

A Simulated Annealing Enhancement of the Best-Fit Heuristic for the Orthogonal Stock-Cutting Problem

Edmund K. Burke, Graham Kendall, Glenn Whitwell

School of Computer Science, University of Nottingham, Nottingham, NG8 1BB, United Kingdom
{ekb@cs.nott.ac.uk, gxk@cs.nott.ac.uk, gxw@cs.nott.ac.uk}

The best-fit heuristic is a simple yet powerful one-pass approach for the two-dimensional rectangular stock-cutting problem. It had achieved the best published results on a wide range of benchmark problems until the development of the approaches described in this paper. Here, we illustrate how improvements in solution quality can be achieved by the hybridisation of the best-fit heuristic together with simulated annealing and the bottom-left-fill algorithm. We compare and contrast the new hybrid approach with other approaches from the literature in terms of execution times and the quality of the solutions achieved. Using a range of standard benchmark problems from the literature, we demonstrate how the new approach achieves significantly better results than previously published methods on almost all of the problem instances. In addition, we provide results on 10 new benchmark problems to encourage further research and greater comparison between current and future methods.

Key words: stock cutting; simulated annealing; heuristics; industries; manufacturing

History: Accepted by Michel Gendreau, Area Editor for Heuristic Search and Learning; received July 2004; revised October 2005, August 2006, July 2007, January 2008, June 2008; accepted July 2008. Published online in *Articles in Advance* February 5, 2009.

1. Introduction

Cutting and packing motivates many areas of operations research and arises, amongst others, in the paper, wood, glass, and metal industries. There have been several articles that provide categorised bibliographies and general overviews for the research undertaken under the term cutting and packing (Gilmore 1966, Golden 1976, Coffman et al. 1984, Dyckhoff 1990, Dowland and Dowland 1992, Sweeney and Paternoster 1992, Lodi et al. 2002). Cutting and packing problems have many different formulations that are usually distinguished by the dimensionality of the problem. For example, the one-dimensional problem involves the subdivision of material stock lengths (i.e., rods) into the required lengths such that the material is utilised efficiently (Gilmore and Gomory 1961, Belov and Scheithauer 2006). The two-dimensional problem involves the placement of shapes onto one or many sheets of material in order to maximise material utilisation. The problem can be further classified by the type of items that are to be placed. Whereas the rectangular (or orthogonal) problem involves the placement of rectangles onto rectangular stock sheets (Baker et al. 1980, Hopper and Turton 1999), the irregular variant can involve items and sheets of any two-dimensional form that may contain arcs, concavities, and holes (Burke et al. 2006), and typically require more sophisticated geometric algorithms

(Bennell and Song 2008, Burke et al. 2007, Bennell et al. 2001). Three-dimensional problems are generally constrained to the packing of boxes into regular container boxes (i.e., pallet or lorry loading), although the packing of irregular three-dimensional forms has also been attempted (Stoyan et al. 2002). However, this paper specifically addresses the two-dimensional orthogonal stock-cutting problem that also has applications to dynamic memory allocation, multiprocessor scheduling problems, and other general layout problems (Paull 1956, Coffman et al. 1978, Garey and Johnson 1981, Coffman and Leighton 1989).

The orthogonal stock-cutting problem involves the placement of rectangular objects onto a predefined rectangular sheet. In general, the overall objective is to produce as densely packed solutions as possible to maximise material utilisation and therefore increase profits. Automated approaches have utilised many different techniques that include linear and dynamic programming (Gilmore and Gomory 1961), heuristic methods (Baker et al. 1980, Burke et al. 2004), artificial neural networks (Dagli and Poshyanonda 1997), and metaheuristic approaches (Smith 1985, Burke and Kendall 1999, Valenzuela and Wang 2001, Hopper and Turton 2001, Iori et al. 2003) including simulated annealing (Dagli and Hajakbari 1990). It has been shown that automated solutions, produced with these strategies, can often be better than those of their

skilled human counterparts both in terms of packing quality and in the time taken to achieve solutions (Roberts 1984, Li and Milenkovic 1995, Hower et al. 1996).

The two-dimensional variant of the stock-cutting problem that will be addressed in this paper uses a stock sheet of fixed width but no fixed length (equivalent to using a roll of material) and has the following constraints: All of the rectangular objects must be placed within the confines of the stock sheet, must not overlap with each other, and may only be rotated by 90°. It should also be noted that we do not restrict solutions to guillotinable packings (where only horizontal and vertical cuts that entirely span the stock sheet are allowed). The objective is to place all rectangular items so that the overall length required is minimised and that no constraints are violated. This instance of the stock-cutting problem has been shown to be NP-hard (Garey and Johnson 1979).

2. Simulated Annealing Enhancement for the Best-Fit Heuristic

In this section, we present a description of our new approach and discuss the strategy that we use. We begin with a broad overview of the approach that develops our previous work on the best-fit heuristic (Burke et al. 2004). We use the bottom-left-fill heuristic in conjunction with simulated annealing by drawing upon work proposed by Hopper and Turton (2001). Finally, we describe how these *elements* can be combined to produce a strategy that outperforms other approaches from the literature.

2.1. Overview of the Hybrid Method

The method that we describe in this section utilises the best-fit heuristic (Burke et al. 2004) and the simulated annealing hybridised bottom-left-fill heuristic proposed by Hopper and Turton (2001). A solution is generated by a two-step process that, first of all, applies the best-fit heuristic to place an initial number of rectangles onto the stock sheet. Then, the remaining unassigned rectangles are placed using the bottom-left-fill heuristic with different input orderings as guided by the simulated annealing search procedure. The locations of the rectangles that are assigned to the sheet in the initial best-fit stage of the process remain fixed throughout and form the first part of the solution, whereas the rectangles placed during the second stage will move locations according to the orderings produced by simulated annealing.

In this paper, we use the best-fit heuristic as the initial step for our hybrid approach. Unlike Burke et al. (2004), we now use floating point data. This makes the heuristic up to five times slower than an integer representation implementation. However, we can justify this modification by the extra accuracy on the floating

point data sets (as used in some benchmark problems in this paper) and also because the execution times are still less than one second for all but the largest of our problems.

2.2. The Simulated Annealing Bottom-Left-Fill Hybridised Method

We hybridise the bottom-left-fill heuristic by using simulated annealing to produce orderings that are “decoded” into a solution and evaluated using the bottom-left-fill heuristic. To obtain input sequences, we assign a numerical identification to each rectangle. Therefore, a sequence such as 4, 2, 5, 1, 3 describes the solution obtained by the bottom-left-fill heuristic packing the fourth rectangle first, the second rectangle second, and so forth. In the problem studied in this paper, we allow rotations of 90° and encode rectangle orientation by using a Boolean flag for each rectangle within a solution sequence. Because of the experiments of Hopper and Turton (2001), we seed our simulated annealing approach by using a decreasing height sequence.

In preliminary experiments, we hybridised the bottom-left-fill algorithm with tabu search, simulated annealing, and a genetic algorithm. Although all of these performed well, the tabu search and the genetic algorithm were less competitive than simulated annealing. Therefore, this paper will specifically focus on the experimentation using simulated annealing only. The interested reader can find a more detailed description of all the experiments in the Online Supplement (available at <http://joc.pubs.informs.org/ecompanion.html>).

