

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



An Efficient Quasi-Human Heuristic Algorithm for Solving the Rectangle-Packing Problem

Wenqi Huang¹ and Duanbing Chen²

¹ School of Computer Science,
Huazhong University of Science and Technology, Wuhan 430074, P.R. China

² School of Computer Science,
University of Electronic Science and Technology of China, Chengdu 610054,
P.R. China

1. Introduction

Rectangle-packing problem involves many industrial applications. For example, in shipping industry, various size boxes have to be loaded as many as possible in a larger container. In wood or glass industries, rectangular components have to be cut from large sheets of material. In very large scale integration (VLSI) floor planning, various chips have to be laid on the chip board, and so on. The rectangle-packing problem belongs to a subset of classical cutting and packing problems and has shown to be NP hard (Leung *et al.*, 1990). For more extensive and detailed descriptions of packing problem, please refer to Lodi *et al.* (2002) and Pisinger (2002). Various algorithms based on different strategies have been suggested to solve this problem. In general, these algorithms can be classified into two major categories: non-deterministic algorithms and deterministic algorithms. The key aspect of non-deterministic algorithms, such as simulated annealing and genetic algorithm (Hopper & Turton, 1999; Bortfeldt, 2006), is to design data structure that can represent the topological relations among the rectangles. The key aspect of deterministic algorithms is to determine the packing rules, such as *less flexibility first* principle (Wu *et al.*, 2002).

Optimal algorithm for orthogonal two-dimensional cutting is proposed in Beasley (1985), but it might not be practical for large scale problems. In order to improve the quality of solution, some scholars combine genetic algorithm or simulated annealing with deterministic method and obtain hybrid algorithms (Liu & Teng, 1999; Leung *et al.*, 2003). Some heuristic and meta-heuristic algorithms are also presented in literatures (Lodi *et al.*, 1999; Hopper & Turton, 2001; Zhang *et al.*, 2005; Burke *et al.*, 2004). In recent years, some people began to formalize the wisdom and experience of human being and obtain the quasi-human heuristic algorithms (Huang & Jin, 1997; Huang & Xu, 1999; Wu *et al.*, 2002; Huang *et al.*, 2007). The “quasi-human” tries to simulate the behavior of human being in related special work such as bricklaying.

Huang *et al.* (2007) presented a heuristic algorithm based on two important concepts, namely, the corner-occupying action and caving degree. Based on Huang *et al.* (2007), an efficient quasi-human heuristic algorithm (QHA) for solving rectangle-packing problem is proposed on the basis of the wisdom and experience of human being in this paper. The

Source: Simulated Annealing, Book edited by: Cher Ming Tan, ISBN 978-953-7619-07-7, pp. 420, February 2008, I-Tech Education and Publishing, Vienna, Austria

objective is to maximize the area usage of the box. The key point of this algorithm is that the rectangle packed into the box always occupies a corner, even a cave, if possible. Furthermore, the rectangle should occupy as many corners and overlap as many edges with other previously packed rectangles as possible. In this way, the rectangles will be close to each other wisely, and the spare space is decreased. As compared with reviewed literatures, the results from QHA are much improved. For 21 rectangle-packing test instances taken from Hopper & Turton (2001), optimal solutions of 19 instances are achieved by QHA, and two, three and sixteen ones by the algorithm in Wu et al. (2002), Zhang et al. (2005) and Huang et al. (2007), respectively. For each of 13 random instances taken from Burke et al. (2004), the container height obtained by QHA is smaller than that by best fit (BF) heuristic (Burke et al., 2004). Furthermore, optimal solutions of three instances are achieved by QHA. Experimental results show that QHA is rather efficient for solving the rectangle-packing problem.

2. Problem description

Given an empty box B_0 with width w_0 and height h_0 , and a series of rectangles R_i with width w_i and height h_i ($i=1, 2, \dots, n$). The task is to pack as many rectangles into the box B_0 as possible, where the measurement of "many" is the total area of the already packed rectangles. The constraints for packing rectangles are:

1. Each edge of a packed rectangle should be parallel to an edge of the box.
2. There is no overlapping area for any two already packed rectangles, and any packed rectangle should not exceed the box boundary.
3. The rectangle should be packed horizontally or vertically.

Without significant loss of generality, it is usual to assume that all w_i and h_i ($i=0, 1, \dots, n$) are integers.

