceedings of the European Transport Conference, [Online], [cited 2008, August 11], Available from: `http://www.mvaconsultancy.com/papers/Creation%20of%20a%20system%20of%20functional%20areas%20for%20England%20and%20W%85.pdf`

[5] Fischer MM, Essletzbichler J, Gassler H & Trichtl G, 1993, *Telephone communication patterns in Austria: A comparison of the IPFP-based graph-theoretic and the intramax approaches*, Geographical Analysis, **25(3)**, pp. 224–233.

[6] Floor J & De Jong T, 1981, *Development and testing of a residential location model*, PhD Dissertation, Utrecht University, Utrecht.

[7] Goodman L, 1963, *Statistical methods for the preliminary analysis of transaction flows*, Econometrica, **31**, pp. 197–208.

[8] Greater Sekhukhune District Municipality, 2008, *Integrated development plan: 2007/2008 review*, [Online], [cited 2008, August 11], Avialable from: `http://www.sekhukhune.gov.za/2007-08%20FINAL%20IDP.pdf`

[9] Griggs RA, 1998, *The security costs of party-political boundary demarcations: The case of South Africa*, African Security Review, **7(2)**, pp. 22–32.

[10] Harmse AC, 2007, *Socio-economic development regions in the South African space economy*, The South African Geographical Journal, **89**, pp. 83–88.

[11] Harmse AC, Blaauw PF & Schenck CJ, 2008, *Day labourers, unemployment and socio-economic development in South Africa*, Working Paper Number 69, Department of Geography, University of South Africa, Pretoria.

[12] Hirst MA, 1977, *Hierarchical aggregation procedures for interaction data: A comment*, Environment and Planning A, **9**, pp. 99–103.

[13] Hollingworth TH, 1971, *Gross migration flows as a basis for regional definition: An experiment with Scottish data*, Proceedings of the IUSSP Conference held in London in 1969, **4**, pp. 2755–2765.

[14] Masser I & Brown PJB, 1975, *Hierarchical aggregation procedures for interaction data*, Environment and Planning A, **7**, pp. 509–523.

[15] Masser I & Scheurwater J, 1980, *Functional regionalization of spatial interaction data: An evaluation of some suggested strategies*, Environment and Planning A, **12**, pp. 1357–1382.

[16] Mavungu EM, 2007, *Explaining boundary disputes in post-apartheid South Africa: Bushbuckridge, Matatiele and Khutsong*, University of the Witwatersrand, Johannesburg, Newsroom, 23 October 2007, [Online], [cited 2008, August 11], Available from: `http://web.wits.ac.za/NewsRoom/NewsItems/khutsong.htm`

[17] Mitchell W, Bill A & Watts M, 2007, *Identifying functional regions in Australia using hierarchical aggregation techniques*, Working Paper No. 07-06, Centre of Full Employment and Equity, The University of Newcastle, Callaghan (Australia).

[18] Ngalwa S, 2007, *Review of number of provinces under way*, The Star, pp. 6, 20 June 2007.

[19] Ramutsindela MF & Simon D, 1999, *The politics of territory and place in post-apartheid South Africa: The disputed area of Bushbuckridge*, Journal of Southern African Studies, **25(3)**, pp. 479–498.

[20] Slater PB, 1975, *A hierarchical regionalization of Japanese prefectures using 1972 interprefectural migration flows*, Regional Studies, **10**, pp. 123–132.

[21] Smith A, 2007, *Four provinces mooted for SA*, Die Burger, 3 May 2007, [Online], [cited 2008, August 11], Available from: `http://www.news24.com/News24/South_Africa/Politics/0,,2-7-12_2107703,00.html`

[22] Statistics South Africa, 2001, Census 2001, Subdatabase compiled from community profile databases by Statistics South Africa, Pretoria.

[23] Statistics South Africa, 2001, Census 2001, *Community profile databases*, Available from: `http://www.statssa.gov.za/census01/html/C2001CommProfile.asp`.

[24] Tyree A, 1973, *Mobility ratios and association in mobility tables*, Population Studies, **27**, pp. 577–588.

[25] Van der Zwan J, Van der Wel R, Ritsema van Eck J, de Jong T & Floor H, 2003, *FLOWMAP 7. Manual,* Faculty of Geographical Sciences, Utrecht University, Utrecht.

[26] Ward JH, 1963, *Hierarchical grouping to optimize an objective function*, Journal of the American Statistical Association, **58(301)**, pp. 236–244.

Volume 24 (2), pp. 157–183
http://www.orssa.org.za

ORiON
ISSN 0529-191-X
©2008

A survey and comparison of heuristics for the 2D oriented on-line strip packing problem

N Ntene[*] JH van Vuuren[†]

Received: 11 August 2008; Revised: 22 October 2008; Accepted: 27 October 2008

Abstract

The two dimensional oriented *on-line* strip packing problem requires items to be packed, one at a time, into a strip of fixed width and infinite height so as to minimise the total height of the packing. The items may neither be rotated nor overlap. In this paper, ten heuristics from the literature are considered for the special case where the items are rectangles. Six modifications to some of these heuristics are proposed, along with two entirely new shelf algorithms. The performances and efficiencies of all the algorithms are compared in terms of the total packing height achieved and computation time required in each case, when applied to 542 benchmark data sets documented in the literature.

Key words: Heuristic packing, on-line packing, shelf algorithm, strip packing.

1 Introduction

The two dimensional *strip packing problem* involves packing a list of items (in this case, rectangles) into a bin (referred to as a strip) of fixed width and infinite height. The objective is to minimise the total packing height in the strip for which rectangles do not overlap. Each rectangle L_i is specified by the pair of dimensions $(h(L_i), w(L_i))$ referring to its height and width respectively. Ntene and Van Vuuren [22] conducted a survey on heuristics for solving *offline* strip packing problems approximately. These are problems where the entire set of rectangles to be packed is known in advance. There are, however, applications where the entire set of rectangles to be packed is not known in advance and problems of this nature are referred to as *on-line* packing problems. Applications of this class of problems include warehouse storage [2, 3], VLSI design [14] and scheduling with a shared resource [3, 6, 20].

In an *on-line* environment, rectangles are packed one at a time; rectangle L_{i+1} only becomes available once rectangle L_i has been packed [2, 13, 14, 19, 20]. Another condition

[*] Department of Logistics, University of Stellenbosch, Private Bag X1, Matieland, 7602, Republic of South Africa.

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

for a system to be fully *on-line* is that once a rectangle has been packed it may not be moved at a later stage of the packing. The challenge in *on-line* packing problems is due to the potential volatility of rectangle heights that have yet to be packed [19].

The main objective in this paper is to examine and compare the time efficiencies and performances of a number of existing heuristics for *on-line* packing problems in the literature, and to propose some improvements or suggest altogether new algorithmic approaches. The paper is organised as follows. In §2 the mechanisms behind a number of existing *level algorithms* for *on-line* packing problems are reviewed and illustrated by means of a numerical example. In §3 a number of *shelf algorithms* from the literature are briefly described and illustrated by means of an example. A number of algorithms for solving *on-line* packing problems with additional constraints (approximately) are discussed and illustrated by means of an example in §4. Then a number of possible modifications to some of these procedures considered in §2–4 are presented in §5. Two entirely new shelf algorithms are presented in §6 and finally all the algorithms are tested on a large set of existing benchmark problem instances so that their performances and time-efficiencies may be compared statistically in §7.

To illustrate the packing patterns produced by the various algorithms mentioned above, all algorithms are applied to an example instance requiring 10 rectangles to be packed into a strip of width 15 units. This is the same example instance used by Ortmann *et al.* [23] to facilitate comparisons for offline packing algorithms. The rectangle dimensions (height, width) for the example instance are shown in Table 1.

L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}
$(14,5)$	$(5,4)$	$(4,9)$	$(15,1)$	$(6,11)$	$(6,2)$	$(4,6)$	$(2,5)$	$(6,10)$	$(1,7)$

Table 1: *Dimensions $(h(L_i), w(L_i))$ of rectangles $L_1, \ldots, L_{10}$ used as example instance in §2–6.*

2 Level Algorithms

The algorithms considered in this section are a slight variation on the algorithms investigated in [22], namely the *next fit decreasing height* (NFDH) [10], the *first fit decreasing height* (FFDH) [10] and the *best fit decreasing height* (BFDH) [11] algorithms. Since we are dealing with *on-line* packing problems, we do away with the pre-ordering condition in each of these original algorithms.

In the *next fit level* (NFL) algorithm [11], rectangles are packed (one at a time and in the order given) on the current level, left justified. The first level corresponds with the bottom of the strip. If there is insufficient horizontal space on the current level to pack the next rectangle, a horizontal line is drawn across the upper edge of the tallest rectangle on the current level so as to create a new level above the current level. All levels below the current level are never revisited.

In the *first fit level* (FFL) algorithm [11], rectangles are packed (one by one in the order given) on the lowest level into which they fit both height-wise and width-wise; if a rectangle does not fit into any existing level, then a new level is created exactly as in the NFL algorithm and the rectangle in question is packed on that level.

The *best fit level* (BFL) algorithm [11] is similar to the FFL algorithm, except that each rectangle is placed on the lowest level (into which it fits both height-wise and width-wise) with minimum residual horizontal space (the space between the right-most edge of the last rectangle packed on a level and the right-hand boundary of the strip).

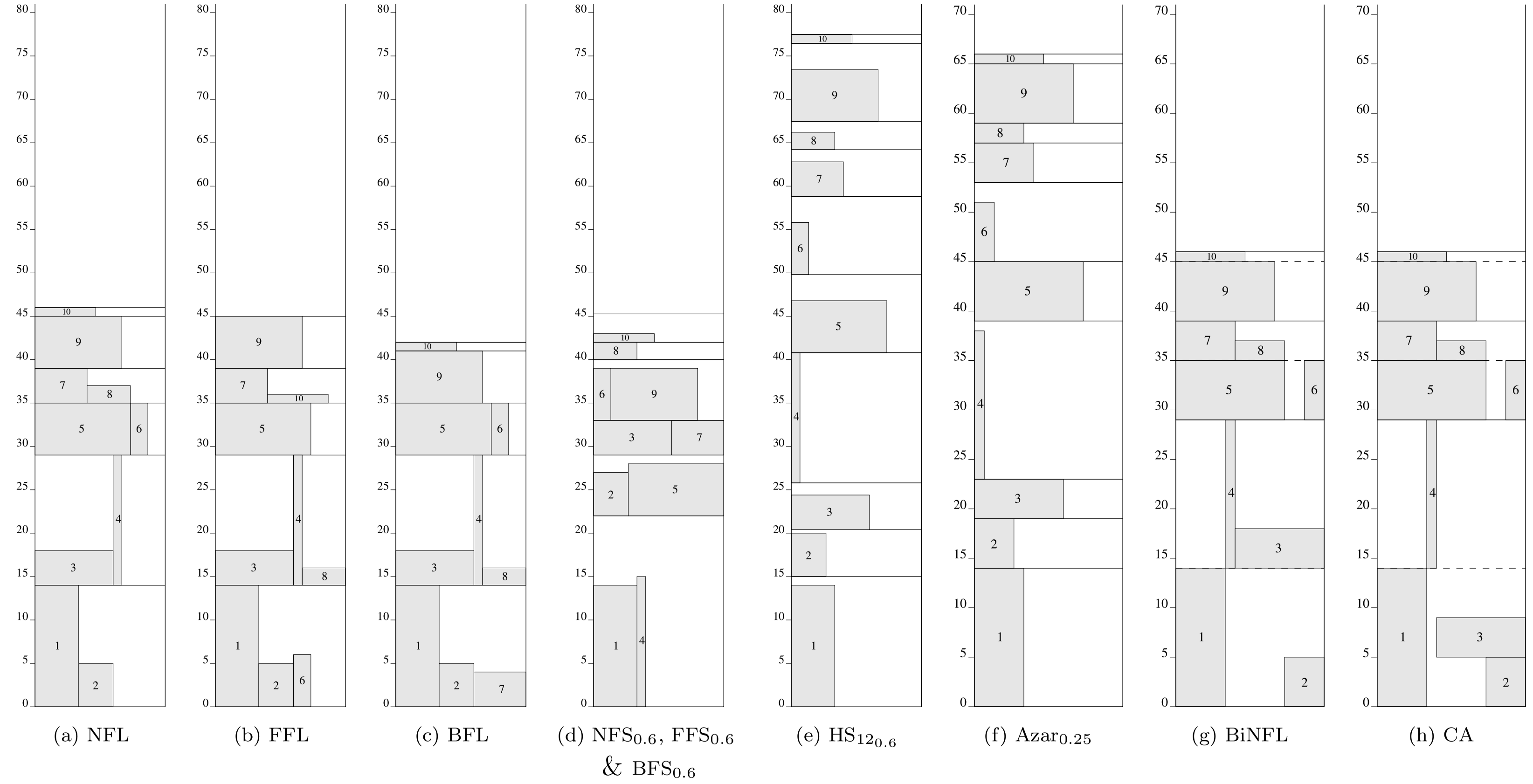
For our example instance in Table 1, total packing heights of 46, 45 and 42 units are obtained by the NFL, FFL and BFL algorithms respectively, as shown in Figure 1(a)–(c).

3 Shelf algorithms

In shelf algorithms, rectangles are also packed on horizontal planes (referred to as shelves) of fixed height as in the case of level algorithms. However, this class of algorithms differs from the class of level algorithms in that additional space (called *free space*) is intentionally left between the top-most edge of the tallest rectangle on a shelf and the position of the next shelf so as to accommodate (to some degree) potential volatility in the heights of rectangles yet to be packed. However, in a level algorithm, the position of a level coincides with the top-most edge of the tallest rectangle on the previous level. The name *shelf algorithm* is derived from the situation where books are packed in a stack of bookshelves [2].

Shelf algorithms were first designed by Baker *et al.* [2] who modified two existing *offline* heuristics, namely the NFDH and FFDH algorithms [10]. The resulting two shelf algorithms are referred to as the *next fit shelf* (NFS$_r$) and *first fit shelf* (FFS$_r$) algorithms, where $0 < r < 1$ is a parameter, and these algorithms are described in §3.1. In these shelf algorithms, the objective is to pack rectangles of similar heights $r^{k+1} < h(L_i) \leq r^k$ on a single shelf of fixed height r^k (for some integer k). The parameter r is a measure of how much free space is allowed on each shelf to accommodate variations in the heights of rectangles to come. A small value of r (approximately equal to zero) results in large-sized shelves — hence allowing for rectangles with large variations in height to be packed on the same level. On the other hand, a large value of r (approximately equal to 1) allows rectangles of almost similar heights to be packed on one level due to the small shelf heights created [25]. For the shelf algorithms applied to our example instance in Table 1, a value of $r = 0.6$ was selected for illustrative purposes.

Coffman [12] modified the BFDH algorithm [11] to arrive at the so-called *best fit shelf* (BFS$_r$) algorithm, also described in §3.1, which differs from the NFS$_r$ and FFS$_r$ algorithms in a manner analogous to the difference between the NFL, FFL and BFL level algorithms. As Csirik and Woeginger [14] mention, shelf algorithms are based on one dimensional bin packing procedures: after determining an appropriate shelf on which a rectangle may be packed, so that it fits height-wise, the problem then becomes the one dimensional bin packing problem of determining amongst which of the shelves of appropriate height the rectangle should be packed (during this last stage only one dimension, namely width, is of interest, because it has been determined that height-wise the rectangle will fit). It is on this basis that another shelf algorithm, known as the *harmonic shelf* (HS$_{M_r}$) algorithm is reviewed in §3.2.



