Engineering Applications of Artificial Intelligence 22 (2009) 823–827

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Brief paper

A revision of recent approaches for two-dimensional strip-packing problems ☆

María Cristina Riff [a,*], Xavier Bonnaire [a], Bertrand Neveu [b]

[a] Departamento de Informatica, Universidad Tecnica Federico Santa Maria, Av. Espana 1680, Valparaiso, Chile
[b] INRIA Sophia-Antipolis, 2004 Route des Lucioles, Sophia-Antipolis Cedex, France

ARTICLE INFO

Article history:
Received 28 February 2007
Received in revised form
17 September 2008
Accepted 28 October 2008
Available online 1 February 2009

Keywords:
Strip packing
Heuristic search
Metaheuristics
Constrained problems
Optimization

ABSTRACT

In this paper, we present a review of the recent approaches proposed in the literature for strip-packing problems. Many of them have been concurrently published, given some similar results for the same set of benchmarks. Due to the quantity of published papers, it is difficult to ascertain the level of current research in this area.

© 2008 Published by Elsevier Ltd.

1. Introduction

In this paper, we focus our attention on methods to solve the two-dimensional strip-packing problem, where a set of rectangles (objects) must be positioned on a container (a rectangular space area). This container has a fixed width and a variable height size. The goal, when it is possible, is to put all the objects into the container without overlap, using a minimum height dimension of the container. We can find many applications related to strip-packing problems. For example, various irregularly shaped pieces cut from a bolt of fabric are required to make a cloth. A piece of furniture may require rectangular pieces of glass large sheet of glass, the same is for pieces of wood. In all cases, it is required to minimize both the amounts of stock and waste. Many approaches have been proposed in the literature, and in our understanding, the more complete revision has been presented in Hopper (2000). However, in the last three years the interest in this subject has increased, as well as the number of papers presenting new approaches and improvements to existing strategies. We review here the most recent results in this research area. This paper is organized as follows: in the next section, we present a formulation of the 2D strip-packing problem. Section three is related to the exact methods. In Section four, we review the methods based on heuristics. In Section five, we briefly review metaheuristic methods. Section six presents existing benchmarks to evaluate new approaches. Finally, section seven presents the conclusion and future trends in this research area.

2. Two-dimensional strip-packing formulation

The strip-packing problem consists in placing a set N of n objects (rectangles) on a rectangular area without overlap, such that the height H of the strip is minimized, considering a fixed width W. Each object i has dimensions h_i (height), and w_i (width), $\forall i = 1, \ldots, n$. The position of an object on the rectangular area is identified by the Cartesian coordinates of its bottom left corner x_i, y_i. The model of the 2D strip-packing problem can be formulated as follows:

Minimize H

Subject to

$$x_i + w_i \leqslant W, \quad \forall i \in N \tag{1}$$

$$y_i + h_i \leqslant H, \quad \forall i \in N \tag{2}$$

$$x_i + w_i \leqslant x_j \text{ or } x_j + w_j \leqslant x_i \text{ or} \tag{3}$$

$$y_i + h_i \leqslant y_j \text{ or } y_j + h_j \leqslant y_i, \quad \forall (i,j) \in N, \quad i \neq j \tag{4}$$

$$x_i + y_i \geqslant 0, \quad \forall i \in N. \tag{5}$$

A feasible solution must fulfill the following constraints: the object must be within the rectangular area, and must not overlap

☆ This work has been partially supported by the Fondecyt Project 1080110.
* Corresponding author.
E-mail addresses: Maria-Cristina.Riff@inf.utfsm.cl (M.C. Riff), Xavier.Bonnaire@inf.utfsm.cl (X. Bonnaire), Bertrand.Neveu@sophia.inria.fr (B. Neveu).

0952-1976/$ - see front matter © 2008 Published by Elsevier Ltd.
doi:10.1016/j.engappai.2008.10.025

with any other object. The perfect packing problem considers that all objects must be placed on the rectangular area without overlap nor holes, i.e. $\sum_{i \in N} w_i h_i = WH$. For guillotinable strip-packing problem a solution has the property that it can be obtained by a sequence of cuts parallel to the axes, each of which crosses either the entire length, or width, or the remaining connected rectangular piece. Any guillotinable problem on n objects with a perfect packing has at least 2^{n-1} perfect packings.

