BULETINUL ACADEMIEI DE ŞTIINŢE
A REPUBLICII MOLDOVA. MATEMATICA
Number 2(66), 2011, Pages 81–88
ISSN 1024–7696

A heuristic algorithm for the non-oriented 2D rectangular strip packing problem

V. M. Kotov*, Dayong Cao

Abstract. In this paper, we construct best fit based on concave corner strategy (BF_{BCC}) for the two-dimensional rectangular strip packing problem (2D-RSPP), and compare it with some heuristic and metaheuristic algorithms from the literature. The experimental results show that BF_{BCC} could produce satisfied packing layouts, especially for the large problem of 50 pieces or more, BF_{BCC} could get better results in shorter time.

Mathematics subject classification: 34C05.
Keywords and phrases: Rectangular strip packing, Heuristic, Best-Fit, Concave corner.

1 Introduction

Two-dimensional packing problem (2D-PP) has been proved to be NP-hard according to the combinatorial explosion with the problem size increasing [1]. Because the applications of the problem in business and industry are very extensive, during recent years, many researchers have provided various methods to process it. These methods could be broadly categorized into three kinds: exact algorithm, heuristic algorithm and metaheuristic algorithm [3].

Some exact algorithms could be found in [5–8]. A major drawback of these methods in that they can not provide good results for large instances of the problem [2].

During recent years, many heuristic packing algorithms have been suggested in the literature. Surveys on these methodologies for various types of the 2D rectangular packing problem could be found in [4], and these heuristic algorithms could produce good packing layout in an acceptable time, especially for large problems. The most documented heuristic approaches are the bottom-left (BL) [9] and bottom-left-fill (BLF) methods [10]. Based on these two methods, many improved algorithms have been presented in literature, see [2, 11, 12].

Now, metaheuristic algorithms have been important methods in producing packing layout for 2D-PP. These are usually hybridized algorithms involving the generation of input sequences interpreting with placement heuristics such as BL, BLF and other placement strategies, see [2, 4, 13, 14].

*This author is corresponding author

In this paper, we construct *best fit based on concave corner* strategy (BF_{BCC}) for the *two-dimensional rectangular strip packing problem* (2D-RSPP), and compare it with some heuristic and metaheuristic algorithms from the literature. According to the category in [15], the problem belongs to the type of RF: the items could be rotated by 90° and no guillotine constraint is proposed. The experimental results show that the BF_{BCC} is an efficient heuristic algorithm for 2D-RSPP.

2 The problem

Two-dimensional rectangular strip packing problem (2D-RSPP) could be formulated as follows: Let W denote the width of the strip with infinite height, and $P = \{p_i(w_i, h_i), i = 1, 2, \cdots, n\}$ be a set of n rectangular pieces. Each piece p_i has width w_i and height h_i ($w_i, h_i \in Z^+$) with at least one edge no bigger than the W. The objective of the algorithm is to pack all pieces onto the strip orthogonally and pieces could be rotated by 90°, at the same time, try to minimize the used height h of the strip with no two pieces overlap, see Figure 1.

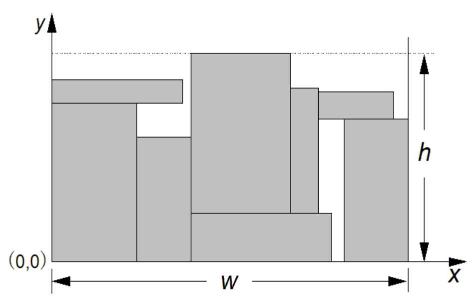

Figure 1. Two-dimensional rectangular strip packing problems

3 A new heuristic algorithm for the 2D-RSPP(BF_{BCC})

3.1 Definitions

For describing the algorithm expediently, we put the strip in the Cartesian system of coordinates, let the left-bottom vertex be superposed over the origin of the coordinate, and the right-bottom vertex in the x-axis, see Figure 1.

1. Let C_i denote the "*Concave Corner*"(CC). The CC is constructed by two edges, and the size of the angle is 90°, at the same time, the CC does not belong to any packed pieces and the corner direction is left-top or right-top.

2. The CC includes two kinds, one is "*Real Concave Corner*(RCC)" with all edges belonging to some packed pieces or the strip, it is denoted as C^+. The other kind is "*Sham Concave Corner*(SCC)" with at least one edge of the corner being the elongation line of the edges of some packed pieces, the SCC is denoted as C^-, see Figure 2.