(a) NFL

(b) FFL

(c) BFL

(d) $\mathrm{NFS}_{0.6}$, $\mathrm{FFS}_{0.6}$ & $\mathrm{BFS}_{0.6}$

(e) $\mathrm{HS}12_{0.6}$

(f) $\mathrm{Azar}_{0.25}$

(g) BiNFL

(h) CA

Figure 1: *Packings produced by the algorithms described in §2–4 for the example instance of the strip packing problem in Table 1. Rectangle L_i is denoted by i in the figure, for all $i \in \{1, \ldots, 10\}$.*

3.1 The Next Fit Shelf, First Fit Shelf and Best Fit Shelf algorithms

The *next fit shelf* (NFS$_r$) algorithm [2] with parameter $0 < r < 1$ is a natural modification of the NFDH algorithm [10], the difference being that the rectangles are not sorted in the NFS$_r$ algorithm; they are merely packed in the order given. In this algorithm, a value of r is initially selected for the entire packing. Before packing each rectangle, the smallest integer k is computed for which $r^{k+1} < h(L_i) \leq r^k$; here r^k is referred to as the *appropriate height* of the shelf to pack rectangle L_i. A rectangle is packed on the highest shelf of appropriate height. If a shelf of appropriate height for rectangle L_i does not exist, a new shelf of appropriate height is created above the top-most shelf and rectangle L_i is packed there, left justified. If a shelf of appropriate height exists, but there is insufficient space to accommodate the rectangle, this shelf is closed off and a new shelf of the same (appropriate) height is created above the top-most level.

The *first fit shelf* (FFS$_r$) algorithm [2] with parameter $0 < r < 1$ is a modification of the FFDH algorithm [10] and it is similar to the NFS$_r$ algorithm, except that a rectangle is placed left justified on the *lowest* shelf of appropriate height instead of on the highest shelf of appropriate height.

The *best fit shelf* (BFS$_r$) algorithm [12] with parameter $0 < r < 1$ is a modification of the *best fit decreasing height* (BFDH) algorithm [11]. The difference between the FFS$_r$ and BFS$_r$ algorithms is that once the parameter r has been selected and different values of k determined, the latter procedure packs a rectangle on the lowest shelf of appropriate height *with minimum residual horizontal space.*

As shown in Figure 1(d), a total packing height of 45.27 units is obtained via all three of the NFS$_{0.6}$, FFS$_{0.6}$ and BFS$_{0.6}$ algorithms for our example instance in Table 1.

3.2 The Harmonic Shelf algorithm

Csirik and Woeginger [14] combined a one dimensional bin packing algorithm, called the *harmonic*$_M$ algorithm and proposed by Lee and Lee [19], with the principles of shelf algorithms. The *harmonic*$_M$ algorithm is used to partition the interval (0,1] non-uniformly into M intervals $I_1, \ldots, I_M$, where $I_p = (1/(p+1), 1/p]$, $1 \leq p < M$ and $I_M = (0, 1/M]$. A reasonable value of M is considered to be in the range $3 \leq M \leq 12$. This harmonic partition allows a rectangle to be classified according to the interval into which it fits width-wise.

The *harmonic shelf* (HS$_{M_r}$) algorithm does not only aim to pack rectangles of similar heights on the same shelf; over and above this objective the rectangles should also have similar widths. Before rectangle L_i is packed, two decisions have to be made. The first decision is to determine the appropriate shelves onto which a rectangle may be packed in terms of its height by selecting a value for r and computing a value of k for which $r^{k+1} < h(L_i) \leq r^k$. The second decision is to determine the interval I_p into which the rectangle belongs width-wise, by computing the value of p for which $1/(p+1) < w(L_i) < 1/p$. Only rectangles belonging to I_p, with r^k as the appropriate height may be packed onto such a shelf. If no shelf of appropriate height exists or if there is insufficient horizontal space on all shelves of appropriate height, then a new shelf of appropriate height is created above the current top-most shelf. In our example instance in Table 1, a total packing height of

77.60 units is obtained via the $\text{HS}_{12_{0.6}}$ algorithm, as depicted in Figure 1(e). Values of $M = 12$ and $r = 0.6$ were used in this example for illustration purposes.

4 Packings observing the tetris constraint

In all the algorithms reviewed thus far, it was assumed that a rectangle may be packed onto any shelf inside the strip as long as it fits. However, there are applications, such as packing boxes from the back of a delivery vehicle, where rectangles have to be transferred through all succeeding levels before being packed (for example, in order to reach the lower levels of the strip which model the front of the vehicle). This constraint is also found in the game *Tetris* where rectangles drop from the top of the strip to reach lower levels and the player has to avoid being blocked by rectangles already packed in other levels. Three existing algorithms in the literature, taking this additional constraint into consideration, are reviewed in this section.

4.1 The Azar$_Y$ algorithm

This algorithm is from a paper by Azar and Epstein [1]. In the Azar$_Y$ algorithm, the rectangle widths are assumed to be in the range (0,1] and the strip has width 1, without loss of generality. However, there is no restriction on the rectangle heights. The Azar$_Y$ algorithm partitions the strip into horizontal levels by means of a real threshold constant $0 < Y < \frac{1}{2}$. Rectangles of particular heights $(2^{j-1} < h(L_i) \leq 2^j)$ and widths $(2^{-x-1} < w(L_i) \leq 2^{-x})$ are packed on the same level, referred to as an (x, j) level (where $j \in \mathbb{Z}$ and $x \in \mathbb{N}$).

A rectangle whose width is at least Y is referred to as a *buffer*. When the next rectangle to be packed arrives, it is classified either as a *buffer* or *non-buffer*, depending on its width. If it is a *buffer*, a new level, whose height is equal to the height of the *buffer*, is created above the top-most level and the rectangle is packed there, left justified. This means that *buffers* are packed on their own within levels. If the rectangle is a *non-buffer*, it is classified as an (x, j) rectangle, for some $j \in \mathbb{Z}$ and some $x \in \mathbb{N}$. The first *non-buffer* rectangle packed on a level determines the height of the level as 2^j and this level becomes an (x, j) level. If a rectangle fits on an (x, j) level and it can reach such a level without being blocked by any of the *buffers*, then it is placed on that level. However, if no such level exists, if the rectangle does not fit on a particular (x, j) level or if the rectangle is blocked, then a new level of height 2^j is created above the top-most level. For our example instance in Table 1, a total packing height of 66 units is obtained via the Azar$_{0.25}$ algorithm, as depicted in Figure 1(f), where the value of $Y = 0.25$ was chosen for illustrative purposes.

4.2 The Bi-level Next Fit algorithm

The *Bi-level next fit* (BiNFL) algorithm [9] is a modification of the NFL algorithm described in §2. As the name suggests, the algorithm packs two levels at a time, referred to as the *lower* and *upper* levels. The height of the *lower* level is determined by the height of the tallest rectangle packed on it.

The first rectangle L_i to be packed on a bi-level is placed on the *lower* level, left justified. If the next rectangle L_{i+1} to be packed fits on the *lower* level, it is placed there, right justified. All other rectangles that follow and fit on the *lower* level are placed there, right justified, next to the previous rectangle packed. If there is not enough room for a rectangle to be packed on the *lower* level, packing proceeds on the *upper* level. A horizontal line is drawn along the top-most edge of the tallest rectangle on the *lower* level and this becomes the lower boundary of the *upper* level.

If, on the *upper* level, rectangle L_{i+1} is the first rectangle to be packed (because it failed to fit on the *lower* level), it is packed left justified on top of L_i since it is the only rectangle on the *lower* level. Subsequent rectangles are packed left justified on this level provided there is sufficient space (see Figure 2(a)). If, on the other hand, L_{i+2} is the first rectangle to be packed on the *upper* level, it is packed above the shorter of L_i and L_{i+1} (because these are the only two rectangles on the *lower* level), justified against the same strip boundary as the shorter of rectangles L_i and L_{i+1}; this scenario is depicted in Figures 2(b) and (c). If there are more than two rectangles on the *lower* level, the first rectangle packed on the *upper* level is packed above the shorter of the first left justified or the first of the right justified rectangles on the *lower* level. If a rectangle does not fit on the *upper* level, a new bi-level is created above the top-most level and similar steps are carried out as defined for the *lower* and *upper* levels until all rectangles are packed.

A total packing height of 46 units is obtained for our example instance in Table 1, as shown in Figure 1(g), with the *lower* and *upper* levels within each bi-level separated by dashed lines.

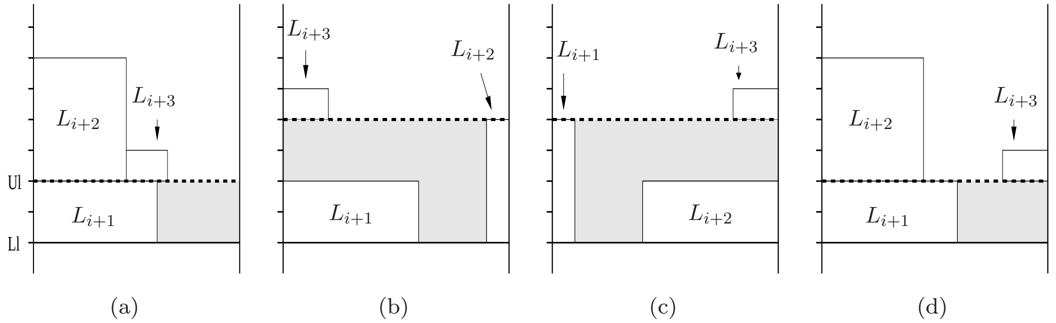


Figure 2: *(a)–(c) Examples of patterns resulting from a BiNFL packing. (d) In the CA algorithm the second rectangle packed on the* upper *level is right justified provided only one rectangle is packed on the* lower *level. Ul and Ll represent the lower boundaries of the* upper *and* lower *levels respectively.*

4.3 The Compression algorithm

The *compression algorithm* (CA) [9] is an extension of the BiNFL algorithm. It exploits certain patterns (when only one or two rectangles are packed on the *lower* level) that result from a BiNFL packing. In the CA algorithm, packing on the *lower* level proceeds in a manner similar to a BiNFL packing. However, if rectangle L_i $(i \geq 3)$ is the first rectangle to be packed on the *upper* level, it is justified according to the shorter of the first left justified or first right justified rectangles on the *lower* level, and it is slid down onto the *lower* level provided there is sufficient space (see Figures 2(b) and (c)) — this process is called *compression.* If rectangle L_i ($i \geq 3$) is the second rectangle to be packed (*i.e.* if there is one rectangle on each level, each of them left justified), it is right justified and if there is sufficient room on the *lower* level, this rectangle is compressed down onto the *lower* level (see Figure 2(d)). Subsequent rectangles that fit on the *lower* level may also be shifted next to previously compressed rectangles. Packing continues on the *upper* level as in the BiNFL algorithm for rectangles that may not be compressed down. A rectangle that fails to fit on the *upper* level is placed in a new bi-level that is created above the top-most level and previous bi-levels are never revisited. In our example instance in Table 1, a total packing height of 46 units is obtained via the CA algorithm, as shown in Figure 1(h).

5 Proposed Modifications

A number of modifications to some of the algorithms reviewed in §2–4 are proposed in this section.

5.1 The Modified Next Fit, First Fit and Best Fit Level algorithms

As the name suggests, the *modified next fit level* (MNFL) algorithm is a newly proposed variation on the NFL algorithm described in §2. In the MNFL algorithm, the first rectangle packed on a level determines the height of that level. If a rectangle is encountered that does not fit onto the current level, that level is closed off in both these algorithms and a new current level is created above it. The NFL algorithm is expected to perform poorly if the rectangles are presented in an order in which they tend to increase in height. However, if the rectangles are presented in an order in which they tend to decrease in height, then the algorithm is expected to perform well. The MNFL algorithm differs from the NFL algorithm in that in the latter procedure, level heights are determined by the tallest rectangle packed on a level, while in the former procedure, level heights are determined by the first rectangle packed on the level. For our example instance in Table 1, a total packing height of 44 units is obtained via the MNFL algorithm, as shown in Figure 5(a).

In the *modified first fit level* (MFFL) algorithm, the height of each level corresponds to the height of the first rectangle packed on that level. The MFFL and FFL algorithms differ in a manner analogous to the difference between the MNFL and NFL algorithms. A total packing height of 41 units is obtained via the MFFL algorithm for our example instance in Table 1, as illustrated in Figure 5(b).

The *modified best fit level* (MBFL) algorithm is similar to the BFL algorithm, except that in the BFL algorithm the height of a level is determined by the height of the tallest rectangle packed on the level, while in the MBFL algorithm the height of a level is determined by the first rectangle packed on the level. A total packing height of 40 units is obtained via the MBFL algorithm for our example instance in Table 1, as illustrated in Figure 5(c).

5.2 The Compression Part Fit algorithm

Downey [15] mentions that the CA algorithm (described in §4.3) is far from optimal, because it only takes a few patterns into consideration (where it may be possible to compress rectangles from the *upper* to the *lower* level). The *compression part fit* (CPF) algorithm is proposed to accommodate more patterns occurring within a *bi-level.* An idea originally introduced by Burke *et al.* [7] of using a linear array whose size equals the width of the strip is employed. Each element of the array is used to store the height of rectangles packed at that coordinate of the array. However, the drawback of using such an array is that it requires the dimensions of the rectangles and the strip to be integers. Two versions of the CPF algorithm are proposed for use when dealing with floating point data. The first version involves rounding the dimensions (up or down) to the nearest integer, which may not necessarily represent a true packing, but it maintains the characteristics of the data. On the other hand, the second version wastes space by rounding *up* the dimensions to the nearest integer, thereby creating a feasible packing for the original rectangles.

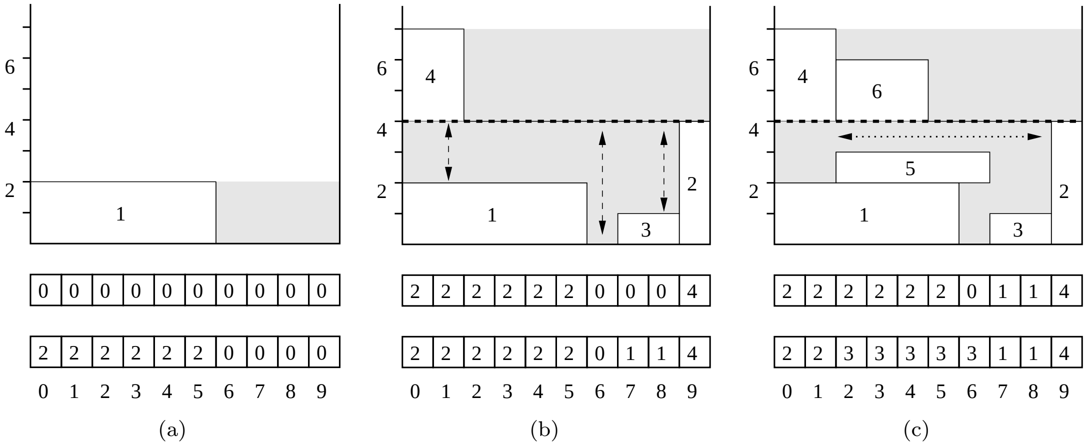


Figure 3: *Examples of how a linear array is populated when a new bi-level is created. (a) The upper linear array containing zeros represents a new bi-level with no rectangles packed. The lower linear array stores the height of the rectangle packed on the* lower *level from coordinates 0 to 5. (b) The upper linear array stores the heights of the first and second rectangles packed. The lower linear array stores height of the third rectangle and the* vertical space *is indicated by the dashed arrows at certain coordinates of the linear array. (c) The fifth rectangle has been compressed down onto the* lower *level by the CFF algorithm. The* horizontal space *is indicated by the horizontal dotted arrow.*

Bi-level Stage. Packing on the *lower* level proceeds exactly as in the BiNFL algorithm, except that a linear array is used to represent the various heights of rectangles packed on the *lower* level only. Before any packing takes place on a bi-level, the linear array contains only zeros. On the *upper* level, the CPF algorithm differs from the BiNFL algorithm in that rectangles are always packed left justified. A *vertical space* on the *lower* level is defined as the space between the lower boundary of the *upper* level and the upper edge of rectangles packed on the *lower* level (or sometimes the lower boundary of the *lower* level) at each coordinate of the linear array. Three vertical spaces of heights 2, 4 and 3 units are indicated by dashed vertical arrows in Figure 3(b) at coordinates 1, 6 and 8 respectively. A *horizontal space* on the *lower* level, on the other hand, is defined as the space between the left-hand edge of a rectangle being considered for compression downwards and the nearest left-hand edge of a rectangle packed on the *lower* level at a height given in the linear array at the coordinate corresponding to the left-hand edge of the rectangle. A horizontal space of 7 units is shown in Figure 3(c) by the horizontal dotted arrows, computed from the coordinates 2 to 8 at a height of 3 (given in the lower linear array, at coordinate 2 which corresponds to the left-hand edge of the sixth rectangle).