2.2.1. Simulated Annealing Parameters. Our simulated annealing cooling scheme is reliant on the progress of the search. We accept worse solutions with a probability of $P(\text{accept}) = 1 - (\Delta E/T)$, where ΔE is the change in solution evaluation quality from the current solution and T is the temperature. Initially, we set $T = 3$. This value was established after a series of experiments. Better solutions are always accepted. The cooling operates after every iteration based on whether we accept a neighbouring solution. If we accept, then we geometrically cool the temperature using a multiplier of 0.999. However, if the neighbouring solution is not accepted, then we heat the temperature by a multiplier of 1.0001. These values were chosen through initial testing and have the effect of cooling quickly if we are accepting many solutions (high temperature) and cooling more slowly when fewer solutions are accepted. As the search progresses, the temperature may become stable due to the rejection of more solutions, thus generating a more frequent heating process. However, when the search moves to explore a different area of the search space, the temperature may increase or

decrease to reflect the acceptance occurring within this new region. We find a neighbouring solution by a random swap and an orientation flip with a probability of 50% of the first random rectangle chosen. The process is summarised as follows: Find one neighbour of the current solution, evaluate the neighbour using the bottom-left-fill heuristic, and find the change in solution quality. If we accept the neighbour, we store it as the current solution and cool. If we reject the neighbour, we keep the current solution and heat. Once again, the initial solution supplied is that of the input rectangles sorted by decreasing height and the best solution seen during the search is returned at the end. Simulated annealing techniques have also been applied to the orthogonal stock-cutting problem in Lai and Chan (1996).

2.3. The Proposed Hybrid Packing Strategy

We presented a brief overview of the hybrid strategy in §2.1. In this section, we present a more detailed description of the approach and discuss its motivation. In Burke et al. (2004), we compare the best-fit heuristic to the previous best-performing approaches in the literature. These were bottom-left-based metaheuristic methods (Jakobs 1996; Hopper and Turton 1999, 2001; Valenzuela and Wang 2001). In the comparison, we showed that the best-fit heuristic is able to consistently outperform these approaches when used on problems of medium to very large sizes and that it is able to achieve solutions that can be within 2% of optimal. However, the best-fit heuristic obtains worse solutions for smaller problems (i.e., those involving less than 50 shapes). Motivated by these observations, we developed a packing strategy that can further improve upon the best-fit heuristic by utilising the respective strengths of both the best-fit heuristic and metaheuristic bottom-left methods. We summarise the advantages and disadvantages of the two strategies in Table 1.

As mentioned above, we experimented with three different metaheuristics, but we just present the simulated annealing method here because it was the most effective.

Given an orthogonal stock-cutting problem involving n rectangles, we propose a two-phase approach to

generate solutions. In the first phase, we invoke the best-fit heuristic. We allow best fit to pack $n - m$ rectangles, where m is the number of rectangles that are to be carried over to phase 2. Phase 2 involves the packing of the remaining rectangles using simulated annealing hybridised with the bottom-left-fill heuristic for a suitable duration of time. At the commencement of phase 2, it should be noted that we already have the foundations of a solution because the best-fit heuristic (phase 1) has already placed a number of the rectangles on the sheet. These rectangles will remain unchanged during the guided search procedure of phase 2.

The reasoning for this approach can be seen in Table 1. Phase 1 aims to utilise best fit's ability to produce high-quality solutions in fast execution times. The termination of the best-fit process, before all rectangles are packed, aims to reduce the untidy arrangements that usually occur during the latter stages of best-fit solutions due to the smaller number of rectangles that are available for selection. We allow a small subproblem to be passed to the simulated annealing method of phase 2, thus limiting the difficulties that these methods have with excessively large search spaces and enabling a good phase 2 partial solution to be found relatively quickly. This is then concatenated to the partial solution of phase 1 to create our overall solution. Although other factors may also be important, such as sheet width or relative size of the problem shapes, we can see how the proposed model aims to promote the advantages of both methods while inhibiting their disadvantages. The transition from one phase to the other is guided by the m variable. If $m = 0$, then all rectangles are packed using phase 1, the best-fit heuristic (Burke et al. 2004). However, if $m \geq n$, then we only use phase 2, and the approach can therefore be considered as using only the simulated annealing bottom-left-fill strategy. From the observations of Burke et al. (2004), we can make the following prediction: If $m > 50$, then we will probably obtain a worse solution than if we used the stand-alone best-fit heuristic; therefore, an effective value for m may be somewhere in the range $0 < m < 50$. Our experiments varied the value of the phase switch variable m to test our assumption that

Table 1 Advantages and Disadvantages of the Best-Fit and Metaheuristic Bottom-Left Heuristics

Best-fit heuristic (phase 1)		Metaheuristic bottom-left methods (phase 2)	
Advantages	Disadvantages	Advantages	Disadvantages
Produces best published results on a range of medium/large standard benchmark problems	Worse solutions found when problem size < 50	Optimal solutions found on small problems	Does not represent the best approaches on medium/large problems
Fast solutions obtained $\approx 1,000$ rectangles/second	Untidy arrangements at end of packing	Can execute for as long as user allows	Similar input sequences may produce vastly differing solutions
No parameters			Definition of search parameters is required

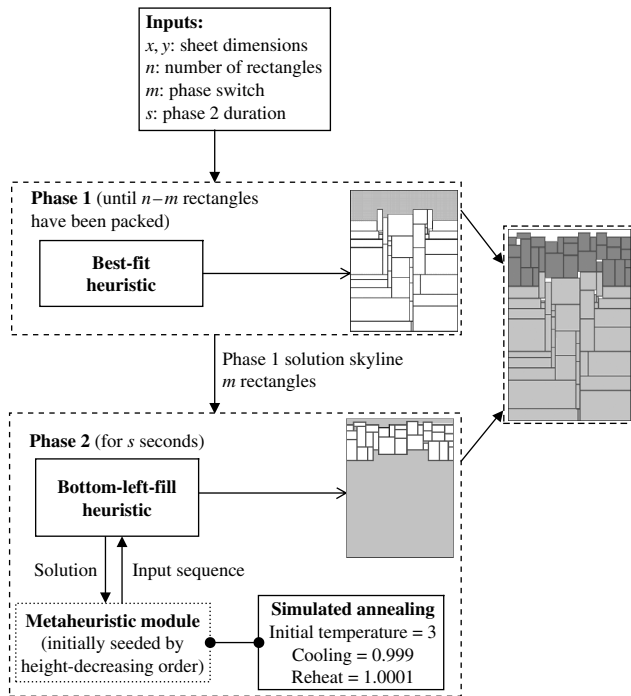


Figure 1 A Summary of the Proposed Strategy

a value in the range $0 < m < 50$ should produce better results. The conclusion is that the best setting for m is between 20 and 40 using 60-second durations. Because of these results, we set m to 30 for the experiments of §4.1. A summary of our proposed approach is shown in Figure 1.

