A Novel Quasi-human Heuristic Algorithm for Two-dimensional Rectangle Packing Problem

Duanbing Chen[1], Wenqi Huang[1, 2]

[1] College of Computer Science, Huazhong University of Science and Technology, Wuhan 430074, China

[2] Corresponding author

Summary

Two-dimensional rectangle packing problem is the problem of packing a series of rectangles into a larger container with maximum area usage of the container. This problem involves many industrial applications, such as shipping, timber cutting, very large scale integration (VLSI) design, etc. It belongs to a subset of classical packing problems and has been shown to be NP hard. For solving this problem, many algorithms such as genetic algorithm, simulated annealing and other heuristic algorithms are presented. In this paper, a novel quasi-human heuristic algorithm is proposed according to the experience and wisdom of human being. 21 rectangle-packing test instances are tested by the produced algorithm, 16 instances of them having achieved optimum solutions within reasonable runtime. The experiment results demonstrate that the produced algorithm is rather efficient for solving two-dimensional rectangle packing problem.

Key words:

Two-dimensional rectangle packing; Quasi-human heuristic; Corner-occupying action; Caving degree

1. Introduction

Two-dimensional rectangle packing problem is the problem of packing a series of rectangles into a larger container with maximum area usage of the container. This problem involves many industrial applications, such as shipping, timber cutting, very large scale integration (VLSI) design, etc. It belongs to a subset of classical packing problems and has been shown to be NP hard [1]. For solving this problem, various algorithms based on different strategies have been suggested. In general, these algorithms can be classified into two major categories: non-deterministic algorithms and deterministic algorithms.

In 2001, Hopper and Turton gave an empirical investigation of meta-heuristic and heuristic algorithms of the orthogonal packing problem of rectangles[2]. Recently, some robust heuristic algorithms were presented[3, 4]. And some literatures combine genetic algorithm or simulated annealing with deterministic method and obtain hybrid algorithm[5, 6]. In addition, some people formalize the experience and wisdom of human being and obtain the quasi-human heuristic algorithm[7].

Inspired by the above approaches, a novel quasi-human heuristic algorithm for two-dimensional rectangle packing problem is proposed according to the experience and wisdom of human being. The objective is to maximize the area usage of the container. The key point of this algorithm is that the rectangle packed into the container always occupies a corner, even a cave, where possible. In this way, the rectangles will be close to each other wisely, and the wasted space is decreased. Compared with those in literatures, the results from our method are much improved. For 21 rectangle-packing test instances taken from reference [2], 16 of them are achieved optimum solutions by our method, 2 instances achieved optimum solutions by Heuristic1[3], and 3 instances achieved optimum solutions by HH[4]. Experiment results show that our algorithm is rather efficient for solving two-dimensional rectangle packing problem.

2. Problem description and mathematical formulation

An empty container C_0 is given of width w_0 and height h_0. And there is a series of rectangles R_i of width w_i and height h_i (i=1, 2,…, n). The objective is to maximize the area usage of the container. The constraints for packing rectangles are:

1. Each edge of a rectangle must be parallel to an edge of the container.
2. There is no overlapping for any two rectangles, i.e., the overlapping area is zero.

3. Description of our approach

3.1. Main idea

If some rectangles have been packed into the container without overlapping, the question is which one is the best candidate for the remainder, and which position is the best one to be filled? An ancient Chinese proverb "*Golden corners, silvery sides, and strawy voids*" can be used to answer this question relevantly. This proverb means the corner inside the container is the best place to be filled, then the boundary line of the empty space, and the void space is the worst. And more, if the rectangle not only occupies a corner, but also touches some other rectangles, the action for packing this rectangle is perfect. We may call the corresponding packing action as cave-occupying action. So, we can develop foresaid proverb into *"Golden corners, silvery sides, strawy voids, and highly valuable diamond cave"*. Thus, the following packing rule is natural: The rectangle to be packed into the container always occupies a corner, and the caving degree of the packing action should be as large as possible. So, the rectangles are close to each other wisely, and the area usage of the container improves consequently.

3.2. Fundamental conceptions

(1) Corner-occupying action (COA)

A packing action is called a *corner-occupying action* (COA), if the rectangle to be packed touches two previously packed rectangles including the container, and the touching lengths are longer than zero. The rectangle to be packed occupies a corner formed by those two previously packed rectangles. For example, in figure 1, the shadowy rectangles have been packed, and the rectangle "1" is outside the container. The packing action is a COA, if rectangle "1" is situated at place A, B, C or D; it is not a COA if situated at place E or F.