Compression Stage. For a rectangle to be compressed down onto the *lower* level, two conditions must be satisfied:

1. The height of the rectangle must exceed the height of the vertical space. The width of a rectangle may be covered by a single value (Figure 4(a)) or different values of the vertical space (Figure 4(b)). If more than one value of the vertical space covers the entire width of the rectangle, the height of the rectangle must exceed the smallest value of the vertical spaces.
2. The width of the rectangle must not exceed the width of the horizontal space.

Provided that the two conditions above are satisfied, the rectangle in question is compressed down so that its bottom edge rests on the top edge of a rectangle on the lower level. The algorithm is expected to perform better if the tallest rectangle on the *upper* level may be compressed onto the *lower* level. A total packing height of 35 units is obtained when the CPF algorithm is applied to our example instance in Table 1, as shown in Figure 6(a).

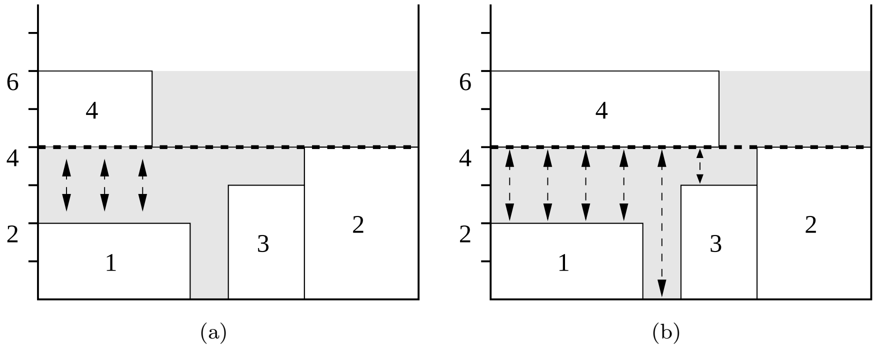


Figure 4: *Example of how the width of a rectangle is covered by: (a) one value of vertical space, or (b) more than one value of vertical space.*

5.3 The Compression Full Fit algorithm

The steps of the *compression full fit* (CFF) algorithm and the CPF algorithm are similar in all respects, except for condition 1 of the compression stage. In the CFF algorithm, a rectangle is compressed down onto the *lower* level provided its height is less than or equal to the vertical space covering the entire width of the rectangle. The advantage of doing this is that the residual vertical space (the vertical space remaining after a rectangle is compressed down) may be considered again when packing the next rectangle. Before rectangle 5 was compressed down in Figure 3(c), there were vertical and horizontal spaces

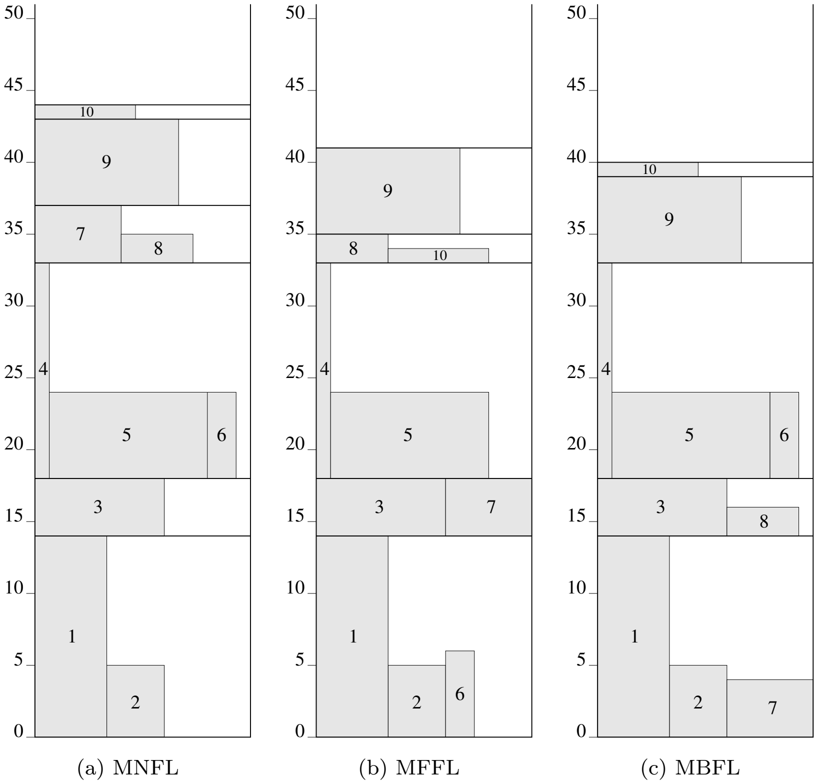


Figure 5: *Packings produced by the modified algorithms described in §5.1 for the example instance of the strip packing problem in Table 1. Rectangle L_i is denoted by i in the figure, for all $i \in \{1, \ldots, 10\}$.*

of 2 and 7 units respectively at coordinate 2. After rectangle 5 was compressed down onto the *lower* level, a vertical space of 1 unit resulted. If rectangle 6 had a height of 1 unit, then it would be compressed down onto the *lower* level. The idea in the CFF algorithm is to increase the probability of packing more rectangles on the *upper* level by utilising the space remaining after compression of a rectangle onto the *lower* level. Once a rectangle is compressed onto the lower level, the space it was supposed to occupy on the upper level may be used to pack other rectangles. The algorithm is expected to perform better if the tallest rectangle on the *upper* level may be compressed onto the *lower* level and if more rectangles fit onto the *upper* level. The latter implies an increased probability of creating fewer levels, hence possibly leading to a decrease in the overall strip height. When the CFF algorithm is applied to our example instance in Table 1, a total packing height of 46 units is obtained, as illustrated in Figure 6(b).

5.4 The Compression Combo algorithm

The *compression combo* (CC) algorithm is a combination of the first conditions of the compression stages of the CPF and CFF algorithms. In the CC algorithm, any rectangle may be compressed down onto the *lower* level regardless of whether it fits fully or partially onto the *lower* level, as long as the second condition is satisfied, namely that the width of the rectangle to be compressed down is at most the width of the horizontal space. When the CC algorithm is applied to our example instance in Table 1, a total packing height of 35 units is again obtained, as illustrated in Figure 6(a).

6 Two New Shelf Algorithms

In this section two new shelf algorithms are suggested. The algorithms highlight two different methods of creating *free space* in between shelves, based on the packing history, so as to cater for the volatility in heights of rectangles still to be packed.

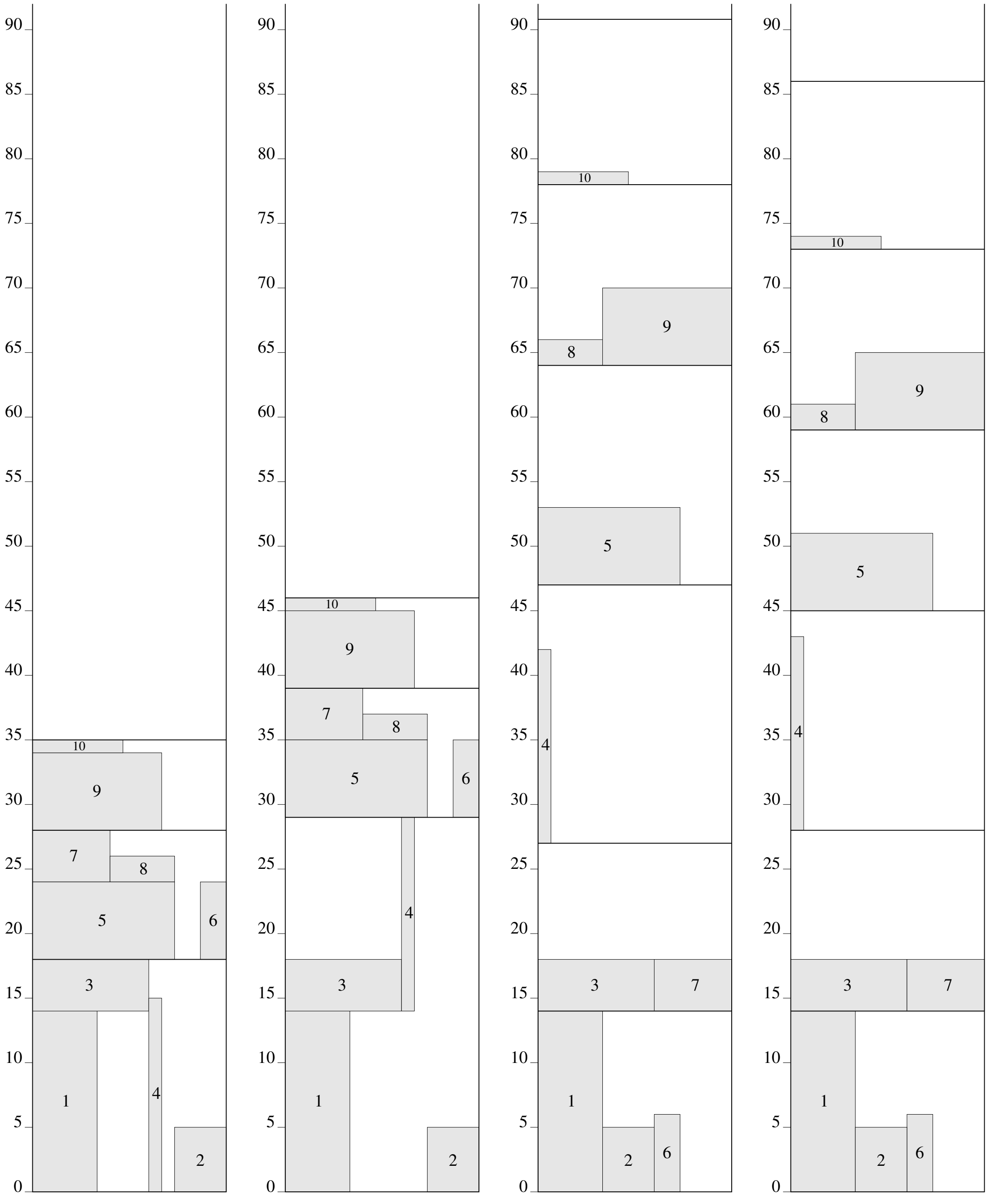
6.1 The Shelf Deviation algorithm

In the newly proposed *shelf deviation* (SDev) algorithm the notion of a shelf *type* refers to a collection of shelves of equal height and the objective is to increase these fixed heights as more *types* are created. A $type_1$ shelf only accommodates rectangles of height $0 < h(L_i) \leq h(L_1)$ where L_1 is the first rectangle to be packed (*i.e.* the height of the first rectangle determines the height of the first shelf *type*). A rectangle whose height fits within this range is referred to as a $type_1$ rectangle. The height of a subsequent shelf of $type_j$ ($j \geq 2$) equals the height of the first rectangle packed on the shelf together with a certain proportion, referred to as the *shelf height increase proportion.* This proportion is computed as the *standard deviation* (stdev) of the rectangle heights already packed on all shelves, *i.e.* $h(type_j) = h(L_{i+1}) + stdev(h(L_1), \ldots, h(L_{i+1}))$. In general, $type_j$ shelves can accommodate rectangles of height $h(type_{j-1}) < h(L_i) \leq h(type_j)$, where $j \geq 2$.

Rectangles are classified according to the shelf *type* to which they belong and are packed onto the lowest shelf of that *type.* New shelf *types* are created above the top-most shelf each time the next rectangle has a height exceeding the height of all existing shelf *types.* It is not necessary for two consecutive shelves to be of the same *type* — the shelf *types* may be interspersed, as long as rectangles are placed onto appropriate shelf *types.* If there is insufficient horizontal space to accommodate a rectangle, a new shelf of the appropriate *type* is created above the top-most shelf for that rectangle. In our example instance in Table 1, a total strip height of 90.80 units is obtained via the SDev algorithm, as shown in Figure 6(c). A pseudocode listing of the steps of this algorithm is given in the appendix.

6.2 The Shelf Difference algorithm

The *shelf difference* (SDiff) algorithm differs from the SDev algorithm only in the way the shelf height increase proportion is computed. In the SDiff algorithm, a $type_1$ shelf is still determined by the height of the first rectangle packed. For a subsequent shelf of type $type_j$ ($j \geq 2$), instead of computing the standard deviation, the shelf height increase is taken as the difference between the height of the rectangle to be packed and the previous shelf height added to the height of the previous shelf *type*, *i.e.* $h(type_j) = (h(L_{i+1}) - h(type_{j-1})) + h(L_{i+1})$. A total packing height of 86 units is obtained when the SDiff algorithm is applied to our example instance in Table 1, as shown in Figure 6(d). A pseudocode listing of the steps of this algorithm is also given in the appendix.



(a) CPF & CC (b) CFF (c) SDev (d) SDiff

Figure 6: *Packings produced by the algorithms described in §5.2 –6.2 for the example instance of the strip packing problem in Table 1. Rectangle L_i is denoted by i in the figure, for all $i \in \{1, \ldots, 10\}$.*

7 Comparison of algorithmic results

The efficiencies of and solution qualities obtained by the algorithms presented in §2–6 were compared by applying them to the 542 benchmark instances of Beasley [4, 5], Burke *et al.* [7], Christofides and Whitlock [8], Hopper and Turton [16, 17] and Mumford-Valenzuela [21]. For a full description on how these benchmark data sets were generated, the reader is referred to [22]. Each algorithm's performance was measured by means of the mean packing height obtained as well as by the mean execution time, computed over all benchmark data sets. Statistical tools used in the comparison of each algorithm's performance include the *student's t-test*, *ANalyses Of VAriance* (ANOVA) and the *chi-squared* test. All these tests were carried out at a 5% level of significance. The *t-test* and *ANOVA* were used to compare the mean packing heights obtained by the algorithms over the 542 instances, while the *chi-squared* test was used to compare the frequencies with which the various algorithms obtained the smallest packing height and to determine whether, statistically, there were any significant differences between these frequencies. Where the results from the ANOVA indicated significant differences, the method of *Least Significance Difference* (LSD) was employed to determine between which algorithms these differences arose.

While testing the algorithms, it was observed that in most of the 542 data sets, the initial rectangles have larger heights than the rectangles towards the ends of the packing lists. Hence each algorithm was tested three times on each data set, by changing the order in which rectangles enter the system from the data set list—either in the *normal or forward* order, in the *reverse* order and in a *random* order.

7.1 Level algorithms

The level algorithms from the literature for *online* packing problems described in §2 were compared with the suggested modifications in §5. The results shown in the first section of Figure 7 indicate that the mean packing height obtained in the *forward* traversal order of the data sets is smaller than in the *reverse* order. This is because in the *forward* order, packing typically begins with rectangles of greater height and for those algorithms that allow revisiting of existing levels, the smaller rectangles may be inserted on any available level with sufficient space—thus decreasing the probability of creating new levels. An ANOVA was carried out separately for each order and in all instances the results revealed that there are significant differences between the mean packing heights obtained. In all three traversal orders, the newly suggested MFFL algorithm obtained the smallest mean packing height, although the LSD indicated that there were no significant differences between the mean packing heights obtained by the MFFL, BFL, FFL and MBFL algorithms (indicated "✕" by entries in Table 2). There were significant differences between the mean packing heights obtained by algorithms that do not revisit existing levels (NFL, MNFL, BiNFL) and those allowing existing levels to be revisited (FFL, BFL, MFFL, MBFL), as expected.