3. Benchmark Test Data

To evaluate our proposed approach, we use a wide range of benchmark test problems from the literature from four different sources (see Table 2). This mirrors the data used in Burke et al. (2004) so as to provide consistent comparisons. Hopper and Turton (2001) provide seven categories of problem with each category containing three instances of similar size. Valenzuela and Wang (2001) provide floating point data sets of both similarly dimensioned rectangles (named “nice” data) and vastly differing dimensions (named “path” data). Although these problems are designed for the guillotine variant of the problem, they provide floating point data that can demonstrate our method’s ability to handle such problems without inaccuracy and also provide further benchmarking opportunities. Each category has data ranging from 25 to 1,000 rectangles (Valenzuela and Wang 2001). Ramesh Babu and Ramesh Babu (1999) use a test problem to compare their genetic algorithm-based method against the method proposed by Jakobs (1996). We also use 13 problems presented in Burke et al. (2004) that contain few large rectangles and many smaller rectangles. These problems have been generated through the subdivision of one large “solution” rectangle to create the individual rectangular pieces (this is often known as a “perfect packing”). Therefore, the optimal solution is known.

We also provide 10 new test problems to promote further comparison between future packing methods. For the first five of these problems, MT1–MT5, the

Table 2 Test Data from the Literature

Data source	Problem category	Test problems given	No. of rectangles	Optimal height	Object dimensions
Hopper and Turton (2001)	C1	P1, P2, P3	16 or 17	20	20 × 20
	C2	P1, P2, P3	25	15	40 × 15
	C3	P1, P2, P3	28 or 29	30	60 × 30
	C4	P1, P2, P3	49	60	60 × 60
	C5	P1, P2, P3	72 or 73	90	60 × 90
	C6	P1, P2, P3	97	120	80 × 120
	C7	P1, P2, P3	196 or 197	240	160 × 240
Valenzuela and Wang (2001)	Nice	P1, P2, P3, P4, P5, P6	25, 50, 100, 200, 500, 1,000	100	100 × 100
	Path	P1, P2, P3, P4, P5, P6	25, 50, 100, 200, 500, 1,000	100	100 × 100
Ramesh Babu and Ramesh Babu (1999)		P1	50	375	Width = 1,000
Burke et al. (2004)		N1	10	40	40 × 40
		N2	20	50	30 × 50
		N3	30	50	30 × 50
		N4	40	80	80 × 80
		N5	50	100	100 × 100
		N6	60	100	50 × 100
		N7	70	100	80 × 100
		N8	80	80	100 × 80
		N9	100	150	50 × 150
		N10	200	150	70 × 200
		N11	300	150	70 × 200
		N12	500	300	100 × 300
		N13	3,152	960	640 × 960

Table 3 New Test Data with Unknown Optimal Heights

New test problem	No. of rectangles	Lower-bound height	Object dimensions
MT1	100	492.63	400 × 600
MT2	150	760.90	300 × 900
MT3	200	767.31	400 × 900
MT4	300	1,174.94	600 × 1,300
MT5	500	1,264.86	800 × 1,400

optimal solution is not known (a lower-bound height is calculated by dividing the total area of the rectangles by the sheet width). The last five problems, MT5–MT10, have been created through the dissection of a larger rectangular object, so for these problems the optimal height is known. Table 3 summarises the first five data sets, MT1–MT5, for which the optimal packing heights are unknown. The new data sets, for which optimal solutions are known, are given in Table 4. The data for all of these problems can be found in the Online Supplement.

4. Experimentation and Results

In this section, we compare the results obtained using the proposed method with the results from previous work. To make this comparison, we have opted for a dual-evaluation approach where we use the height of the solution obtained and the secondary measure of the total area underneath the skyline. The main objective is to minimise the height of the packing, but as several different arrangements may have identical heights, the area under the skyline can also be important because it quantifies the amount of wasted space. Two example solutions having identical heights but differing skyline areas are shown in Figure 2. The solution on the right of Figure 2 is more desirable as more of the sheet is usable for future cutting processes.

Therefore, based on these observations, the following evaluation function has been defined:

Solution Evaluation

$$= \text{Total Height} + (\text{Area Under Skyline} / \text{Sheet Width}).$$

Although both solutions of Figure 2 require the same total height, the left-hand solution has a large amount of wasted space and therefore achieves an evaluation

Table 4 New Test Data with Known Optimal Heights

New test problem	No. of rectangles	Optimal height	Object dimensions
MT6	100	500	400 × 500
MT7	150	500	400 × 500
MT8	200	800	500 × 800
MT9	300	800	600 × 800
MT10	500	800	700 × 800

value of $36 + (2,156/60) = 71.9$. The right-hand solution has no wasted areas and thus obtains a far better evaluation of $36 + (1,800/60) = 66$. For problems that have been created from the dissection of a larger rectangle (where the optimal solution is known), the optimal evaluation value is twice that of the optimal height. All of the techniques discussed in this paper use this evaluation function for comparisons where lower evaluation values indicate better solutions. All experiments have been conducted on a 2 GHz Pentium 4 computer with 256 MB RAM.

4.1. Evaluation of the Hybrid Simulated Annealing Approach on Test Problems from the Literature

Our experiments aimed to compare our proposed simulated annealing hybrid method to other approaches from the literature, including the stand-alone best-fit heuristic as proposed in Burke et al. (2004) and the metaheuristic bottom-left-fill methods proposed in Hopper and Turton (2001). The best and average solution of 10 runs is given for all the problems from the literature in Table 5. The best solution for each method is given in the final two columns as a percentage over optimal, and the best-performing method is highlighted in bold.

Table 5 shows that of the 47 problems from the literature, our proposed hybrid method can significantly improve upon 41 of these (approximately 87%). Of the problems not improved upon by phase 2 of our approach, one problem achieves a very good solution by using best fit alone and can only be beaten by achieving the optimal solution (C2P2).

For some other problems, there are very small rectangles at the top of the stock sheets that consume all of the 30 rectangles, which are operated on by the simulated annealing second phase, and therefore there is not enough scope for improvement (e.g., N10, N11, N12, N13). For these problems, we can increase the phase switch variable, but this will also require longer execution times to achieve suitable exploration and exploitation of the search space. We extend our experimentation for these problems in §4.2.

Table 6 compares the average solutions achieved by the new hybrid simulated annealing algorithm to one of the best-performing approaches in the literature, the metaheuristic bottom-left-fill algorithm (our phase 2 method), which was proposed by Hopper and Turton (2001). The best average result is shown in bold. In Table 6, we present two different versions of the Hopper and Turton (2001) algorithm (using tabu search and simulated annealing). We have also used shading in Table 6 to indicate all of the problem instances involving fewer than 30 rectangles, because for these problems, both algorithms will perform equally. Our hybrid approach will not use the best-fit heuristic (phase 1) because we have fixed