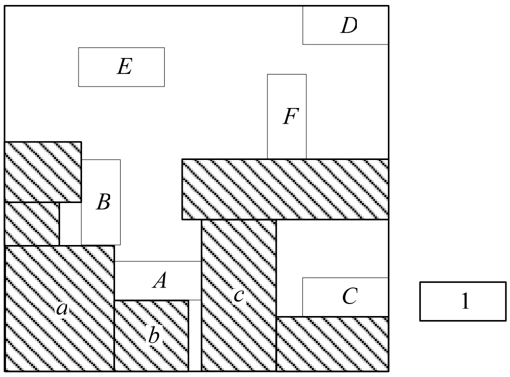


Figure 1 Corner-occupying action

In particular, a COA is called a cave-occupying action, if the rectangle related to this COA not only occupies a corner, but also touches some other previously packed rectangles including the container. For example, in figure 1, if rectangle "1" is situated at place A, it occupies the corner formed by rectangles a and b. Furthermore, it touches rectangle c. Thus, rectangle "1" occupies a cave formed by rectangles a, b and c, and this COA for packing rectangle "1" is a cave-occupying action.

(2) Caving degree of COA

As shown in figure 2, if a rectangle R_i is packed into the container according to a COA, let the Manhattan distance between rectangle R_i and it's nearest rectangles (except the rectangles a and b that form this corner) be $d_{\min}$, the caving degree C_{COA} of the corresponding COA can be defined as following:

$$C_{COA} = 1 - \frac{d_{\min}}{\sqrt{w_i \cdot h_i}} \tag{1}$$

Where w_i and h_i is the width and height of R_i respectively.

Actually, the caving degree reflects the nearness between the rectangle to be packed and the previously packed rectangles (except the rectangles that form this corner). It is equal to 1 when the corresponding rectangle occupies a cave formed by three or more previously packed rectangles, and it is less than 1 when just occupies a corner formed by two previously packed rectangles. In the packing process, we always select the COA with the largest caving degree, and pack the corresponding rectangle into the container at the corresponding position and orientation.

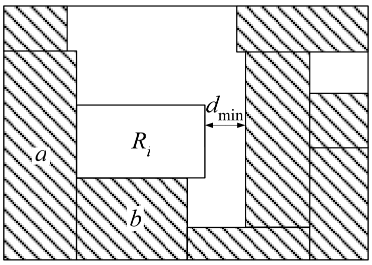


Figure 2 Caving degree of COA

(3) Edge degree of COA

For a given COA, the related rectangle is R, the number of edges that overlap with the rectangle R is defined as edge degree of the corresponding COA. For example, as shown in figure 3, the shadowy rectangles have been packed, if rectangle “1” is situated at place A, B, C, the edge degree of the corresponding COA is 3, 2, 4, respectively.

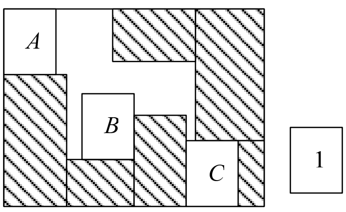


Figure 3 Edge degree of COA

(4) Precedence of point

Let points P_1 (x_1, y_1) and P_2 (x_2, y_2) be two points in the plane rectangular coordinates *o-xy*, P_1 has precedence over P_2 if $x_1 < x_2$, or if $x_1 = x_2$ and $y_1 < y_2$.

3.3. Selecting rule for COA

In order to pack a rectangle at a “good” position and orientation, we construct a series of rules as follows.

Main rule: select the COA with the largest caving degree.

Tiebreaker 1: select the COA with the largest edge degree.

Tiebreaker 2: select the COA with the highest precedence of the bottom left of the corresponding rectangle.

Tiebreaker 3: select the COA with the corresponding rectangle packed horizontally, if both the horizontal and vertical packings are feasible.

Tiebreaker 4: select the COA with the smallest index of the corresponding rectangle.

3.4. Sketch of the algorithm

Under the current configuration, we enumerate all COAs and calculate the caving degree and edge degree for each of them. Then select a COA according to the selecting rule (see section 3.3) and pack the corresponding rectangle into the container at the corresponding position and orientation. Repeat this process until no rectangle remains outside the container or, none of the remaining rectangles can be packed in.

Actually, above text describes a pure greedy packing process. In order to obtain higher area usage of the container, we introduce backtracking process. First, enumerate all COAs by pseudo-packing each remaining rectangle. By “pseudo-pack” we mean the rectangle is temporarily packed and will be removed from the container in the future. Second, for each COA, pseudo-pack the rectangle according to the COA, and calculate a score for this COA by calling greedy packing process. This score represents the quality of the COA, the higher the score is, the higher the quality of COA is. Finally, select the COA with the highest score and pack the corresponding rectangle according to the selected COA, if there are multiple COAs with the highest score, then use the selecting rule (see section 3.3) to break tie. Repeat this process until no rectangle remains outside the container or, none of the remaining rectangles can be packed in.