3. Exact methods

The following two complete methods for strip packing are based on a branch and bound strategy. A branch and bound method is introduced in Lesh et al. (2004) to solve 2D rectangular perfect packings. They only consider the problems with solution where the whole area is used by objects. That means that there are no free space in the solution. The main idea of this approach is to cut branches when it is impossible in order to put a new object without generating a hole. It allows the algorithm to quickly try a new branch. The tests carried out show that the method is a good option when the problem has less than 30 rectangles. The optimal solution has been obtained by this method for the five instances of problems category N_1, N_2, and for four of the five instances in category N_3. These problem instances have been proposed in Hopper and Turton (2001) and are for perfect packing in a fixed area of width = 200 and height = 200. They differ in the number of objects N_1 considers 17 objects, N_2 has 25 objects and N_3 has 29 objects. These instances can be efficiently solved using a complete method. Therefore, they are not very interesting to compare among methods coming from the heuristic and metaheuristic community. In the worst case, the algorithm needs around 10 min to solve the more complex instance of 29 objects. The time reported on the paper is the waiting time to obtain a solution and the tests were done in a Linux machine with 2000 MHz Pentium processor with unoptimized Java.

Martello et al. (2003) consider a 2D rectangular strip packing without object rotations. They present very good results for problems with the same instances than the Lesh et al. paper with less than 30 rectangles. They also evaluate other benchmarks with encouraging results, including problems with 200 objects. However, the Hopper's instances are perfect strip-packing problems, known to be very hard ones. The key idea of the Martello et al. algorithm is to use a computed lower bound by relaxing the constraint of the object area by dividing the object on a set of slides to be placed on the container. In order to find a real solution, the algorithm must merge the pieces to re-construct the object. The most important contribution is the way to compute the lower bound for the branch and bound procedure. Their best results are obtained for the Hopper's instances. The conclusion of this review is that new metaheuristics proposals must be evaluated using hard instances from the Hopper's perfect strip-packing benchmarks with more than 30 objects, as easier problems can efficiently be solved with complete methods.

4. Heuristics-based methods

4.1. Based on bottom-left (BL) heuristic

The work of Baker et al. (1980) introduced the bottom-left heuristic, which orders the objects according to their areas. The objects are placed at the top and pushed down and left as much as possible. This method was improved by Chazelle (1983) and called bottom left fit (BLF): each object is located at the most bottom and left possible place. Hopper Hopper and Turton (2001) presented BLD which is an improved strategy of BL, where the objects are ordered using various criteria (height, width, perimeter and area) and the algorithm selects the best result obtained. Lesh et al. (2005) and Lesh and Mitzenmacher (2006) focus their research on improving BLD heuristic. They named their new heuristic BLD*. At the beginning, BLD* constructs a list of the objects according to a decreasing criteria (height, width, or other). On its turn, an object is placed according to a probability p. If the object is not accepted, the algorithm searches for another one beginning from the first object not yet accepted in the list. This makes a perturbation of the initial order. The distance between both orders is called the Kendall-tau distance or bubble-sort distance. It counts the number of bubble-sort swaps that could be required to transform the initial order into the perturbed one. This strategy is called Bubble Search (Lesh and Mitzenmacher, 2006), and can be applied to any constructive algorithm in order to randomize a fixed ordering. As in GRASP, this strategy repeats greedy placements with this randomized ordering until a time limit is reached. In the BLD* approach, they also include the rotation capability. The decision to rotate the object is made according to the following rules:

- The algorithm evaluates both orientations and selects the one allowing a lower bottom left position in the container area.
- The algorithm evaluates both orientations and selects the one where the center of the object has the lower position.
- The algorithm evaluates both orientations and selects the one that has a lower upper right position.

The results reported indicate that the upper right corner is the most suitable decision, and the best order is to take the objects from the smallest to the biggest width. Finally, Lesh et al. conclude that this method with bubble search is the best one to solve the most well-known benchmarks, including Hopper's benchmarks. They have compared the algorithm with the work of Iori et al. (2003) and the Lesh et al. algorithm shows better results than Iori et al's. approach for the Hopper's greater instances ht_{13} to ht_{18}.

4.2. Based on best-fit heuristic

Burke et al. (2004) have proposed a best fit (BF) that uses a dynamic ordering for the rectangles to be placed. The algorithm goes through the available places from the most bottom left one, and selects for each place the rectangle that best fits in it (if it exists). It is used for hybrid metaheuristic approaches.