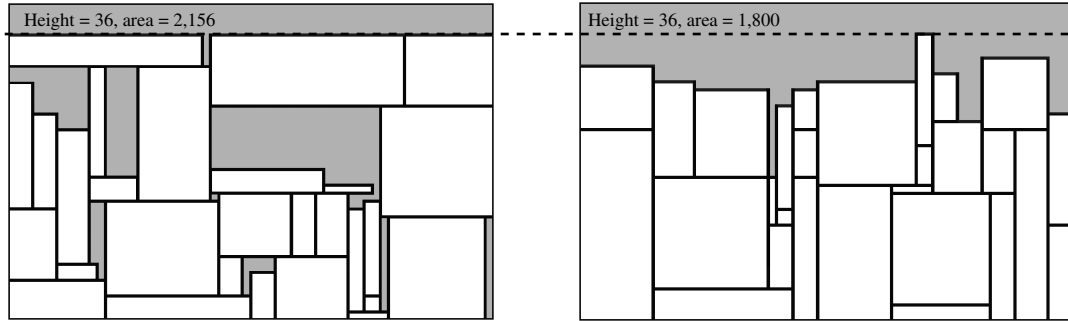


Figure 2 Two Solutions of Identical Total Height but Vastly Different Skyline Area Evaluations

the phase switch variable m at value 30 (see above). Indeed, this is evident from Table 6 because both algorithms achieve similar average solutions. The most interesting problems occur when there are greater than 30 rectangles, which allows the proposed hybrid approach to utilise the respective advantages of both phases. In most of the data sets, our approach obtains much better average solutions than the metaheuristic bottom-left-fill algorithms on problems with more than 30 rectangles. In the final data set of benchmark problems (Burke et al. 2004), the proposed algorithm achieves improved results on over half of the problems and is competitive on the remaining problems. However, the two metaheuristic bottom-left-fill approaches compare better on this set of problems. The reason for this is due to the nature of the data. Each of these problems only involves relatively few large rectangles and many smaller rectangles that can be used to fill holes. Therefore, it can place the larger rectangles in any orientation and in potentially poor layouts, but because of the vast number of smaller rectangles, the holes can easily be filled to produce good layouts. The first three data sets (Hopper and Turton 2001, Valenzuela and Wang 2001, Ramesh Babu and Ramesh Babu 1999) do not share this property, so the metaheuristic bottom-left-fill algorithm cannot perform as well compared with the proposed approach that utilises best fit to lay the foundations of a good solution. We have shown that our simulated annealing hybrid method can achieve better solutions than best fit alone and metaheuristic bottom-left-fill alone on most problems with only 60 seconds of additional execution time.

As an example, Figure 3 shows the best solution achieved by both the best-fit heuristic alone and the proposed hybrid method on the data set C5P1 from Hopper and Turton (2001). From Figure 3(a), it can be seen that by applying the best-fit heuristic alone, a very good overall solution can be generated (three units above optimal). Also, it is apparent that the heuristic produces very dense packings at the bottom of the sheet. However, at the top of the packing, there are fewer rectangles for the heuristic to select

from, and therefore the arrangement begins to suffer from holes and thus wastage. Figure 3(b) shows the results of applying the simulated annealing bottom-left-fill method to the last 30 rectangles of the initial packing given by the best-fit heuristic. In less than 60 seconds, the simulated annealing method can improve upon the best-fit heuristic's attempt at packing the last 30 rectangles both in terms of total packing height and sheet area required. The proposed simulated annealing hybridised method yielded a solution height of 91 units for this problem, which is only one unit above the "perfect packing" (optimal) solution of 90 units.

4.2. Extended Experiments on Benchmark Problems from the Literature

In §4.1, we produced new best solutions for the majority of the literature benchmark problems. In this section, we extend our experiments for all of the problems for which the simulated annealing bottom-left-fill method was unable to improve upon the best-fit heuristic's results. We conduct experiments using a phase switch value of the last 150 rectangles and allow five minutes' execution time. In allowing more rectangles into the second phase, each iteration takes longer due to the extra collision detection required. The best and average solutions of 10 runs for each of these extended experiments are shown in Table 7 (best results are in bold).

From Table 7, the proposed approach (with altered parameters) improved over the stand-alone best-fit heuristic on four of the five problems. In §4.1, we discussed the solution given for problem C2P2. With the new parameters, we can see that we were able to find the optimal solution for this problem by allowing more time to explore the search space. It should be noted that only phase 2 is active in this problem because the problem size is smaller than m .

In §4.1, we observed that the problems N10–N13 had the property that small rectangles are passed to phase 2 of our proposed simulated annealing method, and thus the approach did not appear to have enough scope for improvement. However, increasing the value of the phase switch variable has created

Table 5 Comparison of the Proposed Hybrid Approach to the Best-Fit Heuristic Using Test Problems from the Literature

Data set	Cat.	Prob.	No. of shapes	Opt. height	Opt. eval.	Solution obtained			BF and simulated annealing (SA) bottom-left fill			Best sol. (% over optimal)		
						Height	Area	Eval.	Height	Area	Eval.	Avg. eval	BF	BF + SA
Hopper and Turton (2001)	C1	P1	16	20	40	21	410	41.50	20	400	40.00	40.00	3.75	0.00
		P2	17	20	40	22	400	42.00	20	400	40.00	40.50	5.00	0.00
		P3	16	20	40	24	400	44.00	20	400	40.00	40.00	10.00	0.00
	C2	P1	25	15	30	16	609	31.23	16	600	31.00	31.00	4.08	3.33
		P2	25	15	30	16	600	31.00	16	600	31.00	31.00	3.33	3.33
		P3	25	15	30	16	604	31.10	16	600	31.00	31.00	3.67	3.33
	C3	P1	28	30	60	32	1,814	62.23	31	1,818	61.30	61.59	3.72	2.17
		P2	29	30	60	34	1,801	64.02	31	1,814	61.72	61.72	6.69	2.06
		P3	28	30	60	33	1,806	63.10	31	1,808	61.23	61.46	5.17	1.89
C4	P1	49	60	120	63	3,623	123.38	61	3,621	121.35	122.08	2.82	1.13	
	P2	49	60	120	62	3,621	122.35	61	3,618	121.30	122.20	1.96	1.08	
	P3	49	60	120	62	3,621	122.35	61	3,608	121.13	121.53	1.96	0.94	
C5	P1	72	90	180	93	5,444	183.73	91	5,419	181.32	182.33	2.07	0.73	
	P2	73	90	180	92	5,413	182.22	91	5,410	181.17	182.18	1.23	0.65	
	P3	72	90	180	93	5,469	184.15	92	5,438	182.63	182.79	2.31	1.46	
C6	P1	97	120	240	123	9,681	244.01	122	9,652	242.65	242.80	1.67	1.10	
	P2	97	120	240	122	9,632	242.40	121	9,632	241.40	241.42	1.00	0.58	
	P3	97	120	240	124	9,664	244.80	122	9,647	242.59	242.63	2.00	1.08	
C7	P1	196	240	480	246	38,899	489.12	244	38,839	486.74	487.75	1.90	1.40	
	P2	197	240	480	244	38,715	485.97	244	38,667	485.67	487.02	1.24	1.18	
	P3	196	240	480	245	38,864	487.90	245	38,850	487.81	488.17	1.65	1.63	
Valenzuela and Wang (2001)	Nice	P1	25	100	200	107.99	10,267.7	210.67	104.024	10,049.5	204.52	207.25	5.33	2.26
		P2	50	100	200	109.69	10,423.8	213.93	104.413	10,221.0	206.62	208.24	6.97	3.31
		P3	100	100	200	107.92	10,273.2	210.65	105.04	10,335.0	208.39	208.69	5.33	4.19
	Path	P4	200	100	200	106.89	10,274.1	209.63	104.743	10,266.7	207.41	207.59	4.82	3.71
		P5	500	100	200	103.39	10,195.4	205.35	103.507	10,199.3	205.50	206.01	2.67	2.75
		P6	1,000	100	200	103.84	10,267.6	206.52	103.756	10,269.8	206.45	206.72	3.26	3.23
Ramesh Babu and Ramesh Babu (1999)	Path	P1	25	100	200	110.19	10,253.7	212.73	103.09	10,038.3	203.47	207.21	6.36	1.74
		P2	50	100	200	113.65	10,285.1	216.50	103.4	10,060.9	204.01	204.59	8.25	2.00
		P3	100	100	200	106.75	10,180.6	208.56	103.037	10,125.7	204.29	205.92	4.28	2.15
	Ramesh Babu and Ramesh Babu (1999)	P4	200	100	200	104.12	10,157.4	205.70	103.419	10,116.1	204.58	204.63	2.85	2.29
		P5	500	100	200	103.81	10,189.0	205.70	103.479	10,200.1	205.48	205.60	2.85	2.74
		P6	1,000	100	200	103.09	10,173.0	204.82	102.867	10,165.0	204.52	204.68	2.41	2.26
Burke et al. (2004)	P1	50	375	750	400	378,125	778.13	400	367,875	767.88	767.88	3.75	2.38	
	N1	10	40	80	45	1,800	90.00	40	1,600	80.00	80.00	12.50	0.00	
	N2	20	50	100	53	1,542	104.40	50	1,500	100.00	101.68	4.40	0.00	
	N3	30	50	100	52	1,525	102.83	51	1,500	101.00	101.32	2.83	1.00	
	N4	40	80	160	86	6,547	167.84	82	6,455	162.69	163.50	4.90	1.68	
	N5	50	100	200	105	10,154	206.54	103	10,122	204.22	204.29	3.27	2.11	
	N6	60	100	200	102	5,061	203.22	102	5,047	202.94	202.94	1.61	1.47	
	N7	70	100	200	108	8,230	210.88	104	8,117	205.46	205.97	5.44	2.73	
	N8	80	80	160	83	8,116	164.16	82	8,097	162.97	162.97	2.60	1.86	
	N9	100	150	300	152	7,556	303.12	152	7,544	302.88	302.88	1.04	0.96	
	N10	200	150	300	152	10,567	302.96	152	10,570	303.00	303.00	0.99	1.00	
	N11	300	150	300	153	10,586	304.23	153	10,586	304.23	304.23	1.41	1.41	
	N12	500	300	600	306	30,431	610.31	306	30,431	610.31	610.31	1.72	1.72	
N13	3,152	960	1,920	964	615,637	1,925.9	964	615,629	1,925.9	1,925.9	0.31	0.31		