3. Define $U = \{C_1(xC_1, yC_1), C_2(xC_2, yC_2), \cdots, C_n(xC_n, yC_n)\}$ to denote a set of *Concave Corner* before packing a piece, here, the (xC_i, yC_i) is the coordinate of the vertex of the C_i. Obviously, every C_i is a candidate position for the new piece.

4. *Temp packed height* (TH): After packing a piece onto the strip, the used height of the strip should be computed, which is denoted as TH for current state of the strip.

5. For a piece $p_i(w_i, h_i)$, define $W\_lU = \{C_i | yC_i + h_i \leq TH, w_i \geq h_i\}$, $W\_hU = \{C_i | yC_i + h_i > TH, w_i \geq h_i\}$, $H\_lU = \{C_i | yC_i + h_i \leq TH, w_i < h_i\}$ and $H\_hU = \{C_i | yC_i + h_i > TH, w_i < h_i\}$.

6. *Fitness value* of C_j for one piece: A new piece p_i with $w_i \geq h_i$ is packed onto the board at the position C_j, diagnosing whether the p_i intersects with some packed pieces or the edges of the strip, if the p_i could be deposited. Let s denote the number of edges which is touched with some packed pieces and t denotes the number of concave corner which has been occupied by the piece p_i. Then we compute the parameter $W\_pFit\_C_j$ using formula $W\_pFit\_C_j = 2s + \sum_{k=1}^{t} q_k$.

If piece p_i could be packed onto the strip with corner of the piece at C_j, then s should be computed by querying all packed pieces in the strip. After that check every CC in U: if p_i occupies a *real concave corner* then q_k is equal to 2; if p_i occupies a *sham concave corner* then q_k is equal to 1. Similarly, we could define the $H\_pFit\_C_j$ when $h_i > w_i$.

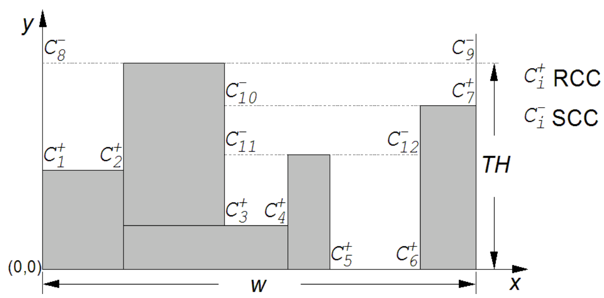

Figure 2. RCC and SCC

3.2 *Best fit based on concave corner* (BF_{BCC}) placement strategy

BF_{BCC} heuristic initially adjusts every p_i in P such that $w_i \geq h_i$ and puts all p_j with w_j bigger than the width of the strip into the front of the packing sequence P, the *startPosition* denotes the number of such pieces. Then BFF_{BCC} sorts the P from the position with subscript *startPosition* to the end by non-increasing height (resolving equal height by non-increasing width).

Before any piece is packed onto the strip, the $U = \{C_1(0, 0), C_2(W, 0)\}$ should be initialized , here C_1 is the left-bottom **concave corner** of the strip and C_2 is the right-bottom **concave corner**.

When a piece p_i being packed onto the board, the W_lU is computed if $|W\_lU| > 0$, namely, there exist positions for p_i such that the p_i does not exceed the TH and the

W_hU should not be computed, otherwise, computes the W_hU. Then exchanges the w_i with the h_i and computes H_lU, if $|H\_lU|$ is zero, then the H_hU should be computed, otherwise, H_hU need not, the following rules decide the position for the piece p_i.

Selecting Rules:

1) if $|W\_lU| > 0$ and $|H\_lU| > 0$, let W_l_{best} denote the position C_j with highest fitness value $W\_pFit\_C_j$ and lowest $yC_j + h_i$, if the positions satisfying this condition are more than one, then select the first. Then change the w_i with h_i such that $w_i < h_i$ and then the $p_i(w_i, h_i)$ would be denoted as $p_i^{'}(w_i^{'}, h_i^{'})$, let H_l_{best} denote the position C_k with highest fitness value $H\_pFit\_C_k$ and lowest $yC_k + h_i^{'}$. If $W\_pFit\_C_j > H\_pFit\_C_k$, then select the position W_l_{best} as the best position for $p_i(w_i, h_i)$, if $W\_pFit\_C_j < H\_pFit\_C_k$, then select the H_l_{best} as the best position for $p_i^{'}(w_i^{'}, h_i^{'})$, if $W\_pFit\_C_j$ equals to $H\_pFit\_C_k$, then select the position with minimal value between $yC_j + h_i$ and $yC_k + h_i^{'}$, if $yC_j + h_i = yC_k + h_i^{'}$ then select the position with minimal value between yC_j and yC_k.