In terms of the algorithmic frequencies of obtaining the smallest packing height (which may be seen in the first section of Figure 9) the results of the chi-squared test revealed that there were significant differences between those frequencies achieved by the various algorithms. The MFFL algorithm has the largest frequency in all traversal orders—hence

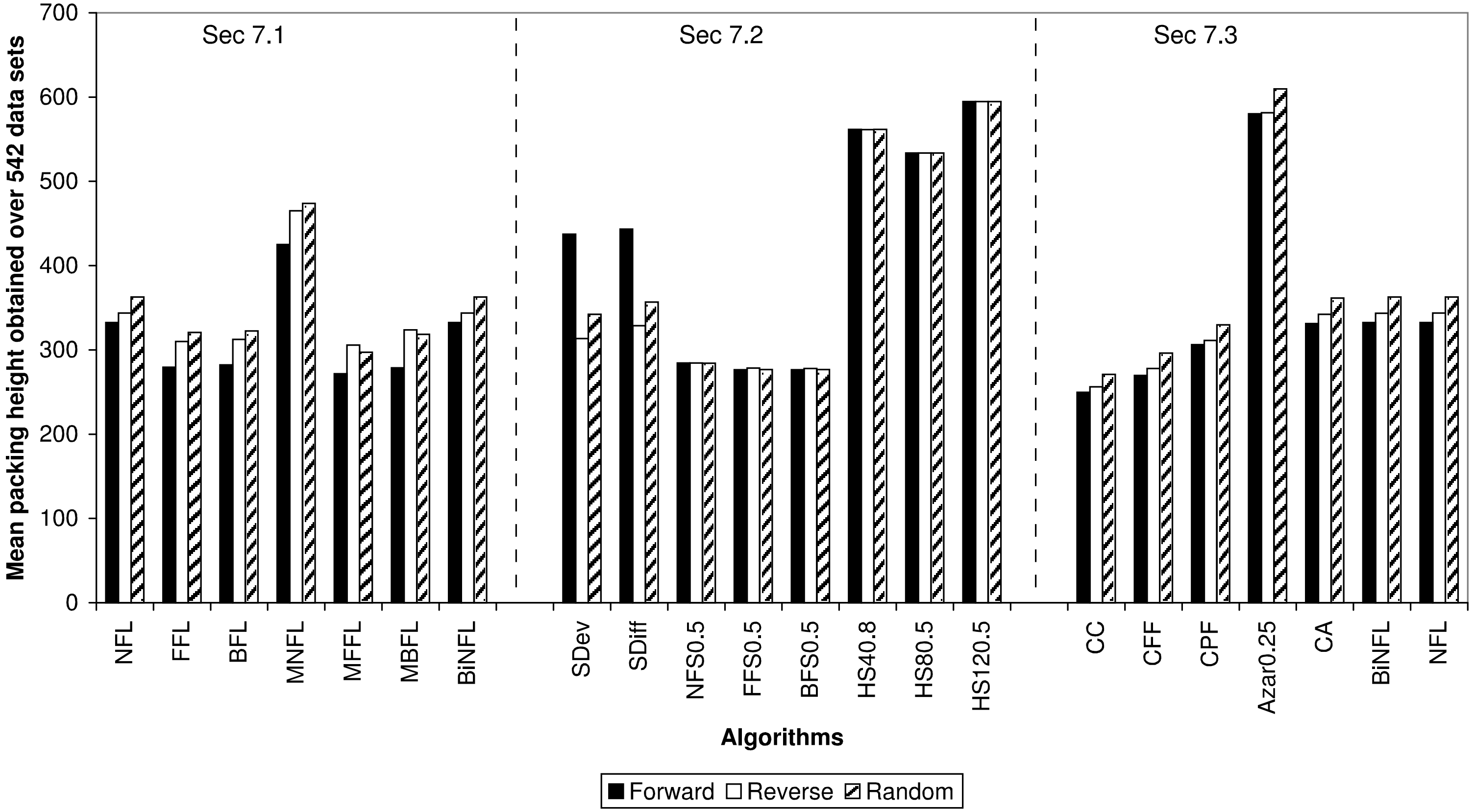


Figure 7: *Comparison of the average packing heights obtained over 542 benchmark data sets by the algorithms described in §2–6. As mentioned in the text, each algorithm was tested for 3 different orders in which rectangles enter the system from the benchmark lists: forward, reverse and random.*

Level algorithms			NFL	FFL	BFL	MNFL	MFFL	MBFL			NFL$_R$	FFL$_R$	BFL$_R$	MNFL$_R$	MFFL$_R$	MBFL$_R$			NFL$_{Ra}$	FFL$_{Ra}$	BFL$_{Ra}$	MNFL$_{Ra}$	FFL$_{Ra}$	MBFL$_{Ra}$
	NFL	332.627							NFL$_R$	343.823							NFL$_{Ra}$	362.742						
	FFL	279.426	✓						FFL$_R$	310.039	✓						FFL$_{Ra}$	320.633	✓					
	BFL	282.215	✓	×					BFL$_R$	312.634	✓	×					BFL$_{Ra}$	322.501	✓	×				
	MNFL	425.084	✓	✓	✓				MNFL$_R$	465.227	✓	✓	✓				MNFL$_{Ra}$	473.743	✓	✓	✓			
	MFFL	271.750	✓	×	×	✓			MFFL$_R$	305.739	✓	×	×	✓			MFFL$_{Ra}$	297.316	✓	×	×	✓		
	MBFL	278.584	✓	×	×	✓	×		MBFL$_R$	323.638	×	×	×	✓	×		MBFL$_{Ra}$	318.585	✓	×	×	✓	×	
	BiNFL	332.627	×	✓	✓	✓	✓	✓	BiNFL$_R$	343.833	×	✓	✓	✓	✓	×	BiNFL$_{Ra}$	362.742	×	✓	✓	✓	✓	✓

Shelf algorithms			SDev	SDiff	NFS$_{0.5}$	FFS$_{0.5}$	BFS$_{0.5}$	HS$_{4_{0.8}}$	HS$_{8_{0.5}}$			SDev$_R$	SDiff$_R$	NFS$_{0.5R}$	FFS$_{0.5R}$	BFS$_{0.5R}$	HS$_{4_{0.8R}}$	HS$_{8_{0.5R}}$			SDev$_{Ra}$	SDiff$_{Ra}$	NFS$_{0.5Ra}$	FFS$_{0.5Ra}$	BFS$_{0.5Ra}$	HS$_{4_{0.8Ra}}$	HS$_{8_{0.5Ra}}$
	SDev	437.459								SDev$_R$	313.592								SDev$_{Ra}$	342.360							
	SDiff	443.299	×							SDiff$_R$	328.616	×							SDiff$_{Ra}$	356.738	×						
	$NFS_{0.5}$	284.532	✓	✓						NFS$_{0.5R}$	284.464	×	✓						NFS$_{0.5Ra}$	284.263	✓	✓					
	$FFS_{0.5}$	276.580	✓	✓	×					FFS$_{0.5R}$	278.543	×	✓	×					FFS$_{0.5Ra}$	276.922	✓	✓	×				
	$BFS_{0.5}$	276.499	✓	✓	×	×				BFS$_{0.5R}$	277.985	×	✓	×	×				BFS05$_{Ra}$	276.923	✓	✓	×	×			
	$HS_{4_{0.8}}$	561.612	✓	✓	✓	✓	✓			HS$_{4_{0.8R}}$	561.216	✓	✓	✓	✓	✓			HS$_{4_{0.8Ra}}$	561.499	✓	✓	✓	✓	✓		
	$HS_{8_{0.5}}$	533.532	✓	✓	✓	✓	✓	×		HS$_{8_{0.5R}}$	533.380	✓	✓	✓	✓	✓	×		HS$_{8_{0.5Ra}}$	533.671	✓	✓	✓	✓	✓	×	
	$HS_{12_{0.5}}$	594.578	✓	✓	✓	✓	✓	×	✓	HS$_{12_{0.5R}}$	594.494	✓	✓	✓	✓	✓	×	✓	HS$_{12_{0.5Ra}}$	594.659	✓	✓	✓	✓	✓	×	✓

Special case algorithms			CC	CFF	CPF	$Azar_{0.25}$	CA	BiNFL			CCR	CFF$_R$	CPF$_R$	Azar$_{0.25R}$	CA$_R$	BiNFL$_R$			CC$_{Ra}$	CFF$_{Ra}$	CPF$_{Ra}$	Azar$_{0.25Ra}$	CA$_{Ra}$	BiNFL$_{Ra}$
	CC	249.664							CC$_R$	256.172							CC$_{Ra}$	271.036						
	CFF	269.568	×						CFF$_R$	278.014	×						CFF$_{Ra}$	296.136	×					
	CPF	306.458	✓	✓					CPF$_R$	311.205	✓	×					CPF$_{Ra}$	329.928	✓	×				
	Azar$_{0.25}$	580.181	✓	✓	✓				Azar$_{0.25R}$	581.475	✓	✓	✓				Azar$_{0.25Ra}$	609.721	✓	✓	✓			
	CA	331.367	✓	✓	×	✓			CA$_R$	342.191	✓	✓	×	✓			CA$_{Ra}$	361.416	✓	✓	×	✓		
	BiNFL	332.627	✓	✓	×	✓	×		BiNFL$_R$	343.644	✓	✓	×	✓	×		BiNFL$_{Ra}$	362.742	✓	✓	×	✓	×	
	NFL	332.627	✓	✓	×	✓	×	×	NFL$_R$	343.823	✓	✓	×	✓	×	×	NFL$_{Ra}$	362.742	✓	✓	×	✓	×	×

Table 2: *LSD results for level, shelf and special case algorithms. A block containing a ✓ (resp. ×) indicates that there are (resp. not) significant differences between the means of the algorithms in the corresponding row and column. The subscripts R and Rand refer to the* reverse *and* random *orders of traversing the data sets respectively.*

this is statistically the best algorithm within this class of level algorithms, at a 5% level of significance.

Further tests were also carried out to determine whether the data set traversal order plays a significant role in each algorithm's performance measured. The results shown in Table 4 indicate that, in terms of the mean packing height obtained, order does not play a significant role in the NFL and BiNFL algorithms. However, when it comes to a frequency analysis, it is only for the MNFL algorithm that traversal order is unimportant (see Table 4).

The class of FFL, MFFL, BFL and MBFL algorithms considered to have achieved better performances in terms of smallest mean packing height obtained, have correspondingly longer execution times than the poorer performing class of NFL, MNFL and BiNFL algorithms (see Table 4). This is an expected result, because in the former class of algorithms the strip has to be searched from the bottom upwards for a level with sufficient space and this is time consuming — particularly for a large number of levels. Ideally the best performing algorithm should achieve the smallest packing height in the quickest time. However, the results indicate that a trade-off exists between algorithms that yield better solutions, but which take longer to execute, and algorithms yielding solutions of lesser quality, but which exhibit faster execution times.

Another investigation was carried out in terms of the aspect ratios of the 1 626 data sets (a combination of all three traversal orders for all 542 benchmark data sets). From the 1 626 instances, only instances where an algorithm obtained the smallest packing height were selected and the *standard deviation* (stdevAR) and *mean* (meanAR) of the aspect ratios of the rectangles in these instances were computed. The fraction $stdevAR/meanAR$, known as the *coefficient of variation* (CV), was used to reflect the variation of rectangle aspect ratios relative to the mean. The numbers of data sets for which each algorithm obtained the smallest height associated with values of the CV are shown in Figure 8. If, for instance, a value of 3 is selected for the CV, it may be seen in the figure that the BFL, MBFL, FFL and MFFL algorithms were all able to obtain the smallest packing height, on average. Of these algorithms, the MFFL algorithm obtained the smallest packing height for the largest number of data sets (825).

An interesting question is the following: Given a data set with a known CV value, which level algorithm should be recommended to give the best solution, on expectation? To answer this question, the CV values for test instances where each algorithm obtained the smallest height were analysed. The objective was to determine a threshold CV value beyond which significant differences occur between frequencies in obtaining the smallest packing height by each level algorithm and below which any of the level algorithms may be used. This was achieved by starting with the smallest value and iteratively determining the frequency with which each level algorithm obtained the smallest packing height for that particular CV value. At each iteration, before the CV value was increased, a chi-squared test was performed to determine whether there were any significant differences between the frequencies obtained by each level algorithm. As the value of CV was increased, a point was reached where a slight increment results in significant differences between the frequencies obtained by each level algorithm. We call such a point the *threshold CV value*, and this value was found to be 0.438 in the case of level algorithms. This means for data

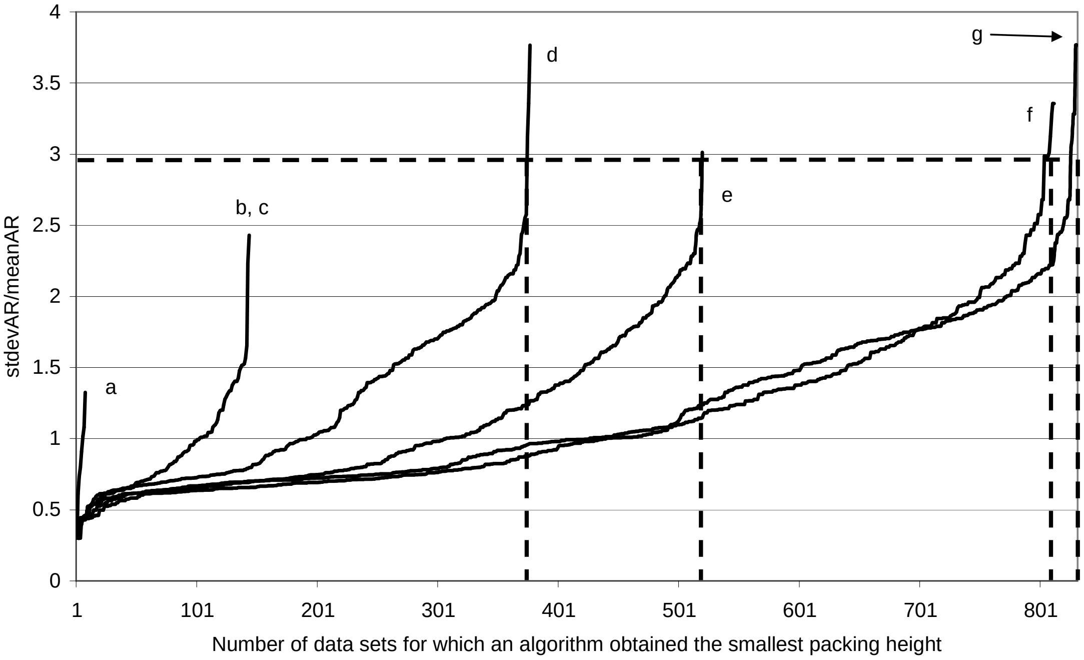


Figure 8: *Aspect ratio analysis for level algorithms: a – MNFL, b – NFL, c – BiNFL, d – MBFL, e – BFL, f – FFL and g – MFFL.*

sets with CV values below the threshold, any of the algorithms may be used, but for values greater than the threshold, the MFFL algorithm is recommended.