4.3. For guillotinable problems

The heuristics first fit decreasing height (FFDH) and next fit decreasing height (NFDH) proposed in Coffmann et al. (1980) and best fit decreasing height (BFDH) proposed initially in Mumford-Valenzuela et al. (2003) are very similar. In each of them, the objects are vertically oriented and ordered from the highest to the lowest. Each object is packed in a rectangular sub-area of the container in the bottom left corner. The width of the sub-area is given by the container, and the height is given by the first object packed in this sub-area. In NFDH, the object is placed left justified, on the same sub-area than the previous object if it fits. Otherwise, a new sub-area is created, and the object is placed left justified into it. With FFDH, the object is placed left justified, on the first sub-area where it fits. If there is no sub-area where the object can be placed, a new sub-area is created and the object is placed left justified. In BFDH, the object is placed left justified on the sub-area where it fits, for which the remaining horizontal space is minimum. BFDH has shown to be useful for guillotinable problems and it has been tested using well-known problems from the literature. Bortfeldt (2006) introduces a genetic algorithm that generates an initial population using a BFDH* heuristic, which is an improvement of BFDH. It uses the following rules:

- Object rotations are allowed, thus when the algorithm searches to include the current object into a sub-area it tests both orientations and selects the best one.
- Before creating a new sub-area, the algorithm searches on the holes produced on the right of the sub-areas, dividing the available hole on guillotinable holes and tries to include the bigger object in the hole, which is on the most left side of the available area.

For all instances of the Hopper's benchmarks, the results obtained by the algorithm are better than the ones from Ratanapan et al., Dagli et al., Kroger, Schnecke, Iori et al.'s. and Mumford-Valenzuela et al.'s.

Zhang et al. (2006) introduced a recursive heuristic *HR* for problems with guillotine constraint. When the first object is positioned in the container (on the bottom left corner), it identifies two remaining areas. It recursively continues placing the remaining objects. To improve the performance of the heuristic, the authors present a deterministic algorithm that gives priority to the objects with bigger areas. Zhang et al. claim that their algorithm quickly obtains good results on Hopper's benchmarks. For the heuristic, the most difficult problem in time was C_7, which considers 196 objects with 36.07 s, but having a very good gap from the known optimal solution of a 1.8%. The results of these tests have been summarized, it is thus difficult to know which problems have really been solved. The key idea is to find a good order of objects for any positioning heuristic.

5. Metaheuristic approaches

These and the other low-level heuristics have been used in metaheuristic approaches, as tabu search, simulated annealing, and genetic algorithms (GAs). The first idea is to build an initial solution with a low level heuristic, and to perform a local search on the layout. Neveu et al. (2007) have presented an incremental move, which allows additions and removals of rectangles. They also implement a generic metaheuristic using this move obtaining competitive results.

Other researchers prefer to work on the order of objects for each positioning heuristic. Soke and Bingul (2006) present a genetic algorithm (GA) and a simulated annealing (SA) algorithm named GA+BLF and SA+BLF, respectively, both of which try to find the best order for the objects to be placed in the container using the BLF strategy. Thus, the problem being treated here is not where to put the objects, but how the solution is focused on finding the best sequencing order of the objects. They do not allow rotation. The paper is focused on comparing their hybrid genetic approach versus their hybrid-simulated annealing. They have specifically analyzed the influence of the crossover operator for the genetic algorithm, and temperature cooling for the simulated annealing-based algorithm. The comparison is done using problems with 17 and 29 pieces from Hopper's benchmarks. The major drawback of this paper is that there is no comparison with other existing approaches.

In the case of fixed orientation problems, the best approach to our knowledge appears to be the GRASP-based approach described in Alvarez-Valdes et al. (2005). This approach repeats the following two-phases: the rectangles are first placed by a randomized BF-like constructive phase. Then the solution is passed through a variable neighborhood search (VNS), and the obtained solution is kept only if it is better than the previous one.

On the other hand, Bortfeldt (2006) introduced a genetic algorithm called strip packing genetic algorithm layer (SPGAL) and obtained the best known results in the literature for problems that allow the rectangles to be rotated. The algorithm generates an initial population using BFDH* heuristic, which is an improvement of BFDH initially proposed in Mumford-Valenzuela et al. (2003). This heuristic works with a layer structure that takes into account the guillotine cut constraint. The genetic algorithm directly performs a search in this layer structure. For problems without the guillotine cut constraint, a post-optimization procedure breaks this layer structure. The same genetic algorithm is used in Bortfeldt and Gehring (2006) for larger instances (1000 pieces). It is divided in GA-1, GA-2, GA-3 and GA-4, each of them initialized with different parameters. The procedure is only applied to problems with the guillotine cut constraint, because the post-optimization procedure is negligible for large instances (Bortfeldt and Gehring, 2006).