Table 6 Comparison of the Proposed Hybrid Approach to Stand-Alone Metaheuristic Bottom-Left-Fill Algorithms

Data set	Metaheuristic bottom-left fill (proposed by Hopper and Turton 2001)																	
	Tabu search						Simulated annealing						Proposed best fit + simulated annealing bottom-left fill					
	Cat.	Prob.	No. of shapes	Opt. height	Opt. eval	Height	Best solution			Avg. eval.	Height	Best solution			Avg. eval.			
							Area	Eval.	Area			Eval.	Area	Eval.		Area	Eval.	
Hopper and Turton (2001)	C1	P1	16	20	40	20	400	40.00	40.82	20	20	400	40.00	40.00	20	400	40.00	40.00
		P2	17	20	40	21	400	41.00	41.00	20	20	400	40.00	40.73	20	400	40.00	40.00
		P3	16	20	40	20	400	40.00	40.50	20	20	400	40.00	40.10	20	400	40.00	40.00
	C2	P1	25	15	30	16	600	31.00	31.00	15	15	600	30.00	30.90	16	600	31.00	31.00
		P2	25	15	30	16	600	31.00	31.00	16	16	600	30.00	31.00	16	600	31.00	31.00
		P3	25	15	30	16	600	31.00	31.00	16	16	600	30.00	31.00	16	600	31.00	31.00
C3	P1	28	30	60	31	1,810	61.17	61.85	31	31	1,800	61.00	61.36	31	1,818	61.30	61.59	
	P2	28	30	60	32	1,800	62.00	62.20	31	31	1,801	61.02	61.63	31	1,801	61.02	61.59	
	P3	28	30	60	31	1,822	61.37	61.96	31	31	1,810	61.17	61.52	31	1,808	61.13	61.46	
	P1	49	60	120	62	3,617	122.28	122.91	62	62	3,643	122.72	123.92	61	3,621	121.35	122.08	
	P2	49	60	120	62	3,617	122.28	122.33	61	62	3,620	121.33	122.66	61	3,618	121.30	122.20	
	P3	49	60	120	62	3,612	122.20	122.32	61	61	3,619	122.32	122.43	61	3,608	121.13	121.53	
C5	P1	72	90	180	92	5,415	182.25	183.13	92	92	5,462	183.03	183.84	91	5,419	181.32	182.33	
	P2	73	90	180	92	5,407	182.12	182.48	91	91	5,422	181.37	183.24	91	5,410	181.17	182.18	
	P3	72	90	180	92	5,431	182.52	183.02	93	93	5,460	184.00	184.12	92	5,438	182.63	182.79	
	P1	240	120	240	122	9,695	243.19	243.75	122	122	9,694	243.18	244.32	122	9,652	242.65	242.80	
	P2	240	120	240	122	9,635	242.44	242.99	122	122	9,668	242.85	244.20	121	9,632	241.40	241.42	
	P3	240	120	240	122	9,726	243.58	244.14	123	123	9,703	244.30	244.52	122	9,647	242.59	242.63	
C7	P1	196	240	480	246	38,741	488.13	488.47	245	245	38,915	488.22	489.39	244	38,839	486.74	487.75	
	P2	197	240	480	244	38,767	486.29	487.12	246	246	38,873	487.72	488.13	244	38,667	485.67	487.02	
	P3	196	240	480	245	38,864	487.90	488.40	245	245	38,899	488.12	488.52	245	38,850	487.81	488.17	
	P1	25	100	200	106.63	10,230.3	208.93	209.83	104.11	104.11	10,049.5	204.61	207.31	104.024	10,049.5	204.52	207.25	
	P2	50	100	200	109.13	10,400.8	213.14	213.77	109.411	109.411	10,392.2	213.33	214.50	104.413	10,221.0	206.62	208.24	
	P3	100	100	200	108.83	10,589.4	214.72	215.71	109.21	109.21	10,614.8	215.36	216.79	105.04	10,335.0	208.39	208.69	
Path	P4	200	100	200	110.93	10,836.1	219.29	220.06	111.45	111.45	10,787.1	219.32	219.67	104.743	10,266.7	207.41	207.59	
	P5	500	100	200	110.45	10,145	218.63	219.45	110.68	110.68	10,736.5	N/A	218.45	103.507	10,199.3	205.50	206.01	
	P6	1,000	100	200	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	103.756	10,269.8	206.45	206.72	
	P1	25	100	200	106.22	10,302.2	209.24	209.99	101.82	101.82	10,030.6	202.13	205.61	103.09	10,038.3	203.47	207.21	
	P2	50	100	200	105.01	10,046.9	205.48	206.06	105.01	105.01	10,072.8	205.74	206.32	103.4	10,060.9	204.01	204.59	
	P3	100	100	200	104.03	10,162.0	205.65	205.89	104.417	104.417	10,216.5	206.58	208.03	103.037	10,125.7	204.29	204.92	
Ramesh Babu and Ramesh Babu (1999)	P4	200	100	200	104.37	10,210.2	206.47	206.84	104.821	104.821	10,236.0	207.18	207.67	103.419	10,116.1	204.58	204.63	
	P5	500	100	200	106.71	10,438.3	211.10	211.25	105.914	105.914	10,452.5	210.44	211.49	103.479	10,200.1	205.48	205.60	
	P6	1,000	100	200	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	102.867	10,165.0	204.52	204.68	
	P1	50	375	750	400	375,000	775.00	775.00	400	400	375,000	775.00	775.00	400	376,875	776.88	776.88	
	Burke et al. (2004)	N1	10	40	80	80	1,600	80.00	80.00	40	40	1,600	80.00	80.00	40	1,600	80.00	80.00
		N2	30	50	100	100	1,500	100.00	101.49	50	50	1,500	100.00	101.42	50	1,500	100.00	101.68
		N3	20	50	100	100	1,501	101.03	101.36	51	51	1,500	101.00	101.34	51	1,500	101.00	101.32
		N4	40	80	160	160	6,400	162.00	163.12	81	81	6,403	161.04	162.98	82	6,455	162.69	163.50
		N5	50	100	200	200	10,147	205.47	206.21	104	104	10,071	204.71	205.56	103	10,122	204.22	204.29
		N6	60	100	200	200	5,039	202.78	203.10	102	102	5,016	202.24	203.36	102	5,047	202.94	202.94
		N7	70	100	200	200	8,009	201.11	203.34	102	102	8,022	202.18	203.12	104	8,117	205.46	205.97
		N8	80	80	160	160	162.61	163.02	163.02	82	82	162.57	162.57	163.41	82	162.97	162.97	162.97
		N9	100	150	300	300	7,556	303.12	304.14	152	152	7,557	304.14	304.80	152	7,544	302.88	317.88
N10		200	150	300	300	10,582	303.14	303.46	153	153	10,595	304.36	304.62	152	10,570	303.00	303.00	
N11		300	150	300	300	10,583	303.19	304.18	153	153	10,582	304.71	305.81	153	10,586	304.23	304.23	
N12		500	300	600	600	30,524	612.24	612.51	306	306	30,508	611.08	612.56	306	30,431	610.31	610.31	
N13		3,152	960	1,920	1,920	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	615,629	1,925.9	1,925.9	