3. Algorithm description

3.1. Main idea

If some rectangles have been packed into the box without overlapping, that is, the overlapping area is zero, the question is which rectangle is the best candidate for the remainder, and which position is the best one to be filled. There is an aphorism in ancient China: "*Golden corners, silvery edges, and strawy voids*". It means that the empty corner inside the box 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 action as cave-occupying action. Therefore, we can develop foresaid aphorism into "*Golden corners, silvery edges, strawy voids, and highly valuable diamond cave*". In addition, we hope that the rectangle occupies as many corners and overlaps as many edges with previously packed rectangles as possible. Thus, the following packing principle is natural: The rectangle to be packed into the box always occupies a corner, and the caving degree of the packing action should be as large as possible, where the caving degree reflects the closeness between the rectangle to be packed and the previously packed rectangles, the details about caving degree will be described in 3.2(6). Furthermore, the rectangle should occupy as many corners and overlap as many edges with other previously packed rectangles as possible. Thus, the rectangles are close to each other wisely. Actually, this

strategy describes a quasi-human idea, that is, to simulate the behavior of human being in related special work such as bricklaying.

3.2. Definitions

The concepts of corner-occupying action, cave-occupying action and caving degree are presented in Huang et al. (2007). We summarize them in this paper again. For more details, the readers are referred to Huang et al. (2007). In this paper, other two important concepts, i.e., corner degree and edge degree, are presented.

(1) Corner-occupying action (COA)

A packing action is called a *corner-occupying action* (COA), if the edges of the rectangle to be packed overlap the different directional edges with other two previously packed rectangles including the box (we can regard the 4 edges of the box as 4 rectangles with very small height which have been packed at the prespecified positions), and the overlapping lengths are longer than zero. Note that the two rectangles are not necessarily touching each other. A COA is called a feasible one, if the rectangle to be packed does not overlap with any previously packed rectangle, i.e., the overlapping area is zero, and does not exceed the box boundary. For example, in Fig. 1, the shadowed rectangles have been packed, and the rectangle "1" is outside the box. The packing action is a feasible COA, if rectangle "1" is situated at place A, B, C or D; it is a non-feasible COA if situated at place E or F; it is not a COA if situated at place G or H.

(2) Cave-occupying action

A packing action is called a cave-occupying action if the rectangle to be packed not only occupies a corner, but also touches some other previously packed rectangles including the box. For example, in Fig. 2, the shadowy rectangles have been packed. Rectangle A occupies the corner formed by rectangles a and b. Furthermore, it touches rectangle c. Thus, rectangle A occupies a cave formed by rectangles a, b and c. The action of packing rectangle A is a cave-occupying action. Actually, a cave-occupying action is a special COA.

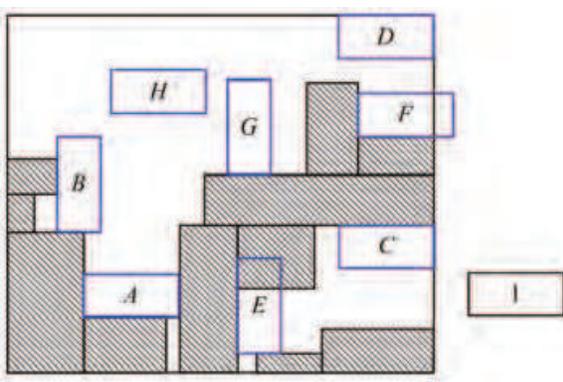


Fig. 1 Corner-occupying action

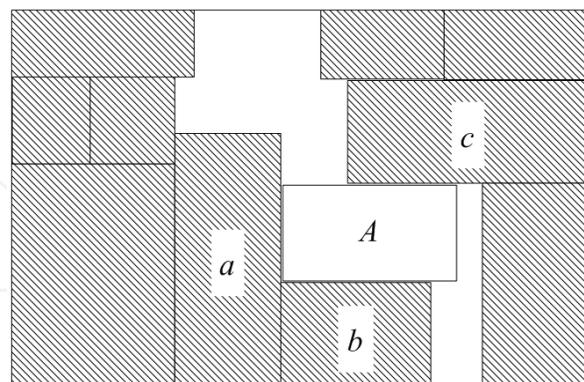


Fig. 2 Cave-occupying action

(3) Configuration

Fig. 3 shows a configuration. Some rectangles have been packed into the box without overlapping area and some remain outside. A configuration is called an initial one if there is no rectangle in the box. A configuration is called an end one if all n rectangles have been packed into the box without overlapping area or, no feasible COA can be done although some rectangles remain outside.

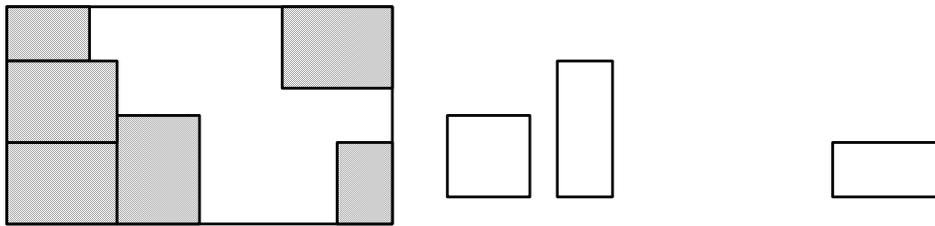


Fig. 3 Configuration

(4) Distance between two rectangles

For two given rectangles R_i with width w_i and height h_i and R_j with width w_j and height h_j , the central coordinate of rectangle R_i and R_j is (x_i, y_i) and (x_j, y_j) , respectively. These two rectangles do not overlap (i.e., the overlapping area is zero) if $|x_i - x_j| \geq \frac{1}{2}(w_i + w_j)$ or $|y_i - y_j| \geq \frac{1}{2}(h_i + h_j)$, and more, the distance d_{ij} between rectangle R_i and R_j is defined as follows:

$$d_{ij} = \max\left(|x_i - x_j| - \frac{1}{2}(w_i + w_j), 0\right) + \max\left(|y_i - y_j| - \frac{1}{2}(h_i + h_j), 0\right).$$

In fact, d_{ij} is the Manhattan distance between two rectangles which is an extension of Manhattan distance between two points.

(5) Distance between one rectangle and several other rectangles

For a given rectangle R and a set of rectangles $\{R_i | i=1, 2, \dots, m\}$. Let the distance between R and R_i ($i=1, 2, \dots, m$) be d_i . The minimum of d_i ($i=1, 2, \dots, m$) is defined as the distance between rectangle R and m rectangles R_1, R_2, \dots, R_m .

(6) Caving degree of COA

As shown in Fig. 4, if a rectangle R_i is packed into the box according to a feasible COA, let the distance between rectangle R_i and all the previously packed rectangles including the box (except the rectangles a and b those form this corner) be d_{\min} . The caving degree C_i of the corresponding COA is defined as follows:

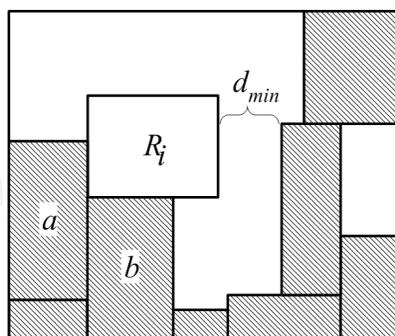


Fig. 4. Caving degree of COA.

$$C_i = 1 - \frac{d_{\min}}{\sqrt{w_i \cdot h_i}},$$

where w_i and h_i is the width and height of R_i respectively. The caving degree reflects the closeness between the rectangle to be packed and the previously packed rectangles

including the box (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 less than 1 when just occupies a corner formed by two previously packed rectangles.

(7) Corner degree of COA

For a given COA, the number of corners occupied by the related rectangle is defined as *corner degree* of the corresponding COA. For example, as shown in Fig. 5, the shadowy rectangles have been packed. If rectangle "1" is situated at place A, it occupies the corner formed by rectangles *a* and *b*. Then, the corner degree of the corresponding COA equals 1. If situated at place B, it occupies two corners. One is formed by rectangle *b* and the bottom boundary and, the other is formed by rectangle *c* and the bottom boundary. Thus, the corner degree of the corresponding COA equals 2. If situated at place C, it occupies 4 corners which are formed by rectangle *d* and *e*, *d* and *f*, *e* and right boundary, *f* and right boundary. In this situation, the corner degree of the corresponding COA becomes 4.

(8) Edge degree of COA

For a given COA, the number of edges that overlap with the related rectangle is defined as *edge degree* of the corresponding COA. For example, as shown in Fig. 6, the shadowy rectangles have been packed. If rectangle "1" is situated at place A, since it overlaps the left and top boundary and one edge of rectangle *a*, the edge degree of the corresponding COA equals 3. If situated at place B, the edge degree of the corresponding COA equals 2 for overlapping one edge of rectangle *b* and *c*, respectively. If situated at place C, the edge degree of the corresponding COA equals 5 for overlapping the bottom boundary and one edge of rectangle *c*, *d*, *e* and *f*, respectively.

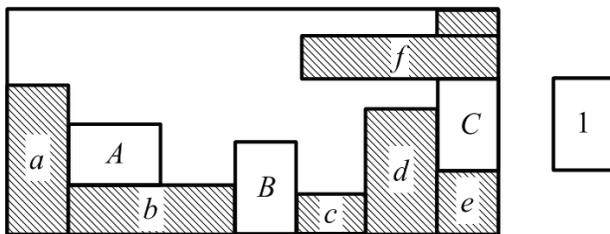


Fig. 5. Corner degree

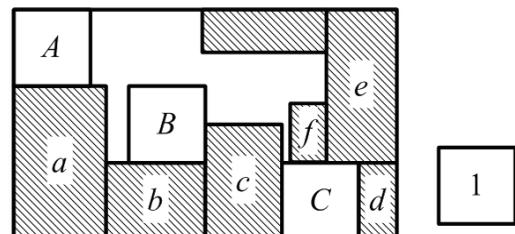


Fig. 6. Edge degree

(9) Precedence of point

Let $P_1(x_1, y_1)$ 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. Sketch of the algorithm

At each step, do the COA with the largest caving degree (if there are more than one COA with the largest caving degree, do the COA with the largest corner degree, if there are still multiple choices, do the COA with the largest edge degree) until no rectangle is left outside the box or no feasible COA can be done according to the current configuration.