7.2 Shelf algorithms

When comparing the shelf algorithms, the algorithms discussed in §3 pose a problem, because they depend on a parameter $0 < r < 1$. Over and above this, the HS_{M_r} algorithms also depend on the value of a parameter $3 \leq M \leq 12$. Hence each of the algorithms was implemented with the representative parameter values $r = 0.2, 0.5, 0.8$ and $M = 4, 8, 12$ resulting in six classes of the algorithms $(\text{NFS}_r, \text{FFS}_r, \text{BFS}_r, \text{HS}_{4_r}, \text{HS}_{8_r}, \text{HS}_{12_r})$. An ANOVA was performed on each of these six classes for the three different traversal orders and the results are shown in Table 3. In the NFS_r class, no significant differences were observed between the $\text{NFS}_{0.5}$ and $\text{NFS}_{0.8}$ algorithms. However, the $\text{NFS}_{0.5}$ algorithm was selected for further comparisons since it achieved a smaller mean packing height over all benchmark sets. For algorithmic instances whose mean heights showed no significant difference, a selection of algorithms to be used for the purposes of further comparison was simply based on the algorithmic instance achieving a smaller mean packing height. Hence the following algorithms were selected in all traversal orders: $\text{FFS}_{0.5}, \text{BFS}_{0.5}, \text{HS}_{4_{0.8}}, \text{HS}_{8_{0.5}}$ and $\text{HS}_{12_{0.5}}$.

The selected shelf algorithms and the two new shelf algorithms were compared in terms of the mean packing height obtained and the results are shown in the second section of Figure 7. The results indicate that, in terms of the mean packing height obtained, the

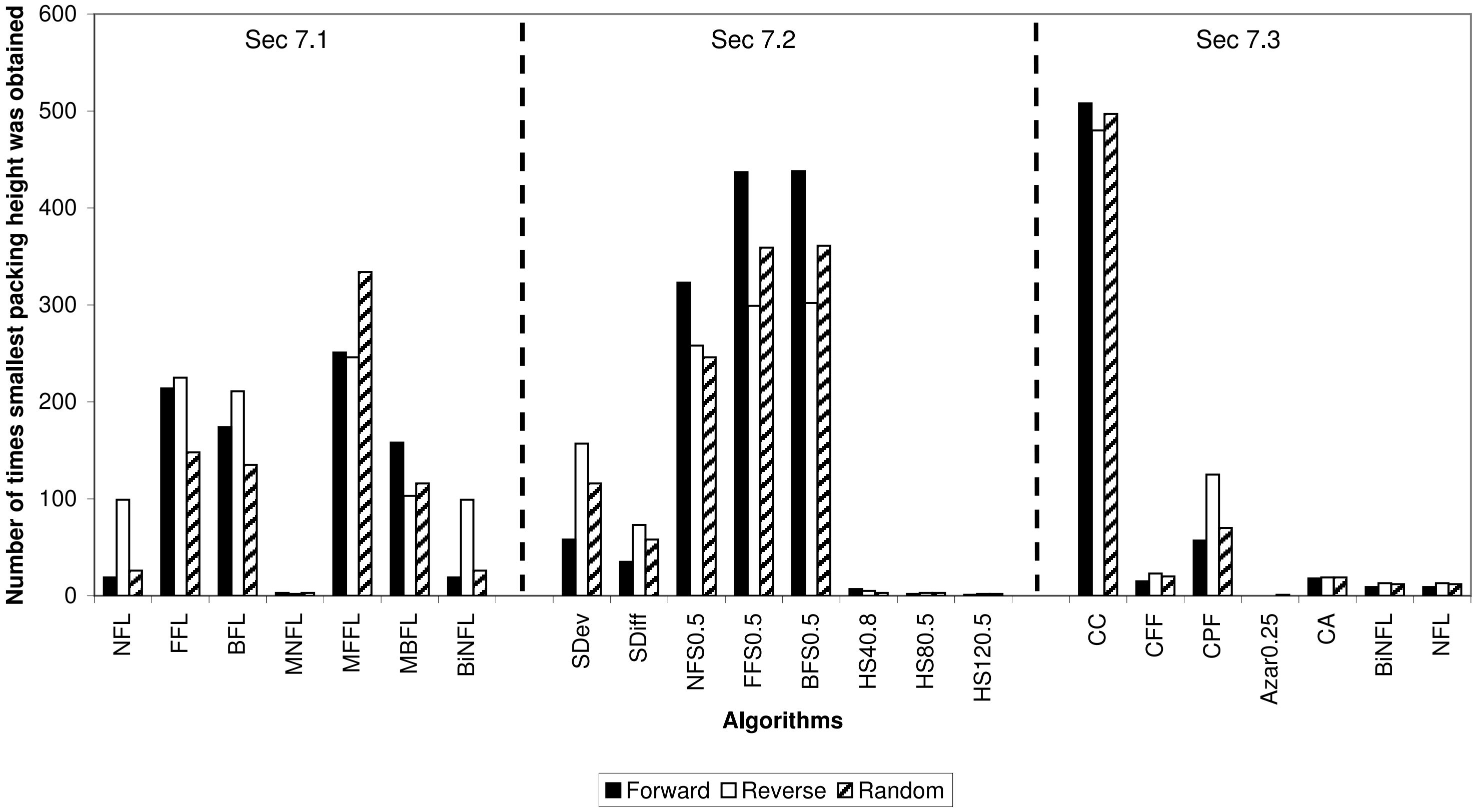


Figure 9: *Comparison of the number of times the smallest packing height was obtained by the various algorithms.*

$\mathrm{NFS}_{0.5}, \mathrm{FFS}_{0.5}$ and $\mathrm{BFS}_{0.5}$ algorithms achieved the best performance, followed by the new SDev and SDiff algorithms.

Considering the algorithms individually and comparing the mean packing heights obtained per traversal order, results from the ANOVA indicated that there were no significant differences, except with the SDev and SDiff algorithms. The results shown in Table 4 indicate that the two algorithms perform better in the *reverse* order. This was an expected result, because the SDev and SDiff algorithms rely on the first rectangle packed and ideally this rectangle must have the smallest height possible. As mentioned, the majority of the benchmark data sets in *reverse* order start with rectangles of relatively small height, hence leading to small increments of each shelf height with an overall smaller total packing height. The mean packing heights obtained by the $\mathrm{NFS}_{0.5}, \mathrm{FFS}_{0.5}$ and $\mathrm{BFS}_{0.5}$ algorithms were not expected to be similar, because a rectangle is classified according to its height, but depending on the widths of the rectangles that are packed first, it may sometimes be necessary to create an additional shelf of appropriate height due to insufficient space on existing shelves of appropriate height. The HS algorithmic instances, on other hand, were expected to yield similar mean packing heights regardless of the order, because the algorithm takes both height and width of the rectangles into consideration before packing on a level.

The results of the chi-squared test indicated that only the HS algorithmic instances display no significant difference with respect to the frequency with which they achieve the smallest packing heights, as illustrated in Table 4 (columns 14–16). The shelf algorithms with parameter r achieve the largest frequency, followed by the SDev and SDiff algorithms (see Figure 9). Based on the results in Table 4 the SDev and SDiff algorithms require shorter execution times than the known shelf algorithms from the literature. A threshold value of 0.456 was computed for the class of shelf algorithms. The $\mathrm{FFS}_{0.5}$ algorithm is recommended for use when dealing with data sets with a CV value larger than this threshold.

7.3 Special case algorithms obeying the tetris constraint

Because the Azar_Y algorithm depends on the threshold constant $0 < Y < 1/2$, three representative values $Y = 0.2, 0.25, 0.3$ were selected in order to determine only one value that may be used for further comparisons with other algorithms obeying the tetris constraint. An ANOVA was carried out and the results revealed that there were no significant differences between the mean packing heights obtained by these three algorithmic instances. The $\mathrm{Azar}_{0.25}$ algorithm was selected, because upon carrying out a chi-squared test, significant differences were found between the frequencies in obtaining the smallest packing heights, showing that the $\mathrm{Azar}_{0.25}$ algorithm achieved the largest frequency (297).

The results shown in the third section of Figure 7 indicate that the newly proposed CC algorithm obtained the smallest mean packing height in the class of algorithms obeying the tetris constraint. An ANOVA was carried out separately for each algorithm to decide whether the traversal order in which rectangles enter the system affects the performance of an algorithm. The results shown in Table 4 indicate that there are no significant differences between mean packing heights obtained per traversal order by each algorithm.

Forward Order

	Height	$NFS_{0.2}$	$NFS_{0.5}$		Height	$FFS_{0.2}$	$FFS_{0.5}$		Height	$\mathrm{BFS}_{0.2}$	$\mathrm{BFS}_{0.5}$		Height	$\mathrm{HS4}_{0.2}$	$\mathrm{HS4}_{0.5}$		Height	$\mathrm{HS8}_{0.2}$	$\mathrm{HS8}_{0.5}$		Height	$\mathrm{HS12}_{0.2}$	$\mathrm{HS12}_{0.5}$
$\mathrm{NFS}_{0.2}$	414.0018			$\mathrm{FFS}_{0.2}$	396.7122			$\mathrm{BFS}_{0.2}$	394.7841			$\mathrm{HS4}_{0.2}$	948.1646			$\mathrm{HS8}_{0.2}$	773.7797			$\mathrm{HS12}_{0.2}$	845.9694		
$\mathrm{NFS}_{0.5}$	284.5318	✓		$\mathrm{FFS}_{0.5}$	276.5798	✓		$\mathrm{BFS}_{0.5}$	276.4986	✓		$\mathrm{HS4}_{0.5}$	608.5404	✓		$\mathrm{HS8}_{0.5}$	533.5324	✓		$\mathrm{HS12}_{0.5}$	594.578	✓	
$\mathrm{NFS}_{0.8}$	336.8876	✓	×	$\mathrm{FFS}_{0.8}$	333.8336	✓	✓	$\mathrm{BFS}_{0.8}$	333.805	✓	✓	$\mathrm{HS4}_{0.8}$	561.6116	✓	×	$\mathrm{HS8}_{0.8}$	549.6021	✓	×	$\mathrm{HS12}_{0.8}$	623.055	✓	×

Reverse Order

	Height	$\mathrm{NFS}_{0.2R}$	$\mathrm{NFS}_{0.5R}$		Height	$\mathrm{FFS}_{0.2R}$	$\mathrm{FFS}_{0.5R}$		Height	$\mathrm{BFS}_{0.2R}$	$\mathrm{BFS}_{0.5R}$		Height	$\mathrm{HS4}_{0.2R}$	$\mathrm{HS4}_{0.5R}$		Height	$\mathrm{HS8}_{0.2R}$	$\mathrm{HS8}_{0.5R}$		Height	$\mathrm{HS12}_{0.2R}$	$\mathrm{HS12}_{0.5R}$
$\mathrm{NFS}_{0.2R}$	413.6771			$\mathrm{FFS}_{0.2R}$	400.8192			$\mathrm{BFS}_{0.2R}$	398.7085			$\mathrm{HS4}_{0.2R}$	947.507			$\mathrm{HS8}_{0.2R}$	772.8757			$\mathrm{HS12}_{0.2R}$	846.5967		
$\mathrm{NFS}_{0.5R}$	284.4645	✓		$\mathrm{FFS}_{0.5R}$	278.5429	✓		$\mathrm{BFS}_{0.5R}$	277.9848	✓		$\mathrm{HS4}_{0.5R}$	608.5404	✓		$\mathrm{HS8}_{0.5R}$	533.3796	✓		$\mathrm{HS12}_{0.5R}$	594.4944	✓	
$\mathrm{NFS}_{0.8R}$	337.1546	✓	×	$\mathrm{FFS}_{0.8R}$	334.4034	✓	✓	$\mathrm{BFS}_{0.8R}$	334.3017	✓	✓	$\mathrm{HS4}_{0.8R}$	561.2164	✓	×	$\mathrm{HS8}_{0.8R}$	549.7103	✓	×	$\mathrm{HS12}_{0.8R}$	624.0787	✓	×

Random Order

	Height	$\mathrm{NFS}_{0.2Ra}$	$\mathrm{NFS}_{0.5Ra}$		Height	$\mathrm{FFS}_{0.2Ra}$	$\mathrm{FFS}_{0.5Ra}$		Height	$\mathrm{BFS}_{0.2Ra}$	$\mathrm{BFS}_{0.5Ra}$		Height	$\mathrm{HS4}_{0.2Ra}$	$\mathrm{HS4}_{0.5Ra}$		Height	$\mathrm{HS8}_{0.2Ra}$	$\mathrm{HS8}_{0.5Ra}$		Height	$\mathrm{HS12}_{0.2Ra}$	$\mathrm{HS12}_{0.5Ra}$
$\mathrm{NFS}_{0.2Ra}$	414.2122			$\mathrm{FFS}_{0.2Ra}$	395.8708			$\mathrm{BFS}_{0.2Ra}$	395.7897			$\mathrm{HS4}_{0.2Ra}$	947.6693			$\mathrm{HS8}_{0.2Ra}$	773.9753			$\mathrm{HS12}_{0.2Ra}$	847.1502		
$\mathrm{NFS}_{0.5Ra}$	284.2634	✓		$\mathrm{FFS}_{0.5Ra}$	276.922	✓		$\mathrm{BFS}_{0.5Ra}$	276.923	✓		$\mathrm{HS4}_{0.5Ra}$	608.6096	✓		$\mathrm{HS8}_{0.5Ra}$	533.6707	✓		$\mathrm{HS12}_{0.5Ra}$	594.6587	✓	
$\mathrm{NFS}_{0.8Ra}$	335.5847	✓	×	$\mathrm{FFS}_{0.8Ra}$	333.832	✓	✓	$\mathrm{BFS}_{0.8Ra}$	333.693	✓	✓	$\mathrm{HS4}_{0.8Ra}$	561.4988	✓	×	$\mathrm{HS8}_{0.8Ra}$	549.9066	✓	×	$\mathrm{HS12}_{0.8Ra}$	624.1357	✓	×

Table 3: *LSD results for shelf algorithms for the values* $r = 0.2, 0.5, 0.8$ *and* $M = 4, 8, 12$.