4. Implementations and Experiments

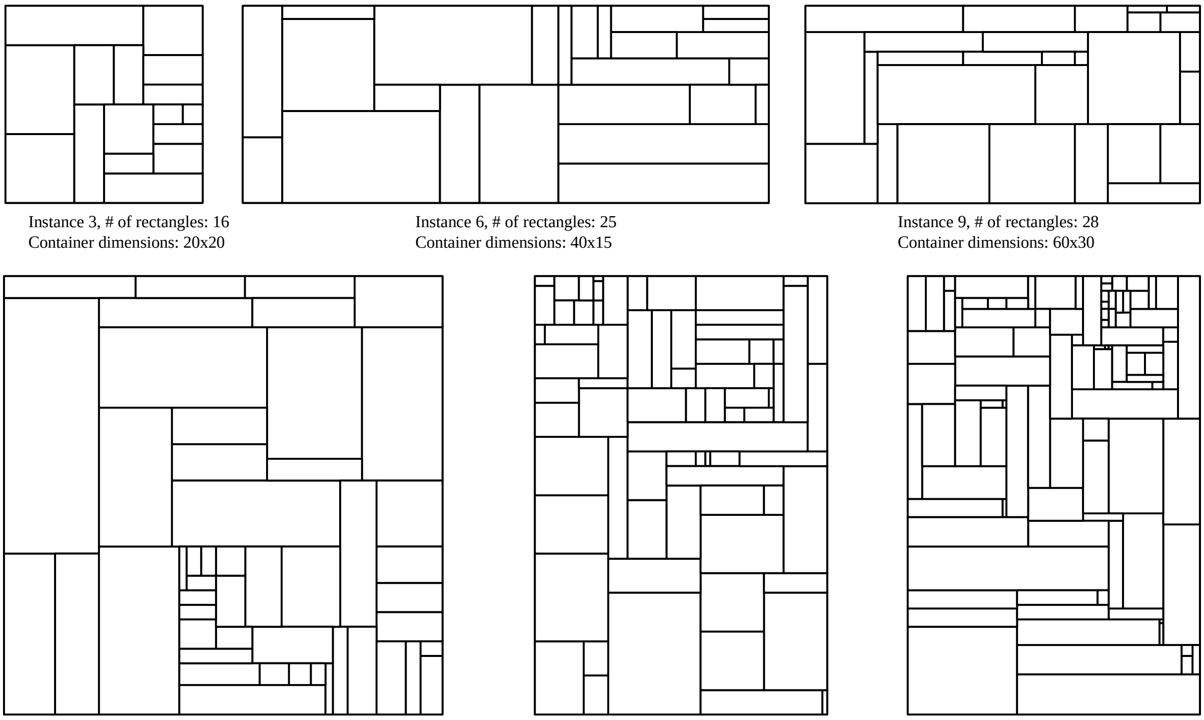
The algorithm proposed in this paper is implemented by C#.net programming language running on an IBM portable PC with 2.0GHz processor and 256MB memory. The performance of the algorithm has been tested with 21 rectangle-packing test instances taken from Hopper and Turton [2].

We compare our algorithm with some most advanced algorithms, Heuristic1[3] and HH[4]. Algorithm Heuristic1 and HH are not implemented in this paper, so the results are directly taken from references [3] and [4]. Heuristic1 is run on a SUN Sparc20/71 with a 71MHz SuperSparc CPU and 64M RAM; HH is run on a Dell GX260 with a 2.4GHz CPU; and our algorithm is run on an IBM portable PC with a 2.0GHz processor and 256MB memory. As an example, the layouts of instances 3, 6, 9, 12, 15 and 17 achieved by our algorithm are shown in figure 4. In figure 4, each layout is the perfect layout, that is, all rectangles are packed into the container, and there is no dead space for the container.

For 21 rectangle-packing instances provided by Hopper and Turton[2]. The area usage of the container of each instance is obtained by our algorithm. For 21 instances, 16 of them having achieved optimum solutions, i.e., all rectangles are packed into the container without overlapping, the area usage of the container is 100%, and % of unpacked area is 0%, where % of unpacked area is defined by:

% of unpacked area =100 (*the total area of unpacked rectangles* / *container area)* (2)

3 and 2 instances achieved optimum solutions by HH and heuristic1, respectively. Table 1 shows the results of 21 instances between heuristic1, HH and our method. From table 1, we can see that our algorithm is rather efficient for solving two-dimensional rectangle packing problem.