In fact, this describes a greedy packing process. On the basis of greedy packing process, we introduce backtracking process, and so, develop the greedy packing algorithm into a new algorithm which can achieve better solution than greedy packing algorithm.

4. Computing program

4.1. Selecting rule

Rule 1. Select the COA with the largest caving degree, if there is more than one COA satisfying the condition, then:

Rule 2. Select the COA with the largest corner degree based on rule 1, if there is more than one COA satisfying the condition, then:

Rule 3. Select the COA with the largest edge degree on the basis of rule 2, if there is more than one COA satisfying the condition, then:

Rule 4. Select the COA with the highest precedence of the left-bottom vertex of the corresponding rectangle, if there is more than one COA satisfying the condition, then:

Rule 5. Select the COA with the corresponding rectangle packed with the longer sides horizontal if both the horizontal and vertical packings are feasible, if there is more than one COA satisfying the condition, then:

Rule 5. Select the COA with the smallest index of the corresponding rectangle.

4.2. Basic program

Step 1. If there is no feasible COA under the current configuration, output the unpacked area and stop the program. Otherwise, enumerate all feasible COAs, and then calculate the caving degree, corner degree and edge degree for each COA.

Step 2. Select a COA according to the selecting rule (see section 4.1) and pack the corresponding rectangle. Then reach a new configuration.

Step 3. If all rectangles have been packed into the box, output the packing result and stop successfully. Otherwise, return to step 1.

4.3. Strengthened program

Step 1. If there is no feasible COA under the current configuration, stop the program. Otherwise, enumerate all feasible COAs as candidates.

Step 2. For each candidate COA, pseudo-pack ("pseudo-pack" means to pack the rectangle into the box temporarily which will be removed from the box in the future) the corresponding rectangle and reach a new configuration. Based on this new configuration, pseudo-pack the remainder rectangles according to the *basic program*. If all rectangles have already been packed, output the packing result and stop successfully. Otherwise, calculate the area usage of the box according to the tentative end configuration as the score of the corresponding candidate COA.

Step 3. Select the COA with the highest score and pack the corresponding rectangle. Then reach a new configuration and return to step 1. If there are multiple COAs with the highest score, go to step 4.

Step 4. Select the COA according to the selecting rule (see section 4.1) and pack the corresponding rectangle. Then reach a new configuration and return to step 1.

4.4. Computational complexity

As each iteration of packing will occupy one or more corners and generate some new corners. The number of corners left should be proportional to n^2 . Therefore, for each rectangle to be packed, the number of COAs generated will be bounded by $O(n^3)$. For *basic program*, the process is repeated once for each rectangle packed. As a result, the worst-case

time complexity of *basic program* will be $O(n^3 \times n) = O(n^4)$. For *strengthened program*, the *basic program* is repeated $O(n^3)$ times for each rectangle packed. So the worst-case time complexity of *strengthened program* will be $O(n^4 \times n^3 \times n) = O(n^8)$. It should be noted that the time complexity of our algorithm is polynomial and relative small compared with the exponential time complexity of the original problem.

5. Experimental results

Two group benchmarks taken from Hopper & Turton (2001) and Burke et al. (2004) are used to test the performance of the algorithm proposed in this paper. The first group has 21 instances with the number of rectangles ranging from 16 to 197. The second group includes 13 instances with the number of rectangles ranging from 10 to 3152. The optimal solutions of these two groups are known.

5.1 21 rectangle-packing instances provided by Hopper and Turton

The performance of QHA has been tested with 21 rectangle-packing test instances taken from Hopper & Turton (2001). For each instance, the optimal solution is perfect, i.e., all rectangles can be packed into the box without overlapping area, the area usage of the box is 100%, and the unutilized area is zero. For more details about these instances, please refer to Hopper & Turton (2001).

Wu et al. (2002), Hopper & Turton (2001), Zhang et al. (2005) and Huang et al. (2007) reflect the most advanced algorithms that have already been published up to now. Heuristic1 (Wu et al., 2002) is based on the conception of flexibility; SA+BLF (Hopper & Turton, 2001) means simulated annealing+bottom left fill, GA+BLF (Hopper & Turton, 2001) means genetic algorithm+bottom left fill; hybrid heuristic (HH) (Zhang et al., 2005) is based on divide-and-conquer; and heuristic for rectangle packing (HRP) (Huang et al., 2007) is based on corner-occupying action and caving degree. Heuristic1, SA+BLF, GA+BLF, HH and HRP are not implemented in this paper, so the results are directly taken from Wu et al. (2002), Hopper & Turton (2001), Zhang et al. (2005) and Huang et al. (2007). Heuristic1 is run on a SUN Sparc20/71 with a 71MHz SuperSparc CPU and 64MB RAM; SA+BLF and GA+BLF are run on a Pentium pro with a 200MHz processor and 65MB RAM; HH is run on a Dell GX260 with a 2.4GHz CPU; QHA and HRP are run on IBM notebook PC with a 2.0GHz processor and 256MB memory. As an example, the packing results of instances 2, 5, 8, 14, 17 and 20 achieved by QHA are shown in Fig. 7.