2) if $|W\_lU| > 0$ and $|H\_lU| = 0$, select the position C_j with highest fitness value $W\_pFit\_C_j$ and lowest $yC_j + h_i$ as the best position for the $p_i(w_i, h_i)$.

3) if $|W\_lU| = 0$ and $|H\_lU| > 0$, exchange the w_i with h_i such that $w_i < h_i$, then select the position C_k with highest fitness value $H\_pFit\_C_k$ and lowest $yC_k + h_i^{'}$ as the best position for the piece $p_i^{'}(w_i^{'}, h_i^{'})$.

4) if $|W\_lU| = 0$ and $|H\_lU| = 0$, select the position C_t from W_hU and H_hU with lowest $yC_t + h_i$ and yC_t.

After selecting the best position for the $p_i(w_i, h_i)$ and packing it, updates the TH and U according to the current packing layout of the strip.

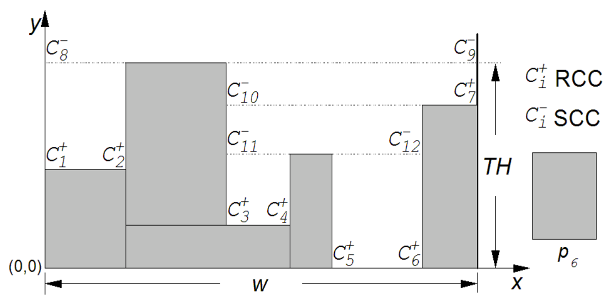

Figure 3. An example of BF_{BCC} placement strategy

An example is given in Figure 3: before the piece p_6 arrival, the $U = \{C_1^+, C_2^+, \cdots, C_7^+, C_8^-, C_9^-, \cdots, C_{12}^-\}$, which means that there exist 12 candidate positions for the piece p_6. By checking every C_i in U, we could get $W\_lU = \{C_1^+, C_2^+, C_5^+, C_6^+, C_{11}^-, C_{12}^-\}$, then exchange w_i with h_i, we have $H\_lU = \{C_1^+, C_2^+, C_3^+, C_4^+, C_5^+, C_6^+, C_{11}^-, C_{12}^-\}$, according to the **rules 1** with $|W\_lU| > 0$ and $|H\_lU| > 0$, the W_hU and the H_hU need not be computed. Then compute *Fitness value* for every "*Concave Corner*" in W_lU using formula(1), we have: $W\_pFit\_C_1^+ = W\_pFit\_C_2^+ = 10$, $W\_pFit\_C_5^+ = W\_pFit\_C_6^+ = 6$, $W\_pFit\_C_{11}^- = 5$, $W\_pFit\_C_{12}^- =$

3, so $W\_l_{best} = W\_pFit\_C_1^+$. Then exchange w_i with h_i such that $h_i > w_i$ and piece p_6 we denoted $p_6^{'}(w_6^{'}, h_6^{'})$. After that we have $H\_pFit\_C_3^+ = H\_pFit\_C_4^+ = 10$, $H\_pFit\_C_5^+ = H\_pFit\_C_6^+ = 6$, $H\_pFit\_C_{11}^- = 3$, $H\_pFit\_C_{12}^- = 3$, so $H\_l_{best} = C_3^+$. Here we found the fitness of W_l_{best} and H_l_{best} that are same, but $yC_1^+ + h_6 > yC_3^+ + h_6^{'}$. So, we select C_3^+ as the best position for the piece $p_6^{'}(w_6^{'}, h_6^{'})$. After packing the piece p_6, U should be updated, but TH needn't be updated because $yC_3^+ + h_6^{'}$ is no more than TH.

The whole placement heuristic algorithm could be described as Algorithm 1.