Burke et al. Burke et al. (2006) hybridize the best fit heuristic with metaheuristic approaches such as tabu search (BF+TS), simulated annealing (BF+SA) and genetic algorithms (BF+GA). BF+SA has obtained the best results.

6. Benchmarks

We resume the results for the Hopper's instances for classes of problems $C_1, \ldots, C_7$. The characteristics of these classes are resumed in Fig. 1. Each class is identified by the number of objects to be placed and the stripwidth. A class is composed by three different instances.

GA+BLF and SA+BLF have been proposed by Hopper. In these algorithms, object rotation is allowed. In the test reported in Iori's work object rotation is not allowed, and they have only tested $C_1, \ldots, C_6$. The algorithm found these solutions in less than 32 s. In HR before applying the algorithm, each object is vertically oriented, and then object rotations are forbidden. For SPGAL, the tests are reported with rotations (SPGAL-R) and without rotations (SPGAL). For BLD*, the reported results are only for problems C_5 and C_6, without rotation. A solution is found in 60 s. The reactive grasp algorithm (Alvarez-Valdes et al., 2005) is with fixed orientation of the objects. The Martello's paper reports the results for the tests with at most 30 rectangles using a complete branch and bound method. Considering these instances, i.e. C_1, C_2 and C_3, their algorithm was unable only to solve two problems of C_3 (ht_{07},

Class	Stripwidth	Objects
C_1	20	16-17-16
C_2	40	25
C_3	60	28-29-28
C_4	60	49
C_5	60	73
C_6	80	97
C_7	160	196-197-196

Fig. 1. Hopper's Instances for perfect packing.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	Average
GA+BLF	4	7	5	3	4	4	5	4.57
SA+BLF	4	6	5	3	3	3	4	4
BF+SA	0	6.25	3.33	1.67	1.48	1.39	1.77	2.27
Iori	1.59	2.08	2.15	4.75	3.92	4.00	-	3.98*
HR	8.33	4.45	6.67	2.22	1.85	2.5	1.8	3.97
SPGAL-R	1.7	0.0	2.2	0.0	0.0	0.3	0.3	0.6
SPGAL	1.59	2.08	3.16	2.70	1.46	1.64	1.23	1.98
*BLD**	-	-	-	-	2	2.4	-	2.2*
R-GRASP	0	0	1.08	1.64	1.10	0.83	1.23	0.84
Martello B & B	0	0	2.15					0.71*

Fig. 2. Gap from the optimal solution. "*" means a partial average that is computed using only the available results.

ht_{08}). Fig. 2 shows the percentage from both the optimal solution to the best solution found (gap % $= (best_{\text{found}} - opt)/opt$) and the average for each algorithm.

The optimum value can be obtained for 12 of the 21 problems. However, none of ten algorithms is able to find all of the 12 optimum. The best results obtained by SPGAL-R which is able to find 9 of the 12 optimum and which gives solutions very closed to the optimum for the other problems, and the worst ones are obtained by GA+BLF. SPGAL-R and R-GRASP have the best average for all problems. Almost none of the algorithms can give very good or good results for all problems, which means that each one is more adapted to a specific kind of problems or to a specific configuration. Algorithms like BLD* or Martello B&B are very difficult to evaluate as they have been tested with a very few number of problems.

The most used benchmarks with known optimal solutions are from Hopper and Turton (2001) and Iori et al. (2003). Recently, a new set of 360 new instances of larger strip-packing problems have been proposed in Bortfeldt and Gehring (2006). Optimal solutions for these instances are unknown which only permit to compare algorithm among themselves. However, a comparison point could be made by the algorithm proposed in Bortfeldt (2006).