		Analysis of variance					Chi-squared test					Time		
		Fwd	**Rev**	**Rand**	F_{calc}	F_{crit}	**Frd**	**Rev**	**Rand**	χ^2_{calc}	χ^2_{crit}	**Fwd**	**Rev**	**Rand**
Level algorithms	NFL	332.627	343.823	362.742	1.583	3.001	19	99	26	**81.792**	**5.990**	13.465	9.771	14.554
	FFL	279.426	310.039	320.633	**4.276**	**3.001**	214	225	148	**17.727**	**5.990**	21.077	17.647	21.216
	BFL	282.215	312.634	322.501	**4.143**	**3.001**	174	211	135	**16.665**	**5.990**	21.899	17.472	20.823
	MNFL	425.084	465.227	473.743	**3.402**	**3.001**	3	2	3	0.25	5.99	13.044	9.771	12.423
	MFFL	271.75	305.739	297.316	**3.228**	**3.001**	251	246	334	**17.639**	**5.990**	20.456	16.93	19.454
	MBFL	278.584	323.638	318.585	**6.176**	**3.001**	158	103	116	**13.151**	**5.990**	21.055	17.619	20.07
	BiNFL	332.627	343.833	362.742	1.583	3.001	19	99	26	**81.792**	**5.990**	11.373	11.056	12.432
Shelf algorithms	SDev	437.459	313.592	342.36	**23.409**	**3.001**	58	157	116	**44.852**	**5.990**	10.742	10.103	11.478
	SDiff	443.299	328.616	356.738	**18.809**	**3.001**	35	73	58	**13.241**	**5.990**	11.188	11.623	13.066
	$\text{NFS}_{0.5}$	284.532	284.464	284.263	0	3.001	323	258	246	**12.452**	**5.990**	15.86	16.369	17.426
	$\text{FFS}_{0.5}$	276.58	278.543	276.922	0.007	3.001	437	299	359	**26.236**	**5.990**	18.406	16.285	18.096
	$\text{BFS}_{0.5}$	276.499	277.985	276.923	0.004	3.001	438	302	361	**25.346**	**5.990**	16.686	15.199	16.887
	$\text{HS}_{4_{0.8}}$	561.612	561.216	561.499	0	3.001	7	5	3	1.6	5.99	15.246	14.769	15.544
	$\text{HS}_{8_{0.5}}$	533.532	533.38	533.671	0	3.001	2	3	3	0.25	5.99	12.773	12.301	13.347
	$\text{HS}_{12_{0.5}}$	594.578	594.494	594.659	0	3.001	1	2	2	0.4	5.99	12.906	13.38	14.007
Special case alg.	CC	249.664	256.172	271.036	1.111	3.001	508	480	497	0.804	5.99	11.236	11.506	9.29
	CFF	269.568	278.014	296.136	1.618	3.001	15	23	20	1.69	5.99	10.86	10.946	9.909
	CPF	306.458	311.205	329.928	1.147	3.001	57	125	70	**31.024**	**5.990**	11.799	12.707	11.887
	$\text{Azar}_{0.25}$	580.181	581.475	609.721	0.897	3.001	0	0	1	2	5.99	12.847	13.556	15.495
	CA	331.367	342.191	361.416	1.648	3.001	18	19	19	0.036	5.99	11.12	11.327	12.673
	BiNFL	332.627	343.644	362.742	1.583	3.001	9	13	12	0.765	5.99	11.373	12.198	12.432
	NFL	332.627	343.823	362.742	1.583	3.001	9	13	12	0.765	5.99	13.465	11.056	14.554

Table 4: *Results from the ANOVA, chi-square test and execution times of the level, shelf and special case algorithms obeying the* tetris *constraint. Bold faced entries indicate that significantly different results are achieved for different traversal orders of the data sets.*

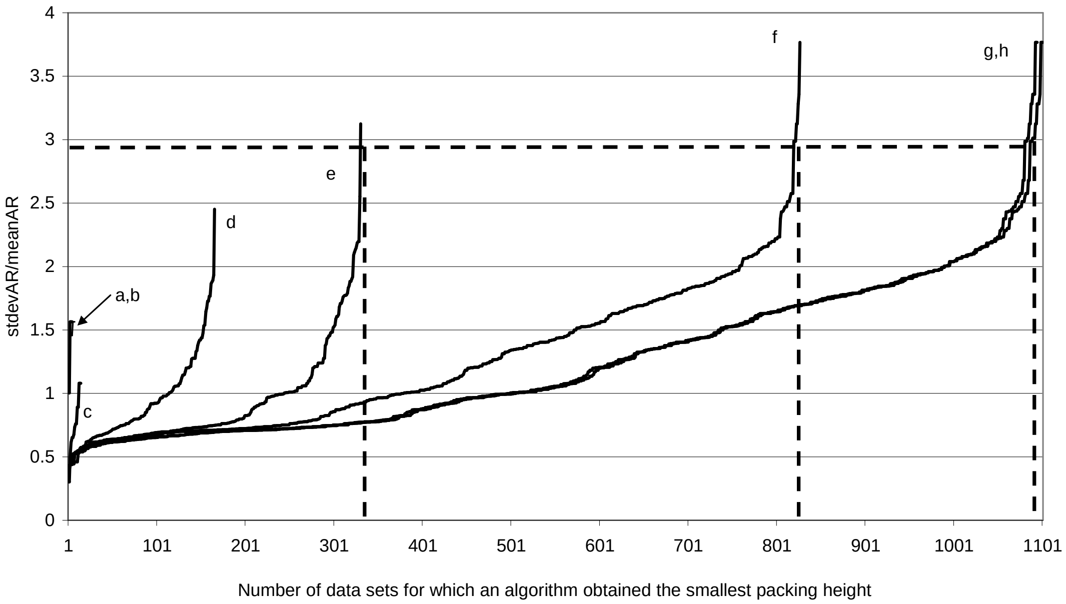


Figure 10: *Aspect ratio analysis for shelf algorithms: a –* $\text{HS}8_{0.5}$*, b –* $\text{HS}12_{0.5}$*, c –* $\text{HS}4_{0.8}$*, d – SDiff, e – SDev, f –* $\text{NFS}_{0.5}$*, g –* $\text{FFS}_{0.5}$ *and h –* $\text{BFS}_{0.5}$.

Comparing the frequencies of obtaining the smallest packing height separately for each algorithm, the results of the chi-squared test (see Table 4) showed that only the CPF algorithm is affected by the order in which rectangles enter the system, achieving the largest frequency in the *reverse* traversal order.

When comparing all the algorithms obeying the tetris constraint, the results of the ANOVA indicate that there are significant differences in terms of their mean packing heights obtained by the various algorithms over all 542 test instances. The results from the LSD (see Table 2) suggest that the newly proposed CC and CFF algorithms are the best performing algorithms with no distinguishable difference between the mean packing heights obtained. However, in terms of the frequency of obtaining the smallest packing height, the two algorithms are distinguishable with the CC algorithm achieving the largest frequency, as may be seen in the results of the chi-squared test.

A CV threshold value of 0.443 was computed, implying that for data sets with CV values smaller than the threshold, any of the special case algorithms may be used. However, for CV values larger than the threshold, the CC algorithm is recommended.

8 Final Remarks

We have investigated a number of on-line algorithms from the literature and classified them into the three classes of *level*, *shelf* and *special case* algorithms. For each class, we were able to find a threshold value for the *coefficient of variation* (CV) such that, given a data set with CV value above this threshold, certain heuristics are recommended above

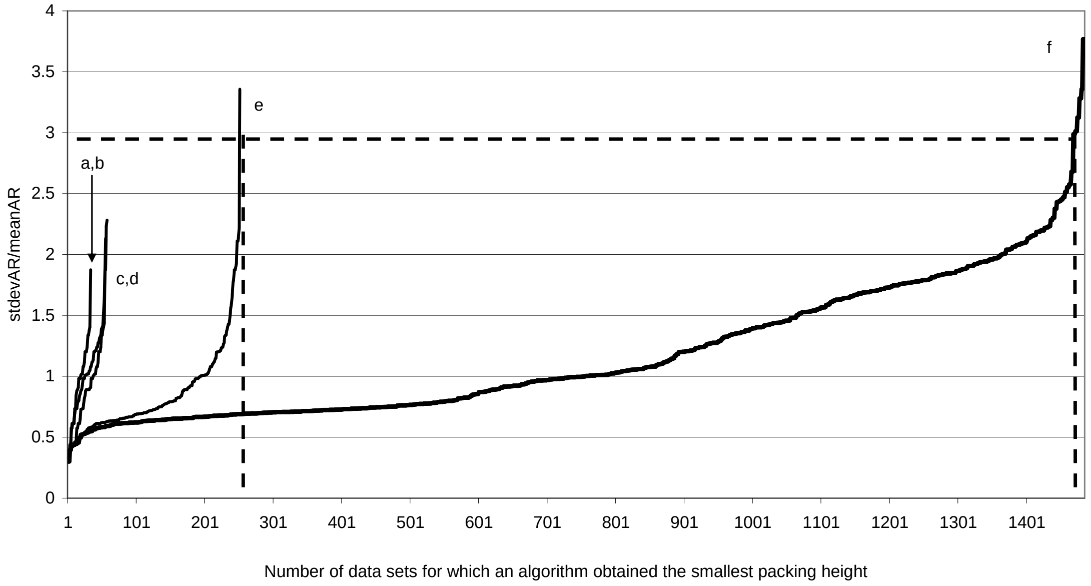


Figure 11: *Aspect ratio analysis for special case algorithms: a – BiNFL, b – NFL, c – CCF, d – CA, e – CPF and f – CC.*

others. In particular, for level (resp. shelf) algorithms, data sets with CV values beyond 0.438 (resp. 0.456), the MFFL (resp. $\text{FFS}_{0.5}$) algorithm is recommended. For special case algorithms obeying the *Tetris* constraint, the CC algorithm is recommended for CV values beyond a threshold value of 0.443.

Two new shelf algorithms (SDev and SDiff) were introduced with an entirely different way of generating additional space within shelves. Instead of using a parameter value (as in some of the classical shelf algorithms from the literature), the new algorithms use the history of the rectangles packed to determine how much free space to create. In the SDev algorithm, the standard deviation of the heights of the rectangles already packed is used while in the SDiff algorithm, the difference in height between the previous shelf and the rectangle to be packed is used. The advantage of the new algorithms is that they do not rely on the selection of any parameter value which, if badly chosen, may lead to poor performance of the algorithm. The new algorithms achieve a better performance than the HS_{M_r} algorithm and can even perform better than the NFS_r, FFS_r and BFS_r algorithms for certain values of the parameter r.

Three modifications (CPF, CFF and CC algorithms) to the CA algorithm [9] have also been proposed, which take more patterns into consideration. When tested on benchmark data sets, the CC algorithm obtained the smallest packing height with the highest frequency.

Finally, it is worth mentioning that in terms of execution time, all algorithms were able to provide a solution to any benchmark instance within 1 second on a 2.00 GHz processor with 224 MB of RAM.

Acknowledgments

Work towards this paper was supported financially by the Third World Organisation for Women in Science (TWOWS), by the South African National Research Foundation (Gun 2072815) and by the Harry Crossley Foundation. We are grateful to Dr Werner Gründlingh and Mr John Part for respectively typesetting assistance and programming advice.

References

[1] Azar Y & Epstein L, 1997, *On two-dimensional packing*, Journal of Algorithms, **25(2)**, pp. 321–332.

[2] Baker BS & Schwarz JS, 1983, *Shelf algorithms for two-dimensional packing problems*, SIAM Journal on Computing, **12(3)**, pp. 508–525.

[3] Bartholdi III JJ, Vate JHV & Zhang J, 1989, *Expected performance of the shelf heuristic for 2-dimensional packing*, Operations Research Letters, **8**, pp. 11–16.

[4] Beasley JE, 1985, *Algorithms for unconstrained two-dimensional guillotine cutting*, Journal of the Operational Research Society, **36(4)**, pp. 297–306.

[5] Beasley JE, 1985, *An exact two-dimensional non-guillotine cutting tree search procedure*, Operations Research, **33(1)**, pp. 49–64.

[6] Brown DJ, Baker BS & Katseff HP, 1982, *Lower bounds for on-line two-dimensional packing algorithms*, Acta Informatica, **18**, pp. 207–225.

[7] Burke EK, Kendall G & Whitwell G, 2004, *A new placement heuristic for the orthogonal stock-cutting problem*, Operations Research, **52(4)**, pp. 655–671.

[8] Christofides N & Whitlock C, 1977, *An algorithm for two–dimensional cutting problems*, Operations Research, **25(1)**, pp. 31–44.

[9] Coffman Jr. EG, Downey PJ & Winkler P, 2002, *Packing rectangles in a strip*, Acta Informatica, **38**, pp. 673–693.

[10] Coffman Jr. EG, Garey DS & Tarjan RE, 1980, *Performance bounds for level oriented two-dimensional packing algorithms*, SIAM Journal on Computing, **9(4)**, pp. 808–826.

[11] Coffman Jr. EG & Shor PW, 1990, *Average-case analysis of cutting and packing in two dimensions*, European Journal of Operational Research, **44**, pp. 134–144.

[12] Coffman Jr. EG & Shor PW, 1993, *Packings in two dimensions: Asymptotic average-case analyisis of algorithms*, Algorithmica, **9**, pp. 253–277.

[13] Coppersmith D & Raghavan P, 1989, *Multidimensional on-line bin packing: Algorithms and worst-case analysis*, Operations Research Letters, **8**, pp. 17–20.

[14] Csirik J & Woeginger GJ, 1997, *Shelf algorithms for on-line strip packing*, Information Processing Letters, **63**, pp. 171–175.

[15] Downey PJ, 2006, *Personal communication via e-mail*, 2006 March 23, contactable at `pete@cs.arizona.edu`.

[16] Hopper E & Turton BCH, 2002, *Problem generators for rectangular packing problems*, Studia Informatica Universalis, **2(1)**, pp. 123–136.

[17] Hopper E & Turton BCH, 2001, *An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem*, European Journal of Operational Research, **128(1)**, pp. 34–57.

[18] Imreh C, 2001, *Online strip packing with modifiable boxes*, Operations Research Letters, **29**, pp. 79–85.

[19] Lee CC & Lee DT, 1985, *A simple on-line bin packing algorithm*, Journal of the Association of Computing Machinery, **32(3)**, pp. 562–572.

[20] Miyazawa FK & Wakabayashi Y, 1998, *Parametric on-line packing*, In Anais do XXX Simpsio Brasileiro de Pesquisa Operacional / Workshop da III Oficina Nacional de Problemas de Corte and Empacotamento, Curitiba-Pr, pp. 109–121.

[21] Mumford-Valenzuela C, Wang PY & Vick J, 2001, *Heuristic for large strip packing problems with guillotine patterns: An empirical study*, Proceedings of the 4th Metaheuristics International Conference, pp. 417–421.

[22] Ntene N & Van Vuuren JH, 2008, *A survey and comparison of guillotine heuristics for the 2D oriented offline strip packing problem*, Submitted.

[23] Ortmann FG, Ntene N & van Vuuren JH, 2008, *New and improved level heuristics for the rectangular strip packing and variable-sized bin packing problems*, European Journal of Operational Research (Submitted).

[24] Ramanan P, Brown DJ, Lee CC & Lee DT, 1989, *On-line bin packing in linear time*, Journal of Algorithms, **10**, pp. 305–326.

[25] Smith H, 2006, *Strip packing algorithms*, [Online], [cited 2006, March 15], Available from: `http://users.cs.cf.ac.uk/C.L.Mumford/ heidi/Algorithms.html`

Appendix

Algorithm 1: Shelf Deviation and Shelf Difference algorithms

Input: Dimensions of the rectangles $\langle w(L_i), h(L_i)\rangle$ and the strip width W.

Output: The height H of the packing obtained in the strip.

```
$h(type_{0,1}) \leftarrow 0$, $h(type_{1,1}) \leftarrow h(L_1)$, $H \leftarrow h(type_{1,1})$
$w(type_{1,1}) \leftarrow W - w(L_1)$
$i \leftarrow 1$, $j \leftarrow 1$, $k \leftarrow 1$, NumTypes $\leftarrow 1$, NumShelfType$_1 \leftarrow 1$
while there is a rectangle to be packed do
  $i \leftarrow i + 1$ (going to the next rectangle)
  while $j \leq$ NumTypes Or rectangle is not packed do
    $k \leftarrow 1$
    if $h(type_{j-1,k}) < h(L_i) \geq h(type_{j,k})$ then
      while $k \leq$ NumTypes$_j$ do
        if $w(type_{j,k}) \geq w(L_i)$ then
          pack rectangle
        else $\{w(type_{j,k}) < w(L_i)\}$
          $k \leftarrow k + 1$ (move on to the next shelf of the same type)
        end if
      end while
      if $k >$ NumShelfType$_j$ then
        NumShelfType$_j \leftarrow$ NumShelfType$_j + 1$ (increase the number of shelves
        of this particular type)
        $w(type_{j,k}) = W - w(L_i)$
        $H \leftarrow H + h(type_{j,k})$
      end if
    else $\{h(type_{j-1,k}) \geq h(L_i)$ or $h(L_i) < h(type_{j,k})\}$
      $j \leftarrow j + 1$ (move on to the next type)
    end if
  end while
  if $j >$ NumTypes then
    create a new shelf type
    NumTypes $\leftarrow$ NumTypes $+ 1$, $k \leftarrow 1$
    $proportion \leftarrow stdev(h(L_1), \ldots, h(L_i))$ SDev algorithm
    $proportion \leftarrow (h(L_i) - h(type_{j-1,k}))$ SDiff algorithm
    $h(type_{j,k}) \leftarrow proportion + h(L_i)$
    $H \leftarrow H + h(type_{j,k})$
  end if
end while
```

Volume 24 (2), pp. 185–193
http://www.orssa.org.za

ORiON
ISSN 0529-191-X
©2008

The Steiner ratio for points on a triangular lattice[*]

PO de Wet[†]

Received: 26 August 2008; Revised: 16 October 2008; Accepted: 24 October 2008

Abstract

The study of spanning trees and Steiner trees arises naturally in applications, such as in the design of integrated circuit boards, communication networks, power networks and pipelines of minimum cost. In such applications the Steiner ratio is an indication of how badly a *minimum* spanning tree performs compared to a Steiner *minimal* tree. In this paper a short proof is presented for the Steiner ratio for points on a triangular lattice in the Euclidean plane. A Steiner tree in two dimensions is "lifted" to become a rectilinear tree in three dimensions, where it is altered. The rectilinear tree is then projected back into the plane and the result readily follows. A short note at the end of the paper compares our three-dimensional rectilinear trees to "impossible objects" such as Escher's "Waterfall."