Algorithm 1 heuristic packing (packing sequence P, strip width $stripWidth$)

adjusts every $p_i(w_i, h_i)$ in P such that $w_i \geq h_i$;
$startPosition \Leftarrow 0$;
for $i = 1$ to $|P|$ **do**
  if $w_i \geq stripWidth$ **then**
    exchanges p_i with $p_{startPosition}$;
    $startPosition \Leftarrow startPosition + 1$;
  end if
end for
sorts the P by non-increasing height (resolving equal height by non-increasing width) from the position with subscript $startPosition$ to the end;
$TH \Leftarrow 0$;
j $\Leftarrow$ 0;
$U \Leftarrow \{C_1(0, 0), C_2(stripWidth, 0)\}$;
while packing sequence P is not null **do**
  $W\_lU \Leftarrow \varnothing$, $W\_hU \Leftarrow \varnothing$;
  $H\_lU \Leftarrow \varnothing$, $H\_hU \Leftarrow \varnothing$;
  gets p_j from the packing sequence P;
  if w_j is bigger than $stripWidth$ **then**
    exchanges w_j with h_j;
  end if
  computes the W_lU, W_hU, H_lU and H_hU based on the **definitions** mentioned above;
  selects the best position C_s for p_j from W_lU, W_hU, H_lU and H_hU according to the **selecting rules**;
  packs the p_j onto the board at the position C_s;
  removes the p_j from packing sequence P;
  if the used height of current strip is exceeded than TH **then**
    updates TH;
  end if
  updates U such that U includes all CC at the current state;
  j $\Leftarrow$ j + 1;
end while

Obviously, BF_{BCC} algorithm includes the Bottom-left(BL), Best-fit-fill(BLF) algorithms etc, and it could process the "hole" easily.

4 Experiments

The test program has been coded in c++ language and run on a IBM T400 notebook PC with 2.26 GHZ CPU and 2048 MB RAM, test data coming from [2, 4] are used to compare the BF_{BCC} with some heuristic and metaheuristic algorithms. All test results except BF_{BCC} are obtained from [2], which is performed on a pc with 850 MHz CPU and 128 MB RAM. For all instances, the best solutions are shown in bold type.

Table 1 shows that the BF_{BCC} outperforms Bottom-Left, Bottom-Left-Fill and Best-Fit [2] in almost all test data, even when preordering is allowed (DW means "decreasing width" and DH means "decreasing height). The computational results of metaheuristic approaches (GA+BLF, SA+BLF) and Best-Fit [2] could be found in Table 2. We can see that BF_{BCC} could gives better results than $GA + BLF$, $SA + BLF$ and $Best - Fit$ quickly.

Table 1. Comparison of the BF_{BCC} heuristic with some heuristic algorithm (% over optimal)

Category:	C1			C2			C3			C4			C5			C6			C7		
Problem:	P1	P2	P3	P1	P2	P3	P1	P2	P3	P1	P2	P3	P1	P2	P3	P1	P2	P3	P1	P2	P3
Number:	16	17	16	25	25	25	28	29	28	49	49	49	72	73	72	97	97	97	196	197	196
BL	45	40	35	53	80	67	40	43	40	32	37	30	27	32	30	33	39	34	22	41	31
BL-DW	30	20	20	13	27	27	10	20	17	17	22	22	16	18	13	22	25	18	16	19	17
BL-DH	15	10	**5**	13	73	13	10	10	13	12	13	6.7	4.4	10	7.8	8.3	8.3	9.2	5	10	7.1
BLF	30	35	25	47	73	47	37	50	33	25	25	27	20	23	21	20	18	21	15	20	17
BLF-DW	10	15	15	13	20	20	10	13	13	10	5	10	5.6	6.7	5.6	5	4.2	4.2	4.6	3.4	2.9
BLF-DH	10	10	**5**	13	73	13	10	**6.7**	13	10	5	5	4.4	5.6	4.4	5	2.5	6.7	3.8	2.9	3.8
BF	**5**	10	20	**6.7**	**6.7**	**6.7**	**6.7**	13	10	**5**	**3.3**	**3.3**	3.3	**2.2**	3.3	2.5	**1.7**	3.3	2.9	1.7	2.1
BF_{BCC}	**5**	**5**	10	**6.7**	**6.7**	**6.7**	**6.7**	**6.7**	**6.7**	**5**	**3.3**	**3.3**	**2.2**	3.3	**1.1**	**1.7**	**1.7**	**1.7**	**1.7**	**1.3**	**1.7**

5 Conclusion

In this paper, a new heuristic algorithm (BF_{BCC}) for no-oriented 2D-SP problem has been proposed. The approach is tested on a set of instances taken from the literature and compared with some heuristic algorithms (Bottom-Left, Bottom-Left-Fill and Best Fit) and some metaheuristic algorithms (GA + BLF and SA + BLF), the experimental results show that BF_{BCC} could produce better-quality packing layouts than these algorithms, especially for the large problem of 50 pieces or more, BF_{BCC} could get better results in shorter time.