7. Conclusions

This survey presents a synthetic overview of the latest advances in solving the two-dimensional strip-packing problem. The increasing number of papers published about the strip-packing problem during the last three years in journal and international conferences makes very difficult the task to compile all the existing approaches. Because, we are interested in both Metaheuristics and Heuristics methods, we can then conclude for our further researches that:

- It is absolutely not interesting to compare new incomplete algorithms using Hopper's benchmark problems N_1 and N_2, because they can be efficiently solved by complete branch and bound techniques.
- The most recent research that has shown the best results with benchmarks as a genetic algorithm proposed by Bortfeldt (2006) using the BFDH* heuristic.
- The approach proposed in Zhang et al. (2006) is an inexpensive heuristic that could be further studied to be incorporated in a population algorithm, which could improve the exploration of the search space. It is important to note that in the Zhang et al's. approach the *HR* heuristic could never find a solution, even if the algorithm tries with all the possible objects ordering. This is a major drawback of this technique compared to *BF*. In the BF approach, we know that it exists at least one object ordering which is able to find a solution.

Our future research will both focus on designing inexpensive methods and improving existing ones to solve strip-packing problems. One of the major challenge is to be able to solve large problems that are part of the hardest ones.

Acknowledgments

This work is partially supported by the Collaboration Chilean-French Project INRIA-CONICYT.

The authors wish to thank the anonymous referees for their constructive comments.

References

Alvarez-Valdes, R., Parre no, F., Tamarit, J.M., 2005. Reactive grasp for the strip packing problem. In: Proceedings Metaheuristic Conference MIC, 1.

Baker, B.S., Coffman, E.G., Rivest, R.L., 1980. Orthogonal packings in two dimensions. SIAM Journal on Computing 9, 846–855.

Bortfeldt, A., 2006. A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces. European Journal of Operational Research 172, 814–837.

Bortfeldt, A., Gehring, H., 2006. New large benchmarks for the two-dimensional strip-packing problem with rectangular pieces. In: IEEE Proceedings of the Thirty Ninth Hawaii International Conference on Systems Sciences, p. 30.2.

Burke, E., Kendall, G., Whitwell, G., 2004. A new placement heuristic for the orthogonal stock-cutting problem. Operations Research 52, 697–707.

Burke, E., Kendall, G., Whitwell, G., 2006. Metaheuristic enhancements of the best-fit heuristic for the orthe orthogonal stock-cutting problem. Technical Report, University of Nottingham, NOTTCS-TR-2006-3.

Chazelle, B., 1983. The bottom left bin packing heuristic: an efficient implementation. IEEE Transactions on Computers 32, 697–707.

Coffmann Jr., E., Garey, D., Tarjan, R., 1980. Performance bounds for level oriented two-dimensional packing algorithms. SIAM Journal on Computing 9 (1), 808–826.

Hopper, E., 2000. Two-Dimensional Packing Utilising Evolutionary Algorithms and other Meta-Heuristic Methods. Ph.D. Thesis Cardiff University.

Hopper, E., Turton, B.C.H., 2001. An empirical investigation on metaheuristic and heuristic algorithms for a 2d packing problem. European Journal of Operational Research 128, 34–57.
Iori, M., Martello, S., Monaci, M., 2003. Metaheuristic Algorithms for the Strip Packing Problem. Kluwer Academic Publishers, Dordrecht, pp. 159–179.
Lesh, N., Mitzenmacher, M., 2006. Bubble search: a simple heuristic for improving priority-based greedy algorithms. Information Processing Letters 97, 161–169.
Lesh, N., Marks, J., Mc. Mahon, A., Mitzenmacher, M., 2004. Exhaustive approaches to 2D rectangular perfect packings. Information Processing Letters 90, 7–14.
Lesh, N., Marks, J., Mc. Mahon, A., Mitzenmacher, M., 2005. New heuristic and interactive approaches to 2D rectangular strip packing. ACM Journal of Experimental Algorithmics 10, 1–18.
Martello, S., Monaci, M., Vigo, D., 2003. An exact approach to the strip-packing problem. INFORMS Journal of Computing 15, 310–319.
Mumford-Valenzuela, C., Vick, J., Wang, P.Y., 2003. Heuristics for Large Strip Packing Problems with Guillotine Patterns: An Empirical Study. Kluwer Academic Publishers, Dordrecht, pp. 501–522.
Neveu, B., Trombettoni, G., Araya, I., 2007. Incremental move for strip-packing. Proceedings of ICTAI, Greece.
Soke, A., Bingul, Z., 2006. Hybrid genetic algorithm and simulated annealing for two-dimensional non-guillotine rectangular packing problems. Engineering Applications of Artificial Intelligence 19, 557–567.
Zhang, D., Kang, Y., Deng, A., 2006. A new heuristic recursive algorithm for the strip rectangular packing problem. Computers and Operations Research 33, 2209–2217.