For 21 rectangle-packing test instances, optimal solutions of 19 ones are achieved by QHA, i.e., all rectangles are packed into the box without overlapping area, the area usage of the box is 100%, and percent (%) of unpacked area, which is defined by $100(\text{box area} - \text{total area of already packed rectangles})/\text{box area}$, is 0%. And optimal solutions of 2, 3 and 16 ones are achieved by heuristic1, HH and HRP, respectively. The comparisons of the results of 21 instances between HRP, heuristic1, HH and QHA are listed in table 1. From table 1, we see that the runtime of QHA on some larger instances is shorter than that on some smaller instances because the program stops successfully when all rectangles are packed into the box without overlapping. As a result, it is natural that the runtime of QHA on instance 20 is much shorter than that on instance 19 and 21, as shown in table 1.

The original problem can be equivalently described as to minimize the container height under packing all rectangles without overlapping into a fixed width rectangular container.

This is so-called strip packing. In this paper, the minimal container height is calculated by QHA for each of 21 instances. The optimal solutions of all 21 ones except instances 19 and 21 are achieved by QHA. The best solution (i.e., the minimal box height) of instance 19 and 21 achieved by QHA is listed in table 2. Comparisons of the relative distance of the best solution to optimal solution (%) which is defined by $100(\text{best solution} - \text{optimal solution})/\text{optimal solution}$ and the runtime (min) between SA+BLF (Hopper & Turton, 2001), GA+BLF (Hopper & Turton, 2001) and QHA are listed in table 3, where the relative distance and runtime is the average value of three instances, as shown in table 3.

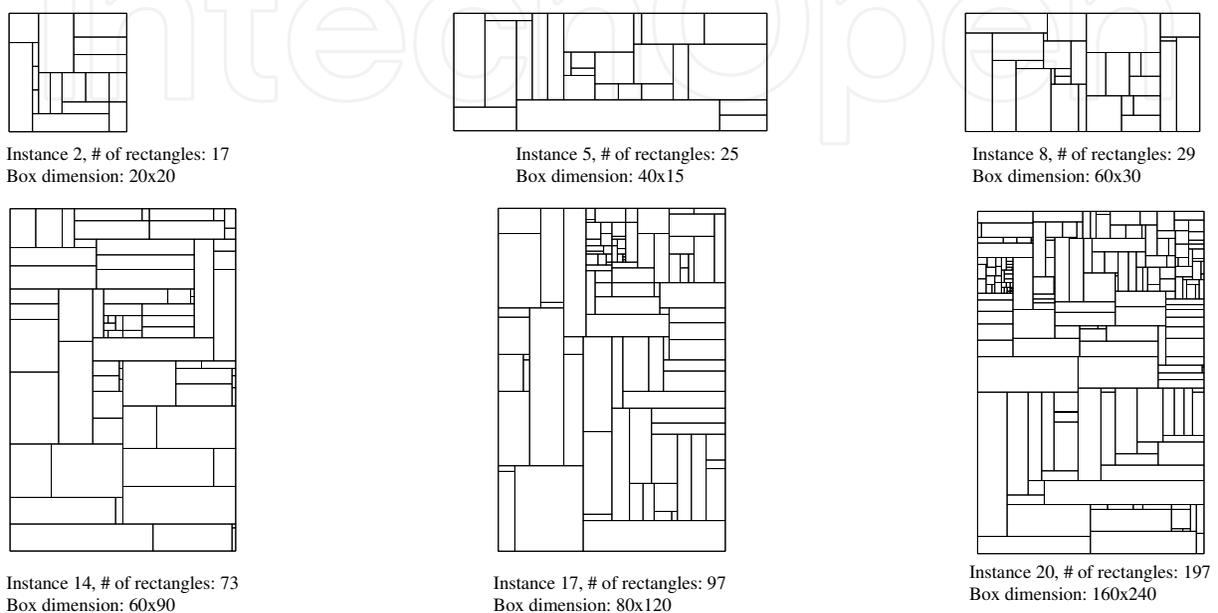


Fig. 7. The packing results of instances 2, 5, 8, 14, 17 and 20

5.2 13 random instances provided by Burke et al

We also use 13 random instances¹ provided by Burke et al. (2004) to test our algorithm. The comparisons of the box height and runtime between BF heuristic (Burke et al., 2004) and QHA are listed in table 4. For these 13 instances, optimal solutions of 3 ones are achieved by QHA, but none of them by BF heuristic. For each of the 13 instances, the container height obtained by QHA is smaller than that by BF heuristic, as shown in columns 4 and 6 of table 4. From table 4, we can see that the integrated performance of QHA is also rather satisfying for random instances. As an example, the packing result of instance N13 is shown in Fig. 8.