Table 2. Comparison of the BF_{BCC} heuristic with BF and some metaheuristic methods (GA + BLF, SA + BLF)

					GA+BLF		SA+BLF		Best Fit		BF_{BCC}		GA_{best}-BF_{BCC}		SA_{best}-BF_{BCC}		BF-BF_{BCC}	
Data set	Cat.	Problem	Number	Optimal height	Best	Time(s)	Best	Time(s)	Sol.	Time(s)	Sol.	Time(s)	Abs.	%Impv.	Abs.	%Impv.	Abs.	%Impv.
Hopper	C1	P1	16	20	**20**	3.4	**20**	1.1	21	< 0.01	21	0.01	-1	-5.0	-1	-5.0	**0**	**0**
		P2	17	20	**21**	0.5	**21**	0.8	22	< 0.01	**21**	0.01	**0**	**0**	**0**	**0**	**1**	**4.5**
		P3	16	20	**20**	7.1	**20**	0.8	24	< 0.01	22	0.01	-2	-10.0	-2	-10	**2**	**8.3**
	C2	P1	25	15	**16**	1.3	**16**	6.5	**16**	< 0.01	**16**	0.02	**0**	**0**	**0**	**0**	**0**	**0**
		P2	25	15	**16**	2.2	**16**	13.9	**16**	< 0.01	**16**	0.03	**0**	**0**	**0**	**0**	**0**	**0**
		P3	25	15	**16**	1.0	**16**	13.6	**16**	< 0.01	**16**	0.02	**0**	**0**	**0**	**0**	**0**	**0**
	C3	P1	28	30	**32**	7.4	**32**	20.3	**32**	< 0.01	**32**	0.03	**0**	**0**	**0**	**0**	**0**	**0**
		P2	29	30	**32**	12.4	**32**	22.5	34	< 0.01	**32**	0.05	**0**	**0**	**0**	**0**	**2**	**5.9**
		P3	28	30	**32**	11.6	**32**	18.3	33	< 0.01	**32**	0.04	**0**	**0**	**0**	**0**	**1**	**3**
	C4	P1	49	60	64	35	64	65	**63**	< 0.01	**63**	0.16	**1**	**1.6**	**1**	**1.6**	**0**	**0**
		P2	49	60	63	48	64	46	**62**	< 0.01	**62**	0.13	**1**	**1.6**	**2**	**3.1**	**0**	**0**
		P3	49	60	**62**	61	63	70	**62**	< 0.01	**62**	0.14	**0**	**0**	**1**	**1.6**	**0**	**0**
	C5	P1	72	90	95	236	94	501	93	0.01	**92**	0.40	**3**	**3.2**	**2**	**2.1**	**1**	**1.1**
		P2	73	90	95	440	95	285	**92**	0.01	93	0.37	**2**	**2.1**	**2**	**2.1**	-1	-1.1
		P3	72	90	95	150	95	425	93	0.01	**91**	0.29	**4**	**4.2**	**4**	**4.2**	**2**	**2.2**
	C6	P1	97	120	127	453	127	854	123	0.01	**122**	0.67	**5**	**3.9**	**5**	**3.9**	**1**	**0.8**
		P2	97	120	126	866	126	680	**122**	0.01	**122**	0.64	**4**	**3.2**	**4**	**3.2**	**0**	**0**
		P3	97	120	126	946	126	912	124	0.01	**122**	0.61	**4**	**3.2**	**4**	**3.2**	**2**	**1.6**
	C7	P1	196	240	255	4330	255	4840	247	0.01	**244**	4.11	**11**	**4.3**	**11**	**4.3**	**3**	**1.2**
		P2	197	240	251	5870	253	5100	244	0.01	**243**	4.31	**8**	**3.2**	**10**	**4.0**	**1**	**0.4**
		P3	196	240	254	5050	255	6520	245	0.01	**244**	3.39	**10**	**3.9**	**11**	**4.3**	**1**	**0.4**
Burke		N1	10	40	**40**	1.02	**40**	0.24	45	< 0.01	44	< 0.01	-4	-10	-4	-10	**1**	**2.2**
		N2	20	50	**51**	9.2	52	8.14	53	< 0.01	54	0.02	-3	-0.59	-2	-3.8	-1	-0.3
		N3	30	50	**52**	2.6	**52**	39.5	**52**	< 0.01	54	0.04	-2	-3.8	-2	-3.8	-2	-3.9
		N4	40	80	**83**	12.6	**83**	84	**83**	< 0.01	**83**	0.11	**0**	**0**	**0**	**0**	**0**	**0**
		N5	50	100	106	52.3	106	228	**105**	0.01	106	0.15	**0**	**0**	**0**	**0**	-1	-1
		N6	60	100	103	261	103	310	103	0.01	**102**	0.19	**1**	**1**	**1**	**1**	**1**	**1**
		N7	70	100	106	671	106	554	107	0.01	**103**	0.35	**3**	**2.8**	**3**	**2.8**	**4**	**3.7**
		N8	80	80	85	1142	85	810	84	0.01	**82**	0.39	**3**	**3.5**	**3**	**3.5**	**2**	**2.4**
		N9	100	150	155	4431	155	1715	**152**	0.01	155	0.61	**0**	**0**	**0**	**0**	-3	-2.0
		N10	200	150	154	2×10^4	154	6066	**152**	0.02	**152**	2.79	**2**	**1.3**	**2**	**1.3**	**0**	**0**
		N11	300	150	155	8×10^4	155	3×10^4	**152**	0.03	154	6.68	**1**	**0.5**	**1**	**0.6**	-2	-0.7
		N12	500	300	313	4×10^5	312	6×10^4	**306**	0.06	**306**	26.88	**7**	**2.2**	**6**	**1.9**	**0**	**0**
		N13	3152	960	-	-	-	-	964	1.37	**962**	3291.20					**2**	**0.2**