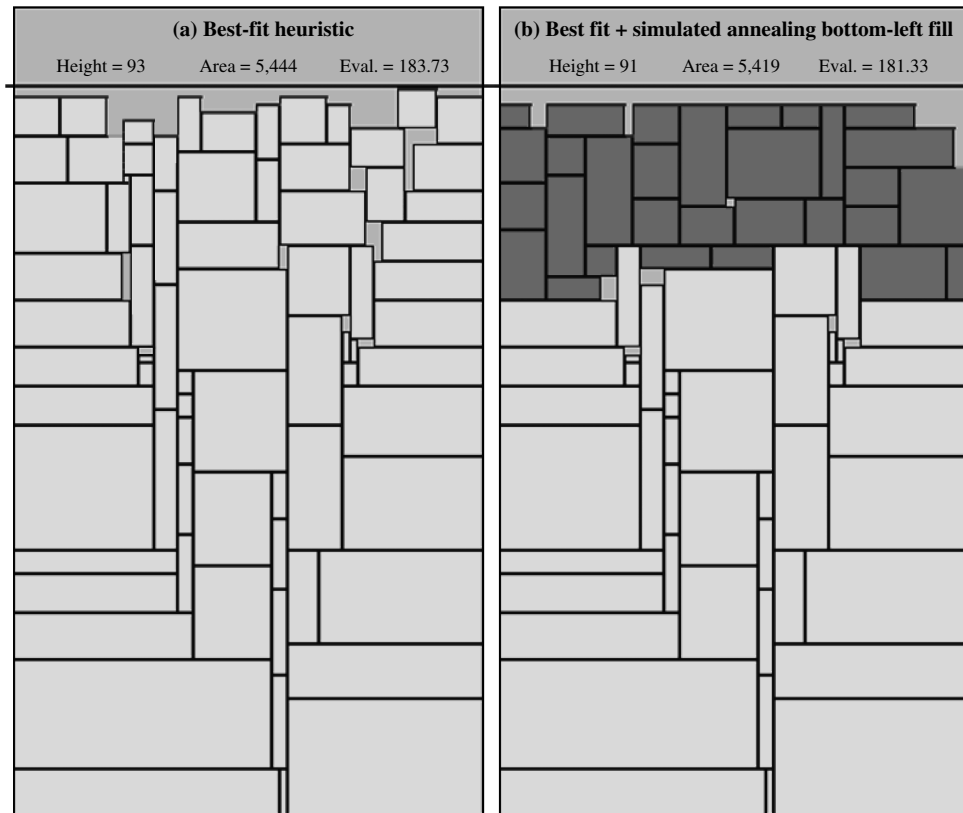


Figure 3 Solutions Obtained by (a) the Solitary Best-Fit Heuristic and (b) the Best-Fit Heuristic Hybridised with Simulated Annealing Bottom-Left Fill on Problem C5P1

enough scope for improvement on three of these problems when using our proposed hybrid approach. For example, the height of the solution obtained for N10 is only one unit above optimal. We were unable to improve upon the largest of our problems, N13. Further improvements may be found by extending the experiments in a similar manner, but this is ultimately a time-versus-quality decision that would depend on the user’s requirements.

4.3. Experiments on New Test Problems

In our fourth set of experiments, we further analyse the improvements that the proposed hybrid method can achieve with larger periods of time and by

varying the value of m . We generate solutions for the 10 new test problems given in Tables 3 and 4 using the proposed approach with values of 30, 50, and 100 rectangles for the phase switch variable m . To compare how variations of computational time affect the results, we use 60-, 300-, and 600-second phase 2 durations. Table 8 displays the results with the best result for each of the durations highlighted (black text on grey shading) and the best evaluation average for each data set being highlighted (white text on black). This identifies the phase switch value that performed the best under each phase 2 time duration.