Key words: Spanning tree, Steiner tree, Steiner ratio, rectilinear Steiner tree, hexagonal Steiner tree, equilateral triangular lattice, Escher.

1 Introduction

Let V be a finite, non-empty set of points in the real space $\mathbb{R}^d$. Let an *arc* be a finite union of straight line segments in $\mathbb{R}^d$ which is homeomorphic to the closed unit interval $[0, 1]$. Let E be a finite set of arcs such that both endpoints of each arc are elements of V. The set of *vertices* V together with the set of *edges* E are called a *topological graph* (which naturally defines a graph) in general, and a *spanning tree* of V if it is furthermore connected and acyclic. If c is a vector in $\mathbb{R}^d$ and G is a topological graph in $\mathbb{R}^d$, then the topological graph $G + c = \{x + c : x \in G\}$ is called a *translate* of G. A *Steiner tree* of V is a spanning tree of some finite vertex set $V \cup S$ in $\mathbb{R}^d$ where all vertices in S have degree at least 3. The vertices in V are called *terminals* and those in S are called *Steiner points.*

The *length* $\|T\|$ of a tree T is defined as the total length of all its segments. A *minimum spanning tree* (MST) of V is a spanning tree of V of smallest length. A *Steiner minimal tree* (SMT) of V is a Steiner tree of V of smallest length. To see that a SMT exists, note that a Steiner tree with n terminals and m Steiner points has $n + m - 1$ edges. Since terminals have degree at least 1 and Steiner points have degree at least 3, there are at least $n/2 + 3m/2$ edges, and thus $n + m - 1 \geq n/2 + 3m/2$. It follows that there are at most $n - 2$

[*] Communicated by CJ Swanepoel, email: swanecj@unisa.ac.za

[†] Department of Decision Sciences, University of South Africa, PO Box 392, Pretoria, 0003, South Africa.

Steiner points, and thus a finite number of possible graph structures for the Steiner trees of V. A shortest Steiner tree with a specific graph structure will have edges which are straight line segments and Steiner points which are all within a closed ball containing V. If we consider the Steiner points to be variable within such a ball, then the length of the tree is a continuous function defined on a compact set, which has to achieve a minimum.

The *Steiner ratio* ρ is defined as

$$\rho = \sup_{\text{any } V \text{ in } \mathbb{R}^2} \frac{\|\mathrm{MST}(V)\|}{\|\mathrm{SMT}(V)\|},$$

where $\mathrm{MST}(V)$ is an MST of V and $\mathrm{SMT}(V)$ is an SMT of V.

The study of spanning trees and Steiner trees has obvious practical value related to the design of power networks, communication networks and pipelines of minimum cost. It also aids in the design of integrated circuit boards, where shorter networks require less time to charge and discharge, making the circuit boards faster. The Steiner ratio is an indication of how badly a minimum spanning tree will perform compared to a Steiner minimal tree. In practice a spanning tree may indeed sometimes be used instead of a Steiner tree, because a minimum spanning tree can be constructed in polynomial time [9], whereas no such algorithm is known to exist for Steiner minimal trees. (The Euclidean Steiner problem is NP-hard [4].)

In 1968 Gilbert and Pollak [5] conjectured the Steiner ratio to be $2/\sqrt{3}$. The fact that $2/\sqrt{3}$ is a lower bound for the Steiner ratio follows from Figure 1, which shows three equidistant vertices, an SMT with edges meeting at 120°, as well as three dotted lines, any two of which form an MST.

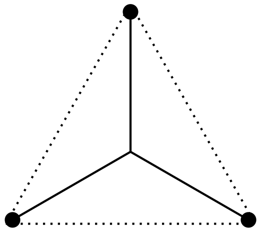


Figure 1: *A Steiner minimal tree.*

It is natural to consider not only three, but also more vertices on an equilateral triangular lattice. It was shown by Du and Hwang [2] that the Steiner ratio for any number of vertices on an equilateral triangular lattice is indeed $2/\sqrt{3}$. The original proof is quite long and incorporates a complicated case analysis. In what follows a shorter proof is presented which is conceptually rather interesting: A Steiner tree in two dimensions is "lifted" to become a rectilinear tree in three dimensions, where it is altered. The rectilinear tree is then projected back into the plane and the result readily follows. The paper closes with a short note which compares three-dimensional rectilinear trees to "impossible objects" such as Escher's "Waterfall."

In 1992 Du and Hwang published a paper [2] and a chapter in a book [3] confirming the correctness of the Gilbert-Pollak conjecture. (The proof for vertices on an equilateral triangular lattice forms an important part of these works.) It has since been shown (see [1] and [7]) that there are fundamental gaps in their argument. The author plans to comment on this extensively in a later paper, but might mention that the general method of Du and Hwang can be adapted for a proof of the Gilbert-Pollak conjecture for 7 points. In this regard their result for vertices on an equilateral triangular lattice has an important consequence.

2 Rectilinear Steiner trees and diagonals

In this section the segments which make up the arcs of a Steiner tree are assumed to be parallel to the x-, y- or z-axis of the space. We refer to such arcs as *rectilinear arcs*. The length of the shortest rectilinear arc between two points is called the *rectilinear distance* between the points. (The norm with which this distance measure is associated is known as the L_1 *norm* or *taxicab norm.*) A Steiner tree consisting only of rectilinear arcs is called a *rectilinear Steiner tree*, and a shortest such tree is called a *rectilinear Steiner minimal tree* (RSMT).

Given n vertices in the plane, a grid can be created by constructing a horizontal line and a vertical line through each vertex. This network is commonly called the *grid graph* of the vertices. The following is a result of Hanan [6], but a new proof is provided.

Lemma 1 *Given n terminals in the plane, then there exists an RSMT with all segments on the grid graph of the terminals. Furthermore, each maximal segment (consisting of a maximal sequence of adjacent collinear segments) contains at least one of the terminals.*

Proof: First consider an RSMT for which the number of horizontal maximal segments that do not contain a terminal is a minimum, and assume it to be greater than zero. Consider the topmost of these maximal segments. Since we have an RSMT, this maximal segment can be moved up or down by a sufficiently small amount Δx without decreasing (or increasing) the length of the tree (Figure 2).

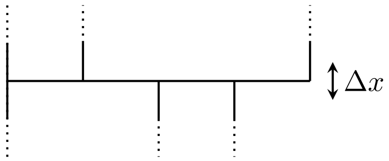


Figure 2: *A maximal segment of an RSMT.*

We move the maximal segment upwards until a terminal or horizontal segment is reached, thus decreasing the number of horizontal maximal segments not containing terminals and providing a contradiction. It follows that there is an RSMT in which each horizontal maximal segment contains a terminal. Among all such RSMTs one may be distinguished in which the number of vertical maximal segments not containing terminals is a minimum. As above, it follows that this number is 0. Hence the RSMT obtained lies on the grid graph. ■

The result of Lemma 1 may be generalized to three dimensions: Given n vertices in $\mathbb{R}^3$ a plane may be constructed perpendicular to each of the axes through each vertex. The intersection of any two planes, with distinct normals, forms a line. The collection of all such lines is known as the *grid graph* of the vertices. By a *maximal planar tree* is meant a tree which lies in a plane perpendicular to one of the axes such that no other tree in the same plane contains it. (See [10] for more on Steiner points in higher dimensions.)

Lemma 2 *Given n terminals in $\mathbb{R}^3$, then there exists a RSMT with all segments on the grid graph of the terminals. Furthermore, each maximal planar tree of the RSMT contains at least one of the terminals.*

Proof: First consider an RSMT for which the number of maximal planar trees in planes perpendicular to the z-axis which do not contain a terminal is a minimum, and assume it to be more than zero. Consider the topmost of these maximal planar trees (*i.e.* with largest z-coordinate). Since we have an RSMT, this maximal planar tree can be moved up or down by a sufficiently small amount Δz without decreasing (or increasing) the length of the tree.

The maximal planar tree may be moved upwards until a terminal or horizontal maximal planar tree is reached, thus decreasing the number of horizontal maximal planar trees not containing terminals and providing a contradiction. It follows that there is an RSMT for which each maximal planar tree which is perpendicular to the z-axis contains a terminal. Among all such RSMTs one may be distinguished in which the number of maximal planar trees perpendicular to the y-axis not containing terminals is a minimum. As above, it follows that this number is 0. Finally the same is done for the x-axis. ■

If a horizontal and a vertical line is constructed through each integer coordinate pair in the plane, an infinite grid graph is obtained. Any translate of this grid graph is called a *square grid*. For three dimensions a *cube grid* is defined similarly, by constructing three lines, parallel to the coordinate axes, through all points with integer coordinates and by considering translates.

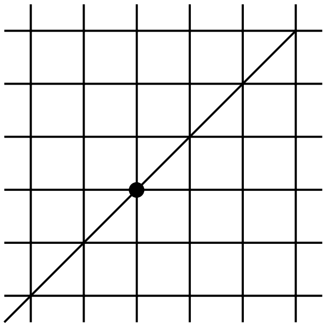


Figure 3: *A square grid.*

In the plane all lines parallel to $y = x$ (*i.e.* parallel to the vector $(1, 1)$) through all integer coordinate pairs, are collectively called *diagonals*. A vertex on one of these diagonals naturally defines a square grid in the plane if an intersection of the square grid coincides with this vertex (see Figure 3). Similarly, *diagonals* in three dimensions pass through all points with integer coordinates and are parallel to the vector $(1, 1, 1)$. A vertex on a diagonal now naturally defines a cube grid. Note that two points on diagonals in the plane (three dimensional space) define the same square grid (cube grid) if they have the same x or y (x or y or z) coordinate.

Consider the following problem: Given n different diagonals, each with a terminal on it, where should the terminals be for the RSMT to have minimal length? For two dimensions it is not difficult to see that the result of the following lemma is true.

Lemma 3 *Given n terminals on diagonals in the plane, the terminals may be slid along the diagonals to new positions so that they all have the same y-coordinate and so that the new RSMT is not longer than the initial one.* ■

In three dimensions the problem is more complex. Note that the RSMT of $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ has minimal length and that the result of Lemma 3 is not true if the points are moved along diagonals to lie in the same plane. To see why this is so, consider the three diagonals which go through $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$. Fix one terminal at $(0, 0, 1)$ while allowing the other two terminals to be moved on their respective diagonals. Next construct around each terminal an octahedron such that all points on the surface of the octahedron are at rectilinear distance 1 from the terminal, as shown in Figure 4.

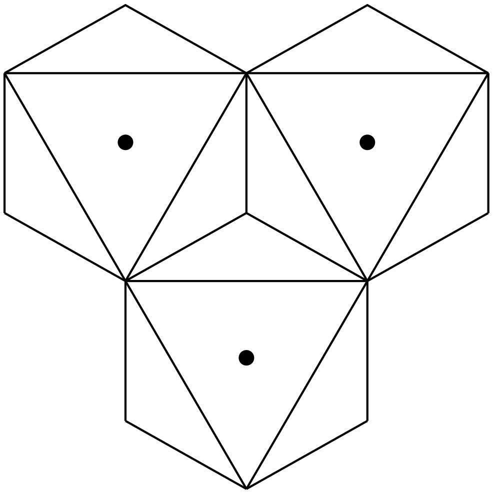


Figure 4: *Octahedrons. Diagonals, as projected onto a plane perpendicular to the vector* $(1, 1, 1)$, *are indicated by means of dots.*

The only positions for the two terminals not yet fixed which will ensure that the union of the three octahedra is connected, are $(1, 0, 0)$ and $(0, 1, 0)$. Now since any RSMT for the three terminals will for each terminal contain a path connecting the terminal to the surface of the octahedron, the RSMT cannot be shorter than 3, and this is only achieved when $(0, 0, 0)$ is a Steiner point. Finally, the RSMT will at best remain the same if another terminal is introduced, so that the RSMT remains the same after introduction of the terminal $(0, 0, 0)$.

Lemma 4 *Given n terminals on diagonals in three dimensional space, then we can slide the terminals along the diagonals to new positions so that they all define the same cube grid and so that the new RSMT is not longer than the initial one.*

Proof: Assume that the result of the lemma is false. This implies that if the positions of the terminals on the diagonals are such that the length of the RSMT is minimal, then the terminals define more than one cube grid. Assume that the number of cube grids defined is as small as possible and that the RSMT is in the form described by Lemma 2.

Consider the set A of all terminals which define a particular cube grid. Each horizontal maximal planar tree which contains a terminal or terminals from A may be moved up or down (together with the terminals) by a sufficiently small amount Δz so that the change in the length of the RSMT is linear. If all horizontal maximal planar trees with terminals from A are moved upwards simultaneously, then the change in the length of the RSMT remains linear until some terminal in A has the same z-coordinate as a terminal which is not in A. Let Z be the length of the upward movement for this to happen.

The maximal planar trees which are perpendicular to the x-axis and which contain terminals in A may similarly be moved in a positive direction until some terminal in A has the same x-coordinate as a terminal which is not in A. Let X be the length of this movement and let Y be the length of the corresponding movement in the positive y-direction. The three movements can be combined to achieve movement of the elements of A along the direction of vector $(1, 1, 1)$ with linear change in the length of the RSMT if this Δd is sufficiently small. Since the RSMT has minimal length, the length of the RSMT has to stay constant. Let $D = \min(X, Y, Z)$. If the elements of A are moved by a distance D in the positive x-, y- and z-directions, then an RSMT with the same length is obtained, with all terminals on diagonals, and for which the number of cube grids defined by the terminals is one fewer, providing a contradiction and showing that it is possible for all terminals to define the same cube grid. ■

3 Hexagonal Steiner trees

Given three directions, each two of which form an angle of 120°, a Steiner tree on n points in the plane for which all line segments are parallel to these directions, is called a *hexagonal Steiner tree.* A shortest such tree is called a *hexagonal Steiner minimal tree* (HSMT). A *junction* is either a Steiner point or a non-terminal point where two segments join at different angles. The example in Figure 5 has four junctions. For terminals on an equilateral triangular lattice, the following result holds. (The result is known [2], but a novel proof is provided.)

Lemma 5 *Consider any set of n terminals on an equilateral triangular lattice. Let the three directions for hexagonal Steiner trees be parallel to the edges of the equilateral triangles of the lattice. Then there exists an HSMT for which all junctions are lattice points.*

Proof: The proof proceeds by using Lemma 4. The projection of all diagonals in $\mathbb{R}^3$ onto a plane perpendicular to the vector $(1, 1, 1)$ forms an equilateral triangular lattice.