Instance 3, # of rectangles: 16
Container dimensions: 20x20

Instance 6, # of rectangles: 25
Container dimensions: 40x15

Instance 9, # of rectangles: 28
Container dimensions: 60x30

Instance 12, # of rectangles: 49
Container dimensions: 60x60

Instance 15, # of rectangles: 73
Container dimensions: 60x90

Instance 17, # of rectangles: 97
Container dimensions: 80x120

Figure 4 The packing results of instance 3, 6, 9, 12, 15 and 17

Table 1. Computational Experiments on Heuristic1[3], HH[4] and our algorithm

Instance	# of rectangles	Container dimensions (wxh)	Our algorithm		Heuristic1		HH	
			% of unpacked area	Runtime (sec)	% of unpacked area	Runtime (sec)	% of unpacked area	Runtime (sec)
1	16	20 x 20	0	0.04	2	1.48	2	0.00
2	17	20 x 20	0	0.59	2	2.42	3.5	0.00
3	16	20 x 20	0	0.03	2.5	2.63	0	0.00
4	25	40 x 15	0	0.12	0.67	13.35	0.67	0.05
5	25	40 x 15	0	0.16	0	10.88	0	0.05
6	25	40 x 15	0	3.01	0	7.92	0	0.00
7	28	60 x 30	0	2.43	0.67	23.72	0.67	0.05
8	29	60 x 30	0	42.83	0.83	34.02	2.44	0.05
9	28	60 x 30	0	3.09	0.78	30.97	1.56	0.05
10	49	60 x 60	0.22	142.56	0.97	438.18	1.36	0.44
11	49	60 x 60	0	195.95	0.22	354.47	0.78	0.44
12	49	60 x 60	0	40.72	No report	No report	0.44	0.33
13	73	60 x 90	0	16.49	0.3	1417.52	0.44	1.54
14	73	60 x 90	0	17.18	0.04	1507.52	0.44	1.81
15	73	60 x 90	0	289.074	0.83	1466.15	0.37	2.25
16	97	80 x 120	0	1409.86	0.25	7005.73	0.66	5.16
17	97	80 x 120	0	476.47	3.74	5537.88	0.26	5.33
18	97	80 x 120	0.13	1748.53	0.54	5604.70	0.5	5.60
19	196	160 x 240	0.10	3745.91	No report	No report	1.25	94.62
20	197	160 x 240	0.08	3158.71	No report	No report	0.55	87.25
21	196	160 x 240	0.13	3712.34	No report	No report	0.69	78.02

5. Conclusion

In this paper, a novel quasi-human heuristic algorithm for two-dimensional rectangle packing problem is proposed. High area usage of the container can be obtained by this algorithm within reasonable runtime. 16 of 21 test instances taken from Hopper and Turton[2] are achieved optimum solutions. The experiment results demonstrate that the algorithm proposed in this paper is rather efficient for solving two-dimensional rectangle packing problem.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China under Grant NO. 10471051 and by NKBRPC (2004CB318000).

References

[1] Leung J, Tam T, Wong C S, Young S, Chin F (1990). Packing squares into square, Journal of Parallel and Distributed Computing, 10: 271-275.

[2] Hopper E, Turton B (2001). An empirical investigation of meta-heuristic and heuristic algorithm for a 2D packing problem, European Journal of Operational Research, 128: 34-57.

[3] Wu Y L, Huang W, Lau S, Wong C K, Young G H (2002). An effective quasi-human based heuristic for solving the rectangle packing problem, European Journal of Operational Research, 141: 341-358.

[4] Zhang D, Deng A, Kang Y (2005). A hybrid heuristic algorithm for the rectangular packing problem, Lecture Notes in Computer Science, 3514: 783-791.

[5] Liu D, Teng H (1999). An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles, European Journal of Operational Research, 112: 413-420.

[6] Leung T W, Chan C K, Troutt M D (2003). Application of a mixed simulated annealing-genetic algorithm heuristic for the two-dimensional orthogonal packing problem, European Journal of Operational Research, 145: 530-542.

[7] Huang W, Li Y, Akeb H, Li C M (2005). Greedy algorithms for packing unequal circles into a rectangular container, Journal of the Operational Research Society, 56: 539-548.

[8]

Duanbing Chen is a PhD candidate of Computer Science at Huazhong University of Science and Technology, P.R. China. He received his BSc in Mathematics in 1994 from Sichuan Normal College, his MS in 1997 from Southwest Petroleum Institute. His main area of research is in combinatorial optimization. He has published some articles on this topic in journals such as *Computers and Operations Research* and *Computer Science (in Chinese).*

Wenqi Huang is a Professor of Computer Science at Huazhong University of Science and Technology, P.R. China. He received his BSc in Mathematics in 1964 from Peking University. From 1981 to 1982, he worked as a visiting scholar at Mathematics Department, Cornell University. He got the gold prize of the Third International Competition for Solving NP-Complete Problem SAT in 1996. His research interests include P =? NP problem, practical algorithm for solving NP-hard problems. He has published numerous articles on this topic in journals such as *European Journal of Operational Research*, *Journal of Operational Research Society*, *Computers and Operations Research*, *Simulation Modeling Practice and Theory*, and *Annals of Operations Research*.