Table 8 shows that allowing longer time durations in general will produce better average solutions given

Table 7 Results of Extended Experiments Using $m = 150$ and Run Times of Five Minutes

Test data set	Size	Best-fit heuristic			Best fit + simulated annealing bottom-left fill ($m = 150$, 5-min. run)			
		Best solution			Best solution			
		Height	Area	Eval.	Height	Area	Eval.	Avg. eval.
C2P2	25	16	600	31.00	15	600	30.00	30.80
N10	200	152	10,567	302.96	152	10,551	302.73	302.77
N11	300	153	10,586	304.23	152	10,581	303.16	303.20
N12	500	306	30,431	610.31	304	30,288	606.88	607.00
N13	3,152	964	615,637	1,925.9	964	615,637	1,925.9	1,925.9

Table 8 Best and Average Results Using Phase 2 Durations of 60, 300, and 600 Seconds on the New Problems

Data set	No. of shapes	Phase switch m	Best fit + simulated annealing bottom-left fill											
			60-second run				300-second run				600-second run			
			Best solution				Best solution				Best solution			
			Height	Area	Eval.	Avg. eval.	Height	Area	Eval.	Avg. eval.	Height	Area	Eval.	Avg. eval.
MT1	100	30	503.08	198,729.6	999.90	1,001.28	502.78	198,626.9	999.34	1,001.36	503.08	198,574.8	999.51	1,000.79
		50	501.38	199,053.4	999.01	1,001.31	502.50	199,153.5	1,000.38	1,001.49	502.50	198,559.3	998.90	999.85
		100	504.50	199,437.8	1,003.09	1,006.92	504.50	199,024.8	1,002.06	1,004.70	502.88	199,352.5	1,001.26	1,002.63
MT2	150	30	775.20	231,146.9	1,545.69	1,549.14	773.70	231,126.6	1,544.12	1,544.80	774.00	231,059.3	1,544.20	1,544.50
		50	774.00	230,946.8	1,543.82	1,548.40	773.20	230,840.4	1,542.67	1,545.88	772.80	230,876.5	1,542.39	1,545.30
		100	781.50	232,762.1	1,557.37	1,560.56	780.90	231,832.8	1,553.68	1,558.23	779.70	232,071.8	1,553.27	1,557.91
MT3	200	30	793.20	313,991.5	1,578.18	1,582.21	793.00	313,508.1	1,576.77	1,579.05	794.10	313,836.8	1,578.69	1,579.80
		50	794.00	313,337.4	1,577.34	1,581.23	796.00	313,531.9	1,579.83	1,582.36	794.00	313,368.5	1,577.42	1,581.48
		100	793.00	313,505.3	1,576.76	1,580.67	790.60	312,996.5	1,573.09	1,578.56	790.30	312,690.7	1,572.03	1,575.14
MT4	300	30	1,199.00	714,773.0	2,390.29	2,390.97	1,198.00	715,312.0	2,390.19	2,390.45	1,198.00	715,012.0	2,389.69	2,390.21
		50	1,200.00	715,022.0	2,391.70	2,392.79	1,197.00	714,443.0	2,387.74	2,390.25	1,197.00	715,158.0	2,388.93	2,390.13
		100	1,211.00	720,719.0	2,412.20	2,414.83	1,198.00	715,015.0	2,389.69	2,392.50	1,199.00	714,488.0	2,389.81	2,391.65
MT5	500	30	1,286.60	1,020,656.6	2,562.42	2,563.63	1,285.60	1,020,681.0	2,561.45	2,561.83	1,286.60	1,020,261.4	2,561.93	2,563.60
		50	1,283.10	1,021,290.7	2,559.71	2,563.03	1,283.20	1,020,722.6	2,559.10	2,559.75	1,282.50	1,020,353.7	2,557.94	2,559.29
		100	1,285.00	1,022,023.3	2,562.53	2,564.48	1,283.40	1,021,242.7	2,559.95	2,562.39	1,282.80	1,020,929.0	2,558.96	2,561.89
MT6	100	30	512.25	203,351.0	1,020.63	1,022.84	512.93	203,303.9	1,021.19	1,022.29	512.77	203,451.1	1,021.40	1,022.17
		50	518.06	204,174.6	1,028.49	1,029.95	518.07	204,032.7	1,028.15	1,029.32	517.17	203,872.5	1,026.85	1,028.06
		100	530.39	206,802.9	1,047.39	1,050.49	527.48	206,714.3	1,044.27	1,047.22	525.25	206,668.7	1,041.93	1,048.11
MT7	150	30	508.16	201,621.9	1,012.22	1,012.67	507.19	201,631.8	1,011.27	1,011.75	507.08	201,600.3	1,011.08	1,011.45
		50	509.29	202,175.7	1,014.73	1,015.69	508.56	201,864.1	1,013.22	1,014.55	508.49	202,030.2	1,013.57	1,014.27
		100	514.16	202,695.8	1,020.90	1,022.02	512.02	203,152.6	1,019.91	1,021.52	512.87	203,155.2	1,020.76	1,022.40
MT8	200	30	818.10	406,701.3	1,631.50	1,632.51	817.93	406,622.6	1,631.18	1,631.53	817.82	406,520.0	1,630.86	1,631.09
		50	821.05	406,762.7	1,634.57	1,635.83	818.11	406,667.9	1,631.45	1,633.72	817.85	406,691.2	1,631.24	1,632.99
		100	823.86	406,911.6	1,637.69	1,642.60	823.91	407,206.7	1,638.32	1,640.44	823.91	407,351.6	1,638.61	1,639.88
MT9	300	30	815.44	485,287.6	1,624.26	1,624.85	814.67	485,194.4	1,623.32	1,623.69	814.20	485,107.0	1,622.71	1,623.25
		50	816.41	485,544.2	1,625.65	1,627.76	814.21	485,310.4	1,623.07	1,626.03	815.46	485,521.1	1,624.66	1,626.56
		100	896.13	487,239.4	1,708.19	1,709.21	820.84	485,492.0	1,629.99	1,676.41	820.64	487,090.6	1,632.45	1,662.91
MT10	500	30	1,516.00	906,931.1	3,027.55	3,028.31	1,514.95	906,697.6	3,026.11	3,026.62	1,514.91	906,824.1	3,026.28	3,026.53
		50	1,520.56	907,169.7	3,032.51	3,032.99	1,517.93	907,245.6	3,030.01	3,030.71	1,517.44	906,775.1	3,028.73	3,029.87
		100	1,526.27	908,840.3	3,041.00	3,042.56	1,523.89	909,188.6	3,039.21	3,040.59	1,522.51	908,946.6	3,037.42	3,037.98

equal m values. This is not surprising given that the simulated annealing search has more time to find good solutions. However, it is more interesting to observe how the best value of the phase switch variable m changes with different time durations. The results show that low values of m yield the best results given only small time durations for phase 2. For some of these problems, the best-fit heuristic creates solutions with several holes during the latter stages of packing that cannot be consumed when m is set to 30 rectangles. To obtain further improvements, more rectangles may be passed to phase 2, although inevitably this requires longer execution times. This can be demonstrated by observing the results for problems MT1 and MT4. With 60-second durations, the best averages for these two problems are achieved with $m = 30$

rectangles. However, if we allow a longer duration of 600 seconds, both problems exhibit better average evaluations using an m value of 50 rectangles. Increasing the number of rectangles passed to phase 2 can yield improvements, although ultimately this depends somewhat on the quality of the solution given by the best-fit heuristic and the properties of the problem being solved. If the best-fit heuristic is able to produce a very good solution with very few holes toward the latter stages of the packing, then by increasing the number of rectangles passed to phase 2, more of best fit's desirable layout is destroyed and must be rediscovered by the simulated annealing search. This inevitably requires much longer search durations and can often result in the simulated annealing being unable to find an equivalent layout given reasonable

time. The results indicate that this is especially the case with problems created from the dissection of a large rectangle (MT6–MT10) because the best average solutions are consistently obtained using an m value of 30 rectangles. Also, the bottom-left-fill heuristic becomes slower with a greater number of rectangles due to the extra collision detection tests that are required. This also favours smaller m phase switch values.