6. Conclusion

In this paper, an efficient quasi-human heuristic algorithm (QHA) for solving rectangle-packing problem is proposed. High area usage of the box can be obtained by this algorithm. Optimal solutions of 19 of 21 test instances taken from Hopper & Turton (2001) and 3 of 13 instances taken from Burke et al. (2004) are achieved by QHA. The experimental results demonstrate that QHA is rather efficient for solving the rectangle-packing problem. We guess the quasi-human approach will be fruitful for solving other NP-hard problems.

¹ The 13th instance N13 is generated on the basis of the 20th instance in Hopper & Turton (2001).

Instance	# of rectangles	Box dimensions (wxh)	QHA		HRP (Huang et al., 2007)		heuristic1 (Wu et al., 2002)		HH (Zhang et al., 2005)	
			% of unpacked area	Runtime (sec)	% of unpacked area	Runtime (sec)	% of unpacked area	Runtime (sec)	% of unpacked area	Runtime (sec)
1	16	20 x 20	0	0.02	0	0.05	2	1.48	2	0
2	17	20 x 20	0	0.22	0	0.23	2	2.42	3.5	0
3	16	20 x 20	0	0.04	0	1.12	2.5	2.63	0	0
4	25	40 x 15	0	0.3	0	0.08	0.67	13.35	0.67	0.05
5	25	40 x 15	0	0.09	0	0.1	0	10.88	0	0.05
6	25	40 x 15	0	0.05	0	0.28	0	7.92	0	0
7	28	60 x 30	0	1.16	0	2.58	0.67	23.72	0.67	0.05
8	29	60 x 30	0	7.77	0	4.19	0.83	34.02	2.44	0.05
9	28	60 x 30	0	2.51	0	2.5	0.78	30.97	1.56	0.05
10	49	60 x 60	0	265.58	0	327.12	0.97	438.18	1.36	0.44
11	49	60 x 60	0	20.13	0	36.59	0.22	354.47	0.78	0.44
12	49	60 x 60	0	20.78	0	135.6	No report	No report	0.44	0.33
13	73	60 x 90	0	72.09	0	55.44	0.3	1417.52	0.44	1.54
14	73	60 x 90	0	5.25	0	29.17	0.04	1507.52	0.44	1.81
15	73	60 x 90	0	38.34	0	51.13	0.83	1466.15	0.37	2.25
16	97	80 x 120	0	1610	0.15	873.38	0.25	7005.73	0.66	5.16
17	97	80 x 120	0	86.29	0	327.61	3.74	5537.88	0.26	5.33
18	97	80 x 120	0	490.81	0.06	577.59	0.54	5604.7	0.5	5.6
19	196	160 x 240	0.04	8303.13	0.24	4276.82	No report	No report	1.25	94.62
20	197	160 x 240	0	1520.27	0.12	3038.6	No report	No report	0.55	87.25
21	196	160 x 240	0.13	9288.21	0.16	3980.65	No report	No report	0.69	78.02

Table 1. Experimental results of 21 instances provided by Hopper & Turton (2001) on HRP, heuristic1, HH and QHA

Instance	# of rectangles	Box width	Optimal solution	Best solution	Runtime(sec)
19	196	160	240	241	8304.89
21	196	160	240	241	9291.44

Table 2. The best solution of instance 19 and 21 under packing all rectangles into the box

Instance	1, 2, 3	4, 5, 6	7, 8, 9	10, 11, 12	13, 14, 15	16, 17, 18	19, 20, 21
QHA (Relative distance, %/ runtime, 0/<0.01 min)	0/<0.01	0/<0.01	0/0.06	0/1.70	0/0.64	0/12.15	0.28/ 106.20
SA+BLF(Hopper & Turton, 2001) (Relative distance, %/ runtime, min)	4/0.7	6/2.4	5/4	3/33	3/115	3/382	4/4181
GA+BLF(Hopper & Turton, 2001) (Relative distance, %/ runtime, min)	4/1	7/2	5/3	3/13	4/36	4/86	5/777

Table 3. Comparisons of the relative distance of best solution to optimal solution (%) and the runtime (min) between SA+BLF, GA+BLF and QHA