References

[1] Garey M. R. Johnson D. S. and others. *Computers and Intractability: A Guide to the Theory of NP-completeness.* Freeman, San Francisco, 1979.

[2] Burke E. K., Kendall G., Whitwell G. *A new placement heuristic for the orthogonal stock-cutting problem.* Operations Research, 2004, No. 52(4), 655–671.

[3] Lodi A., Martello S., Monaci M. *Two-dimensional packing problems: A survey.* European Journal of Operational Research, 2002, No. 141(2), 241–252.

[4] Hopper E., Turton B. *An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem.* European Journal of Operational Research, 2001, No. 128(1), 34–57.

[5] Gilmore P. C., Gomory R. E. *A linear programming approach to the cutting stock problem-part II.* Operations Research, 1963, No. 11(6), 863–888.

[6] Christofides N. Whitlock C. *An algorithm for two-dimensional cutting problems.* Operations Research, 1977, No. 25, 30–44.

[7] Christofides N. Hadjiconstantinou E. *An exact algorithm for orthogonal 2-D cutting problems using guillotine cuts.* European Journal of Operational Research, 1995, No. 83(1), 21–38.

[8] Sweeney P. E., Paternoster E. R. *Cutting and packing problems: a categorized, application-orientated research bibliography.* The Journal of the Operational Research Society, 1992, No. 43(7), 691–706.

[9] Baker B. S., Coffman Jr, E. G., Rivest R. L. *Orthogonal packings in two dimensions.* SIAM Journal on Computing, 1980, No. 9, 846–855.

[10] Chazelle B. *The bottom-left bin-packing heuristic: An efficient implementation.* IEEE Transactions on Computers, 1983, No. 32(8), 697–707.

[11] Liu D., Teng H. *An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles.* European Journal of Operational Research, 1999, No. 112(2), 413–420.

[12] Hopper E., Turton B. *A genetic algorithm for a 2D industrial packing problem.* Computers & Industrial Engineering, 1999, No. 37(1-2), 375–378.

[13] Leung T. W., Chan C. K., Troutt M. D. *Application of a mixed simulated annealing-genetic algorithm heuristic for the two-dimensional orthogonal packing problem.* European Journal of Operational Research, 2003, No. 145(3), 530–542.

[14] Leung T. W., Yung C. H., Troutt M. D. *Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem.* Computers & industrial engineering, 2001, No. 40(3), 201–214.

[15] Lodi A., Martello S., Monaci M. *Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problem.* Informs Journal on Computing, 1999, No. 11, 345-357.

Received September 6, 2010

V. M. Kotov
Department of Applied Mathematics
and Computer Science
Belarusian State University, Minsk, Belarus
E-mail: *kotovvm@yandex.ru*

Dayong Cao
Department of Applied Mathematics
Harbin University of Science and Technology
XueFu road, 52, Harbin 150080 China
E-mail: *caodayong@hrbust.edu.cn*