5. Conclusions

We have previously demonstrated that the best-fit heuristic outperforms other approaches on the two-dimensional orthogonal stock-cutting problem for problem instances with over 50 rectangles. For smaller-sized problems, metaheuristic bottom-left-fill methods were still superior. This paper has proposed an approach that hybridises the best-fit heuristic with simulated annealing. The proposed method was tested on 47 test problems. For 41 of those problems, we made improvements on solution quality when compared against best fit used in isolation. This was possible with only a slight increase in the computational time required. For some of the problem instances, we extended the amount of time, which enabled us to pass a greater number of rectangles to the simulated annealing phase. This yielded further improvements and produced four new best solutions. We also introduced 10 new problems and set initial benchmarks for them.

In summary, we have shown that a best-fit methodology, hybridised with simulated annealing, is able to significantly improve upon the solutions given by other approaches from the literature. It can also produce solutions that are less than 1% over optimum.

Acknowledgments

The authors thank the anonymous referees and the editor for their helpful comments. The authors are also grateful for the support of the Engineering and Physical Sciences Research Council (EPSRC) under Grant GR/S52414/01.

References

Baker, B. S., E. G. Coffman, Jr., R. L. Rivest. 1980. Orthogonal packing in two dimensions. *SIAM J. Comput.* 9(4) 846–855.

Belov, G., G. Scheithauer. 2006. A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. *Eur. J. Oper. Res.* 171(1) 85–106.

Bennell, J. A., X. Song. 2008. A comprehensive and robust procedure for obtaining the no-fit polygon using Minkowski sums. *Comput. Oper. Res.* 35(1) 267–281.

Bennell, J. A., K. A. Dowsland, W. B. Dowsland. 2001. The irregular cutting-stock problem—A new procedure for deriving the no-fit polygon. *Comput. Oper. Res.* 28(3) 271–287.

Burke, E. K., G. Kendall. 1999. A comparison of meta-heuristic algorithms for clustering rectangles. *Comput. Indust. Engrg.* 37(1–2) 383–386.

Burke, E. K., G. Kendall, G. Whitwell. 2004. A new placement heuristic for the orthogonal stock cutting problem. *Oper. Res.* 52(4) 655–671.

Burke, E. K., R. S. R. Hellier, G. Kendall, G. Whitwell. 2006. A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem. *Oper. Res.* 54(3) 587–601.

Burke, E. K., R. S. R. Hellier, G. Kendall, G. Whitwell. 2007. Complete and robust no-fit polygon generation for the irregular stock cutting problem. *Eur. J. Oper. Res.* 179(1) 27–49.

Coffman, Jr., E. G., F. T. Leighton. 1989. A provably efficient algorithm for dynamic storage allocation. *J. Comput. System Sci.* 38(1) 2–35.

Coffman, Jr., E. G., M. R. Garey, D. S. Johnson. 1978. An application of bin packing to multiprocessor scheduling. *SIAM Comput.* 7(1) 1–17.

Coffman, Jr., E. G., M. R. Garey, D. S. Johnson. 1984. Approximation algorithms for bin packing—An updated survey. *Algorithm Design for Computer Systems Design*. Springer-Verlag, Berlin, 49–106.

Dagli, C. H., A. Hajakbari. 1990. Simulated annealing approach for solving stock cutting problem. *Proc. IEEE Internat. Conf. Systems, Man, Cybernetics, Los Angeles, IEEE, Washington, DC*, 221–223.

Dagli, C. H., P. Poshyanonda. 1997. New approaches to nesting rectangular patterns. *J. Intelligent Manufacturing* 3(3) 177–190.

Dowsland, K. A., W. B. Dowsland. 1992. Packing problems. *Eur. J. Oper. Res.* 56(1) 2–14.

Dyckhoff, H. 1990. A typology of cutting and packing problems. *Eur. J. Oper. Res.* 44(2) 145–159.

Garey, M. R., D. S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, San Francisco.

Garey, M. R., D. S. Johnson. 1981. Approximation algorithms for bin packing problems: A survey. D. Ausiello, M. Lucertini, eds. *Analysis and Design of Algorithms in Combinatorial Optimization. CISM Courses and Lectures*, Vol. 266. Springer, Vienna, 147–172.

Gilmore, P. C. 1966. The cutting stock problem. *Proc. IBM Sci. Comput. Sympos. Combin. Problems*. IBM Data Processing Division, White Plains, NY, 211–224.

Gilmore, P. C., R. E. Gomory. 1961. A linear programming approach to the cutting stock problem. *Oper. Res.* 9(6) 849–859.

Golden, B. L. 1976. Approaches to the cutting stock problem. *IIE Trans.* 8(2) 265–274.

Hopper, E., B. C. H. Turton. 1999. A genetic algorithm for a 2D industrial packing problem. *Comput. Indust. Engrg.* 37(1–2) 375–378.

Hopper, E., B. C. H. Turton. 2001. An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem. *Eur. J. Oper. Res.* 128(1) 34–57.

Hower, W., M. Rosendahl, D. Köstner. 1996. *Evolutionary Algorithm Design. Artificial Intelligence in Design '96*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 663–680.

Iori, M., S. Martello, M. Monaci. 2003. Metaheuristic algorithms for the strip packing problem. P. M. Pardalos, V. Korotkikh, eds. *Optimization and Industry: New Frontiers*. Kluwer Academic Publishers, Boston, 159–179.

Jakobs, S. 1996. On genetic algorithms for the packing of polygons. *Eur. J. Oper. Res.* 88(1) 165–181.

Lai, K. K., J. W. M. Chan. 1996. Developing a simulated annealing algorithm for the cutting stock problem. *Comput. Indust. Engrg.* 32(1) 115–127.

Li, Z., V. Milenkovic. 1995. Compaction and separation algorithms for non-convex polygons and their applications. *Eur. J. Oper. Res.* 84(3) 539–561.

Lodi, A., S. Martello, M. Monaci. 2002. Two-dimensional packing problems: A survey. *Eur. J. Oper. Res.* 141(2) 241–252.

- Paul, A. E. 1956. Linear programming: A key to optimum newsprint production. *Pulp Paper Magazine Canada* 57 85–90.
- Ramesh Babu, A., N. Ramesh Babu. 1999. Effective nesting of rectangular parts in multiple rectangular sheets using genetic and heuristic algorithms. *Internat. J. Productions Res.* 37(7) 1625–1643.
- Roberts, S. A. 1984. Application of heuristic techniques to the cutting stock problem for worktops. *J. Oper. Res. Soc.* 35(5) 369–377.
- Smith, D. 1985. Bin packing with adaptive search. *Proc. 1st Internat. Conf. Genetic Algorithms Appl.*, Lawrence Erlbaum Associates, Hillsdale, NJ, 202–207.
- Stoyan, Y., G. Scheithauer, D. Pridatko, T. Romanova, N. Gil. 2002. Phi-functions for primary 3D-objects. Technical Report MATH-NM-15-2002, Technical University of Dresden, Dresden, Germany.
- Sweeney, P. E., E. R. Paternoster. 1992. Cutting and packing problems: A categorized application-orientated research bibliography. *J. Oper. Res. Soc.* 43(7) 691–706.
- Valenzuela, C. L., P. Y. Wang. 2001. Heuristics for large strip packing problems with guillotine patterns: An empirical study. *Proc. 4th Metaheuristics Internat. Conf., (MIC'2001), Porto, Portugal*, 417–421.