Instance	# of rectangles	Optimal height	BF heuristic (Burke et al., 2004)		QHA	
			Box height	Runtime (sec)	Box height	Runtime (sec)
N1	10	40	45	<0.01	40	0.17
N2	20	50	53	<0.01	50	1.73
N3	30	50	52	<0.01	50	2.91
N4	40	80	83	<0.01	81	105.53
N5	50	100	105	0.01	102	134.04
N6	60	100	103	0.01	101	137.92
N7	70	100	107	0.01	101	423.73
N8	80	80	84	0.01	81	865.96
N9	100	150	152	0.01	151	1242.77
N10	200	150	152	0.02	151	7912.36
N11	300	150	152	0.03	151	1.5 × 10 ⁴
N12	500	300	306	0.06	303	4.4 × 10 ⁴
N13	3152	960	964	1.37	962	5.2 × 10 ⁵

Table 4. Comparisons of the box height and runtime between BF heuristic and QHA

7. Acknowledgments

This work was partially supported by National Natural Science Foundation of China under Grant No. 10471051, by NKBRPC (2004CB318000), by National High-tech R&D Program (2006AA01Z414, 2007AA01Z440), by Research and Development Project on Application Technology of Sichuan under Grant No. 2008GZ0009. and by Chinese Information Security Research Plan under Grant No. (242)2007B27.

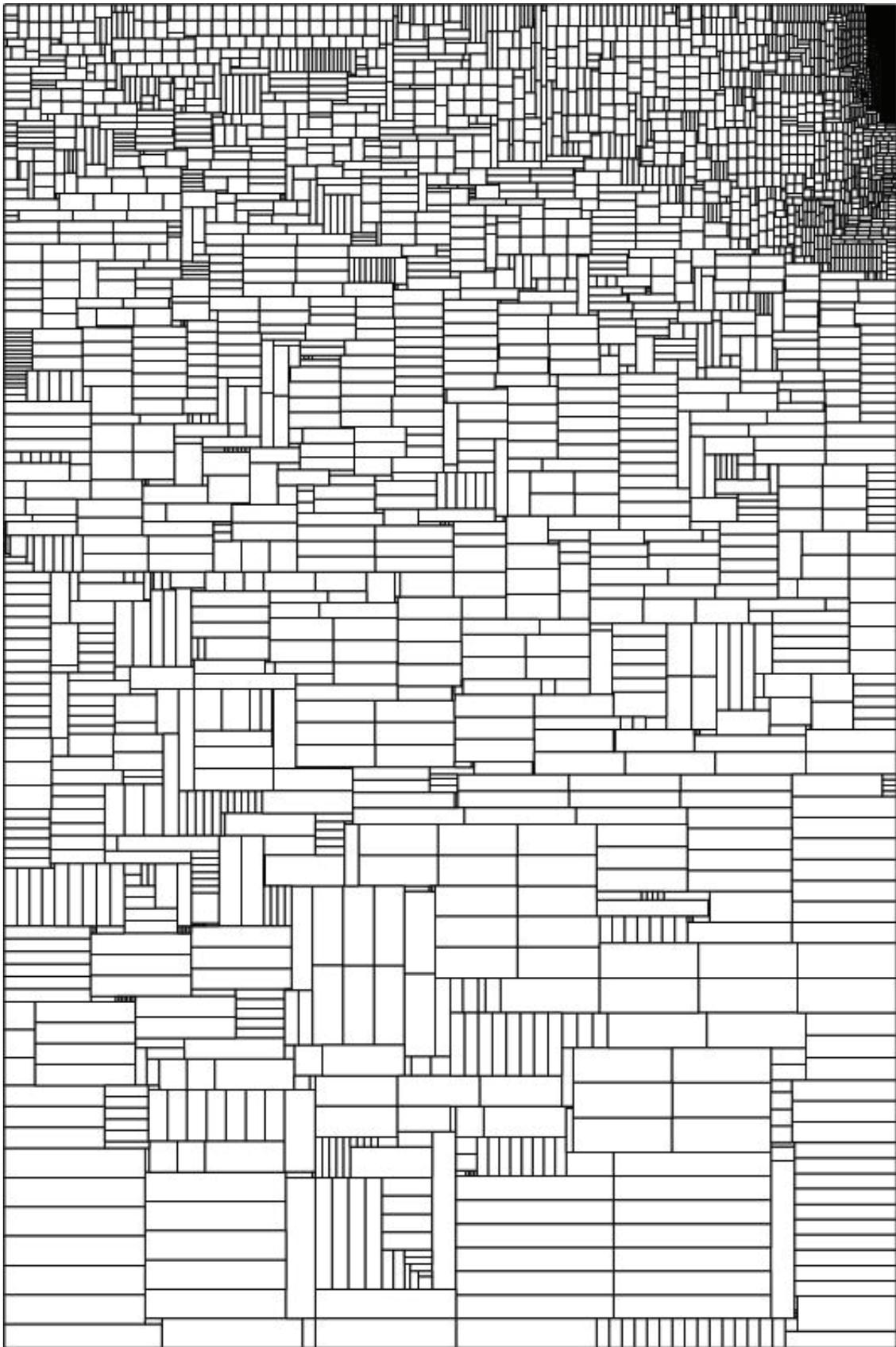
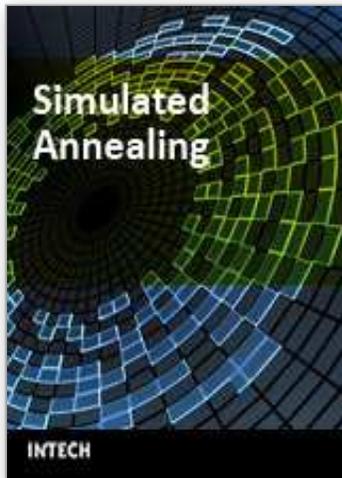