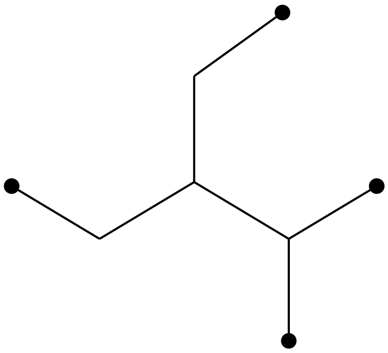


Figure 5: *A hexagonal Steiner tree.*

Furthermore, a hexagonal Steiner tree for n lattice points can be lifted to a rectilinear tree in $\mathbb{R}^3$ with the terminals on diagonals, such that the projection of this rectilinear tree onto the plane parallel to $(1, 1, 1)$ returns the hexagonal Steiner tree. The desired HSMT is now obtained as follows: Begin with any HSMT, lift this tree into $\mathbb{R}^3$, replace it by a tree of equal length according to Lemma 4 (terminals now all lie on vertices of the same cube grid), modify this tree by using Lemma 2 (all segments now also lie on the cube grid), finally project the tree back to the plane (perpendicular to $(1, 1, 1)$) and note that this is still an HSMT for which all junctions are now lattice points. ■

4 Vertices on an equilateral triangular lattice

The following lemma is due to Weng [11].

Lemma 6 *Given a set P of vertices in the plane together with the directions for hexagonal Steiner trees, it follows that*

$$\frac{\|HSMT(P)\|}{\|SMT(P)\|} \leq 2/\sqrt{3}.$$

Proof: Note, for a triangle ABC with a 120° angle at B, that $\|AB\| + \|BC\| \leq 2/\sqrt{3}\|AC\|$. Now each line of an SMT can be replaced by two lines along the given directions, thus forming a sufficiently short hexagonal Steiner tree for the lemma to hold. ■

A set of vertices on an equilateral triangular lattice is called a *cluster* if the graph obtained by connecting adjacent vertices is connected.

Theorem 1 *For any cluster of vertices on an equilateral triangular lattice the Steiner ratio is $\rho = 2/\sqrt{3}$.*

Proof: Choose the directions for hexagonal Steiner trees parallel to the sides of a smallest triangle in the lattice. From Lemma 5 it follows that all lines of an HSMT connect adjacent vertices on the lattice. It follows that the length of an HSMT is equal to that of an MST. Lemma 6 may now be used to complete the proof. ■

5 Conclusion

It was shown in this paper that it is possible to lift a two dimensional hexagonal tree to obtain a three dimensional rectilinear tree, and that this tree may be altered so that all edges lie on a grid. It is much easier to establish the Steiner ratio for any cluster of vertices on an equivalent triangular lattice adopting this lifting proof technique than adopting a direct case-analysis approach.

It is interesting to note that this lifting technique cannot be applied to a circuit, and that this fact can be used to create an impossible object such as MC Escher's "Waterfall," shown with permission in Figure 6. If any part of the circuit is covered (which turns it into a tree), then the picture can be interpreted as an ordinary object. See [12] for a more detailed discussion of Escher's "Waterfall."

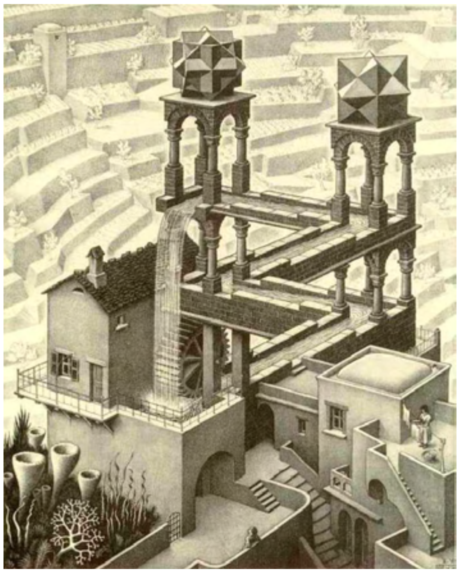


Figure 6: *M.C. Escher's "Waterfall". (©2008 The M.C. Escher Company – the Netherlands [8]. All rights reserved. Used by permission.)*

Acknowledgement

I would like to thank Prof Konrad Swanepoel under whom I completed the PhD at Unisa out of which this paper stems.

References

[1] De Wet PO, 2008, *Geometric Steiner minimal trees*, PhD Dissertation, University of South Africa, Pretoria.

[2] Du DZ & Hwang FK, 1992, *A proof of the Gilbert-Pollak conjecture on the Steiner ratio*, Algorithmica, **7**, 121–135.

[3] Du DZ & Hwang FK, 1992, *Computing in Euclidean geometry*, World Scientific, Singapore.

[4] Garey M, Graham R & Johnson D, 1977, *The complexity of computing Steiner minimal trees*, SIAM Journal on Applied Mathematics, **32**, 835–859.

[5] Gilbert EN & Pollak HO, 1968, *Steiner minimal trees*, SIAM Journal on Applied Mathematics, **16**, 1–92.

[6] Hanan M, 1966, *On Steiner's problem with rectilinear distance*, SIAM Journal on Applied Mathematics, **14**, 255–265.

[7] Ivanov AO & Tuzhilin AA, 2008, *Immersed polygons and their diagonal triangulations*, Izvestiya: Mathematics, **72(1)**, 63–90.

[8] MC Escher Company, 2008, *Waterfall* (M.C. Escher), [Online], [Cited: 2 August 2008], Available: `http://www.mcescher.com`.

[9] Papadimitriou CH & Steiglitz K, 1982, *Combinatorial optimization: Algorithms and complexity*, Prentice-Hall, Englewood Cliffs (NJ).

[10] Snyder TL, 1992, *On the exact location of Steiner points in general dimension*, SIAM Journal on Computing, **21**, 163–180.

[11] Weng JF, *Steiner problem in hexagonal metric*, unpublished manuscript.

[12] Wikipedia, 2008, *Waterfall* (M.C. Escher), [Online], [Cited: 2 August 2008], Available: `http://en.wikipedia.org/wiki/Waterfall_(M._C._Escher)`.

Instructions to Authors

Submission of manuscripts
Anonymous papers (accompanied by a cover e-mail detailing the names and affiliations of authors) may be submitted electronically (preferably as postscript or pdf documents typeset in LaTeX) to the Editor at `vuuren@sun.ac.za` — the only *other* file format that will be accepted is MS Word documents. Hard copies of anonymous papers (accompanied by a covering letter detailing the names and affiliations of authors) may also be submitted **in triplicate** to The Editor: ORiON, Prof JH van Vuuren, Applied Mathematics Division, Department of Mathematical Sciences, University of Stellenbosch, Private Bag X1, Matieland, 7602, South Africa.

Preparation of manuscripts
Authors are requested to conform to the example paper format available in postscript and pdf formats on the ORSSA webpage `http://www.orssa.org.za` → `ORiON` → `Submissions` → `Example of Paper Format`. This format is also supported by the ORiON LaTeX style sheet (which may be downloaded from `http://www.orssa.org.za` → `ORiON` → `Submissions` → `Style Sheets`). Instructions on how to use these style sheets are available in postscript and pdf formats at `http://www.orssa.org.za` → `ORiON` → `Submissions` → `Instructions for Style Sheets`.

Author and affiliation details
The names of all authors, their affiliations, postal addresses, e-mail addresses and fax numbers should be included in a cover letter or e-mail accompanying submissions. These items will be incorporated into the manuscript *by the business manager upon acceptance* (submissions should not originally include this information, so as to facilitate blind peer review).

Abstracts and key words
Papers submitted in English should be preceded by an abstract not exceeding approximately 300 words in length. However, all papers not in English should be accompanied by an *extended and detailed* abstract in English of about 1 000 words in length, in addition to a brief abstract in the language of submission (not exceeding approximately 300 words in length). In all cases a list of suitable key words should be listed directly after the abstract, so as to facilitate searches in electronic databases to which ORiON abstracts are contributed.

Mathematical formulae
All mathematical formulae should form part of sentences (and should hence include punctuation, where necessary, but should not be preceded by colons). Mathematical formulae and expressions should be typeset in text lines where possible, the only exceptions being formulae that are so bulky that they would force increased line spacing if included in the text, or formulae that have to be numbered for further referencing.

Formatting
All Latin abbreviations or phrases, such as *e.g., i.e., et al., vice versa, etc.* should be typeset in italics. If MS Word is used to prepare a manuscript, it should be utilised appropriately. For example, **all** mathematical formulae and expressions should be typed in Microsoft Equation Editor (and not merely as italicised text) and section headings should be typeset as *headings* (and not as enlarged, bold faced normal text). Both the full stop and comma are acceptable as decimal separators — however, a choice between these separators should be made and applied consistently by authors.

Figures and tables
Figures and Tables should be numbered consecutively, using separate numbering sequences (*e.g.* Table 1, Table 2, Figure 1, Table 3, Figure 2, . . . rather than Table 1, Table 2, Figure 3, Table 4, Figure 5, . . .). Tables and figures should be accompanied by detailed captions and should be included in the main body of text (not on separate pages at the end of the manuscript). Authors need not include separate high quality photographs or electronic copies of figures when submitting

manuscripts — these will be requested by the business manager (if necessary) upon acceptance of the manuscript. All Figures and tables should be referenced in the text.

Theorems, algorithms and other numbered environments

Theorems, Algorithms and other numbered environments should be numbered consecutively, using separate numbering sequences (*e.g.* Theorem 1, Theorem 2, Algorithm 1, Corollary 1, Algorithm 2, . . . rather than Theorem 1, Theorem 2, Algorithm 3, Corollary 4, Algorithm 5, . . .). These environments are supported by the official ORiON LaTeX style sheet — further information on how to utilise these environments in LaTeX may be found at `http://www.orssa.org.za` → `ORiON` → `Submissions` → `Instructions for Style Sheets`.

Literature citations

Authors have a choice whether to follow the Harvard (author date) standard or the Vancouver (numerical) standard for literature citations — one of these standards should be applied consistently. Footnotes should not be used for citation purposes. All items in the bibliography should be cited in the text.

According to the Harvard standard literature citations in the text should proceed by listing the relevant author's name and the year of publication (*e.g.* "An optimal solution exists (Dantzig 1963)." or "According to Dantzig (1963) an optimal solution exists."). Additional information, such as page numbers, chapter numbers, theorem numbers, *etc.*, may be given directly after the date, separated by a comma (*e.g.* "An optimal solution exists (Dantzig 1963, p. 69)." or "According to Dantzig (1963, p. 69) an optimal solution exists."). For literture citations involving two authors, both authors' names should be listed, separated by an amprasand (*e.g.* "An optimal solution exists (Dantzig & Wolfson 1967, Theorem 4.2)." or "According to Dantzig & Wolfson (1967, Theorem 4.2) an optimal solution exists."). For literture citations involving more than two authors, only the first author's name should be listed in conjunction with the phrase *et al.* (*e.g.* "An optimal solution exists (Dantzig *et al.* 1972, §3)." or "According to Dantzig *et al.* (1972, §3) an optimal solution exists."). In cases of more than one bibliography entry per author per year, small alphabetical characters should be used to distinguish between references (*e.g.* "An optimal solution exists (Dantzig 1965b)." or "According to Dantzig (1963b) an optimal solution exists.").

According to the Vancouver standard literature citations in the text should proceed by listing the number of the relevant bibliography entry (*e.g.* "An optimal solution exists [7]." or "According to Dantzig [7] an optimal solution exists."). Additional information, such as page numbers, chapter numbers, theorem numbers, *etc.*, may be given directly after the citation number, separated by a comma (*e.g.* "An optimal solution exists [7, p. 69]." or "According to Dantzig [7, p. 69] an optimal solution exists."). For literature citations involving two authors, both authors' names may be listed, separated by an amprasand (*e.g.* "An optimal solution exists [9, Theorem 4.2]." or "According to Dantzig & Wolfson [9, Theorem 4.2] an optimal solution exists."). For literature citations involving more than two authors, only the first author's name may be listed in conjunction with the phrase *et al.* (*e.g.* "An optimal solution exists [10, §3]." or "According to Dantzig *et al.* [10, §3] an optimal solution exists.").

A more comprehensive list of citation examples (using both standards) may be found at `http://www.orssa.org.za` → `ORiON` → `Submissions` → `Example of Paper Format` by clicking on the link `Examples of Reference Citations and Bibliography Listings`.

References

Books should be listed in the bibliography by including the surnames and initials (without punctuation) of all authors and/or editors (IN SMALL CAPITALS), the date of publication, the title (*in italics*, using small letters only, the only exceptions being the first word of the title and proper nouns), the edition (if second or higher), the publisher, the city of publication (followed by the official two-letter abbreviation of the state for cities in the United States — no country names should be listed), and the relevant pages cited (if appropriate), such as in the examples below:

[1] Dantzig B, 1963, *Linear programming and extensions*, 2nd Edition, Princeton University Press, Princeton (NJ).

[2] Gendreau M, Laporte G & Potvin J-Y, 2002, *Metaheuristics for the capacitated vehicle routing problem*, pp. 129–149 in Toth P & Vigo D (Eds.), *The vehicle routing problem*, SIAM, Philadelphia (PA).

Journals should be listed in the bibliography by including the surnames and initials of all authors (in small capitals), the date of the issue, the title of the relevant paper (*in italics*), the title of the journal (not abbreviated), the volume (and issue/part) number (**in bold face**), and the pages of the relevant paper, such as in the example below:

[3] Norese MF & Toso F, 2004, *Group decision and distributed technical support*, International Transactions in Operational Research, **11(4)**, pp. 395–417.

Online resources should be listed in the bibliography by including the surnames and initials of the web page designer (if known, in small capitals), the date of construction of the web page (if known), the title of the web page (if known, *in italics* — this is typically found in the title bar at the very top of the web page), an indication that it is an online reference, the date on which the site was accessed, and the URL (`in true type or courier fonts`), such as in the example below.

[4] Skiena SS, 1997, *The algorithm design manual*, [Online], [Cited September 9th, 2004], Available from `http://www2.toki.or.id/book/algdesignmanual/index.htm`

Theses and dissertations should be listed in the bibliography by including the surnames and initials of the author, the date, the thesis (or dissertation) title, the university where the thesis (or dissertation) was submitted and the city in which the university is situated, such as in the example below [5]. An example of an unpublished technical report [6] is also shown below.

[5] Vumbi AI, 2003, *Algorithmic complexity*, MSc Thesis, University of Stellenbosch, Stellenbosch.

[6] Hamming R, 1956, *On the amount of redundancy required to correct information errors*, (Unpublished) Technical Report TR 1956-371, Bell Laboratories, Murray Hill (NJ).

An example of the format in which an unpublished conference paper should be listed in the bibliography is given in [7] below, whilst an example of the bibliography listing format of a paper published in conference proceedings is shown in [8] below.

[7] Lacomme P, Prins C & Ramdane-Chérif W, 2002, *Fast algorithms for general arc routing problems*, Paper presented at the 16th Triennial Conference of the International Federation of Operations Research Societies, Edinburgh.

[8] Wilkinson C & Gupta SK, 1969, *Allocating promotional effort to competing activities: A dynamic programming approach*, Proceedings of the 5th Triennial Conference of the International Federation of Operations Research Societies, Venice, pp. 419–432.

The bibliography should be arranged in alphabetical order, according to first author surnames.

Note that although authors may use either the Harvard standard or the Vancouver standard (consistently) for citation purposes in the text, all references in the bibliography are expected to adhere to the guidelines above — irrespective of which citation standard is utilised by authors. A more comprehensive list of referencing examples may be found at `http://www.orssa.org.za` → `ORiON` → `Submissions` → `Example of Paper Format` by clicking on the link `Examples of Reference Citations and Bibliography Listings`.

Subscribe to ORiON

Subscription fees per issue

Local	R150,00
Overseas	US$40,00

Make all cheques and postal orders payable to ORSSA

Subscribe electronically
`http://www.orssa.org.za` → `ORiON` → `Subscriptions`

OR

Subscribe by mail
Send all correspondence regarding subscription to:
The Business Manager: ORiON
PO Box 3184
Matieland
7602
South Africa

Order form
Previous volumes can be ordered from the Business Manager.

ORiON **Volume 24(2)** **2008**

I wish to subscribe to the following issue of ORiON

Volume and number: ....

Name: ....

Address: ....

....

....

....

....

....

Postal code: ....

Country: ....