Fig. 8. The packing result of instance N13 by QHA

8. References

- Leung J., Tam T., Wong C.S., Young G. & Chin F. (1990), Packing squares into a square, *Journal of Parallel and Distributed Computing*, Vol. 10, No. 3, November 1990, pp. 271-275.
- Lodi A., Martello S. & Monaci M. (2002), Two-dimensional packing problems: A survey, *European Journal of Operational Research*, Vol.141, No. 2, September 2002, pp. 241-252.
- Pisinger D. (2002), Heuristic for the container loading problem, *European Journal of Operational Research*, Vol. 141, No. 2, September 2002, pp. 382-392.
- Hopper E. & Turton B. (1999), A genetic algorithm for a 2D industrial packing problem, *Computers & Industrial Engineering*, Vol. 37, No. 1-2, October 1999, pp. 375-378.
- Bortfeldt A. (2006), A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces, *European Journal of Operational Research*, Vol. 172, No. 3, August 2006, pp. 814-837.
- 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*, Vol. 141, No. 2, September 2002, pp. 341-358.
- Beasley J. E. (1985), An exact two-dimensional non-guillotine cutting tree search procedure, *Operations Research*, Vol. 33, No. 1, January 1985, pp. 49-64.
- Liu D. & Teng H. (1999), An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles, *European Journal of Operational Research*, Vol. 112, No.2, January 1999, pp. 413-420.
- 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*, Vol. 145, No. 3, March 2003, pp. 530-542.
- Lodi A., Martello S. & Vigo D. (1999), Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems, *INFORMS Journal on Computing*, Vol. 11, No.4, April 1999, pp. 345-357.
- Hopper E. & Turton B. (2001), An empirical investigation of meta-heuristic and heuristic algorithm for a 2D packing problem, *European Journal of Operational Research*, Vol. 128, No.1, January 2001, pp. 34-57.
- Zhang D., Deng A. & Kang Y. (2005), A hybrid heuristic algorithm for the rectangular packing problem, in: *Proceedings of the 5th International Conference on Computational Science, Atlanta, GA, USA, May 22-25, 2005, part I, Lecture Notes in Computer Science*, Vol. 3514, pp. 783-791.
- Burke E. K., Kendall G. & Whitwell G. (2004), A new placement heuristic for the orthogonal stock-cutting problem, *Operations Research*, Vol. 52, No. 4, July/August 2004, pp. 655-671.
- Huang W. & Jin R. (1997), The quasi-physical and quasi-sociological algorithm solar for solving SAT problem, *Science in China, Series E (Technological Sciences)*, Vol. 27, No.2, April 1997, pp. 179-186.
- Huang W. & Xu R. (1999), Two personification strategies for solving circles packing problem, *Science in China, Series E (Technological Sciences)*, Vol. 42, No. 6, December 1999, pp. 595-602.
- Huang W., Li Y., Akeb H. & Li C. (2005), Greedy algorithms for packing unequal circles into a rectangular container, *Journal of the Operational Research Society*, Vol. 56, No. 5, May 2005, pp. 539-548.
- Huang W., Chen D. & Xu R. (2007), A new heuristic algorithm for rectangle packing, *Computers & Operations Research*, Vol.34, No.11, November 2007, pp. 3270-3280.



Simulated Annealing

Edited by Cher Ming Tan

ISBN 978-953-7619-07-7

Hard cover, 420 pages

Publisher InTech

Published online 01, September, 2008

Published in print edition September, 2008

This book provides the readers with the knowledge of Simulated Annealing and its vast applications in the various branches of engineering. We encourage readers to explore the application of Simulated Annealing in their work for the task of optimization.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Wenqi Huang and Duanbing Chen (2008). An Efficient Quasi-Human Heuristic Algorithm for Solving the Rectangle-Packing Problem, Simulated Annealing, Cher Ming Tan (Ed.), ISBN: 978-953-7619-07-7, InTech, Available from: http://www.intechopen.com/books/simulated_annealing/an_efficient_quasi-human_heuristic_algorithm_for_solving_the__rectangle-packing_problem

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen