ISSN 0005-1179, Automation and Remote Control, 2009, Vol. 70, No. 6, pp. 1054–1066. © Pleiades Publishing, Ltd., 2009.
Original Russian Text © V.M. Kartak, M.A. Mesyagutov, E.A. Mukhacheva, A.S. Filippova, 2009, published in Avtomatika i Telemekhanika, 2009, No. 6, pp. 153–166.



# Local Search of Orthogonal Packings Using the Lower Bounds[1]

**V. M. Kartak, M. A. Mesyagutov, E. A. Mukhacheva, and A. S. Filippova**

*Ufa State Technical University of Aviation, Ufa, Russia*

Received December 14, 2007

**Abstract**—Consideration was given to the problem of packing the rectangular half-infinite strip. It was suggested to solve it using a one-point evolutionary algorithm with search of the best solution in the neighborhood of the local lower bound. Proposed was an algorithm to construct this neighborhood including the packings obtained by solving a special problem of one-dimensional cutting. Its solution was shown to be the local lower bound in the considered neighborhood. An improved global lower bound was proposed. The results of numerical modeling were presented. The record value obtained was compared with the global bounds.

PACS number: 02.60.Pn

**DOI:** 10.1134/S0005117909060113

## 1. INTRODUCTION

Consideration was given to the problem of orthogonal packing of the set $\{w, l\}, w = (w_1, \ldots, w_m)$, $l = (l_1, \ldots, l_m)$, of rectangular pieces in a strip of width $W$ and half-infinite length, the so-called two-dimensional strip packing (2DSP). The values of $W$, $w_i$, $l_i$, $i = \overline{1, m}$, are assumed to be integers. An orthogonal arrangement of rectangles in a strip without overlapping each other and the strip edges is called the permissible rectangular packing (RP). It is required to minimize the packing length $L$ on the set of permissible RP's. Since in the strong sense it is an $\mathcal{NP}$-trying problem, it is solved using the heuristic and approximate algorithms among which we note the constructive heuristics and probabilistic methods of local search. The constructive heuristics determine permissible solutions in one pass. These can be simpler heuristics like the *fit* ones [1] used as the decoders of algorithms reproducing the packing by its code. Another type of constructive heuristics is represented by complex single-pass algorithms determining a permissible solution which is close to the optimal one [2]. The methods of local search are exemplified by the metaheuristics including the evolutionary algorithms such as the one-point random search algorithms (1+1)-EA [3], annealing simulation algorithms [4], or genetic algorithms [5, 6]. The scheme of computations by the evolutionary algorithms is as follows: the initial permissible solution and its neighborhood (the set of permissible solutions which are close by some criterion to the initial one) determined and then the local optimal solution is sought in this neighborhood. The procedure is repeated for a new initial solution determined with the use of the *mutation* operator. The process goes on until finding the global optimum, which is not a trivial process. One confines oneself to determining a permissible packing and estimating the quality of solution. As for the local optimum, a neighborhood with the local lower bound approximate to the local optimum was constructed. This is the subject matter also of the present paper which seeks solution in the neighborhood by local variation of the solution of the one-dimensional cutting problem. The global lower bounds (LB) are used to estimate the

[1] This work was supported by the Foundation of the President of Russian Federation, project no. MK.145.2003.01.

solution obtained of which the first uses the solution of the linear programming problem and the second is an improved modification of the first.

The problems presented in the paper find application in technical and organizational problems. First, these are the problems of part packing with their subsequent cutting from a material. Using the proposed bounds, the part packing effectiveness can be estimated and then approached to the optimal one. Second, these are the problems of transport packing logistic that is developing actively in the modern world. Rational packing of loads in the transportation facilities enables one to reduce the fleet of vehicles and maintain the necessary convenience of loading and unloading.

## 2. CODING AND DECODING OF THE ORTHOGONAL PACKING

Packing design faces determination and execution of the following procedures: select the method of packing, write its code, and construct the packing using a decoding algorithm. Various methods of coding were discussed in detail in [7]. Here we just characterize briefly some of them.

Scheme of direct coding. We introduce an orthogonal coordinate system with axes coinciding with the horizontal and vertical sides of the strip. Position of each rectangle $i = \overline{1, m}$ is defined by the coordinates $(x_i; y_i)$ of its lower left vertex. The *direct scheme* of coding is represented by a sequence of vectors. Search of the packing subset $\{x, y\}$ without mutual overlapping of the rectangles also is an $\mathcal{NP}$-trying problem. At the same time, if this subset is established, then design of the packing by its direct code is unique and presents no difficulties. Therefore, other coding schemes endowed with decoders—algorithms to pass from the initial to the direct scheme—are used.

Coding by block structure. Let there be a rectangular packing. We draw vertical lines through the right sides of the rectangles. They decompose the packing into blocks (corteges) called the vertical block structures (VBS) [8].

*Example 1.* The Fig. 1 shows three different packings of the same rectangles $P_1$–$P_6$. In the cases of Figs. 1a and 1b, the packings are coded by the block structure VBS $= ((1, 3, 6)\chi_1;\ (1, 6)\chi_2;\ (4, 2, 6)\chi_3;\ (2)\chi_4;\ (5)\chi_5)$ and differ in the $y$ coordinates of the packed rectangles. The length of packings is $L = \chi_1 + \chi_2 + \chi_3 + \chi_4 + \chi_5 = l_1 + l_2 + l_5$, where $\chi_k$ is the length of the $k$th block and $l_i$ the length of the $i$th rectangle. In the case of Fig. 1c, the packing is coded by the block structure VBS$' = ((1, 3, 6)\chi_1;\ (1, 6)\chi_2;\ (2, 4, 6)\chi_3;\ (2, 5)\chi_4')$, its length is $L' = l_1 + l_2 < L$.

Permutational coding. The permutation $\pi = (1(\pi), 2(\pi), \ldots, i(\pi), \ldots, m(\pi))$, where $i(\pi)$ is the number of the rectangle in the position $i$ of the sequence $\pi$, indicates the order of packing the rectangles in the rightward-upward direction. After preliminary elimination of the repeated elements, the permutation $\pi$ can be obtained by concatenation of the corteges of the VBS block structure.

Substitution. There exist various algorithms (decoders) reproducing the code in packing [8]. The quality of packing depends on the choice of decoder and its algorithmic realization. The

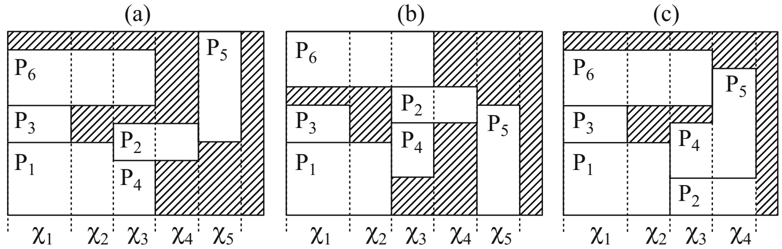


**Fig. 1.** Coding of packings: (a) and (b) are various packings coded by one block structure; (a)–(c) are packings coded by different block structures.

*substitution* (Sub) is used as the basis in the block algorithms. It reproduces the packing block structure from the given permutation $\pi$. In the course of local search, Sub makes use of the *next fit* (NF) strategy [1] which lies in that a next, still non packed element from the list $\pi$ is taken for packing in an empty area. If it does not accommodate itself in this area, then the next free area is used to pack it. The Sub algorithm with the NF strategy (SubNF) generates a block and fixes the free areas in the blocks that follow it.

## 3. Λ-NEIGHBORHOOD OF THE PERMISSIBLE RECTANGULAR PACKING

Let us consider an individual problem of orthogonal packing $(\Upsilon,\ L)$, where $\Upsilon$ is a finite set of packings $p$ with the initial data $\langle W; m; w; l\rangle$, and the packing $p^* \in \Upsilon$ minimizing the objective function $L = \max_i(x_i + l_i)$. The packing $p^*$ of length $L^* = \min L$ is the global minimum. Now, we determine for each permissible packing $p \in \Upsilon$ the neighborhood function $\Lambda$ defining for $p$ the set of neighbor solutions that are close in a sense to the given solution and the $\Lambda$-neighborhood of the permissible solution $p \in \Upsilon$. The following one-dimensional cutting stock problem (1DCSP) is associated with the individual problem $(\Upsilon, L)$. Let us assume that $Z = W$ is the length of a one-dimensional bin;[2] $\lambda = w$ is the vector of piece lengths and $\beta = l$ is the vector of the required number of pieces (assortment vector). Needed is to minimize the number $N$ of the bins used which is equal to the length $\Lambda$ of the cut material. Let us consider the 1DCS problem with additional constraints on the permissible solution:

(1°) only different pieces may be put into the bin (*heterogeneity*);

(2°) if not all pieces of type $i$ fit into the bin, then the remaining are put in the following bins (*continuity*).

The 1DCS problem with constraints (1°) and (2°) is called the *rectangular oriented cutting stock problem* (ROCSP), and its permissible solution, the rectangular oriented orthogonal linear cutting (ROLC) [8]. The number $\Lambda$ of the bins used is the objective function in the ROCS problem. If the NF algorithm is used to solve the problem of ROCS of the set of pieces $\{w, l\}$ with the order of packing defined by the list $\pi$, the rectangular oriented linear cutting is denoted by $\mathrm{ROLC}_{\mathrm{NF}}((\lambda, \beta), \pi)$. Now we present A.S. Filippova's notion of the passive permutations in the list $\pi$ [9].

Passive permutations. The $\Lambda$-neighborhood of interest is defined by the permutations of the elements in $\pi$ leading to the rectangular oriented one-dimensional cuttings differing only in the order of pieces in the same bins. We refer to such cuttings and their corresponding lists as *equivalent*. Let there be one-dimensional linear cutting $\mathrm{ROLC}_{\mathrm{NF}}((\lambda, \beta), \pi)$.

**Definition 1.** The permutation of the elements in $\pi$ is called the *passive* permutation if application of NF to the resulting list $\pi'$ leads to the cutting $\mathrm{ROLC}_{\mathrm{NF}}((\lambda, \beta), \pi')$ which is equivalent to the initial $\mathrm{ROLC}_{\mathrm{NF}}((\lambda, \beta, ), \pi)$.

In both cuttings the number of occupied bins is the same. The following statement is evident.

**Lemma 1** (criterion for passive permutation). *Let in the block $S_j$ of the block structure ROLC two elements $i_1(\pi)$ and $i_2(\pi)$ satisfying the following conditions be chosen:*

• *both elements $i_1$ and $i_2$ begin at the block $j$ and on the list $\pi$ the element $i_1$ precedes the element $i_2$;*

• *the length $\lambda_{i_2}$ of the element $i_2$ exceeds the residual capacity $r_{j-1}$ of the neighbor block $S_{j-1}$. Then, the pair $(i_1(\pi), i_2(\pi))$ is passive in the permutation $\pi$.*

[2] By the bin is meant any packed or cut rectangular object such as strip, rod, basket, or bin. In what follows we stick to this term.

Let there be an orthogonal packing $p \in \Upsilon$, its corresponding list $\pi(p)$, and rectangular oriented linear cutting $\mathrm{ROLC_{NF}}((\lambda, \beta), \pi(p))$ of length $\Lambda$. Then, construction of a solution that neighbors $p$ lies in executing the following steps:

($\Lambda$1)—*Construction of the sequence $\pi'$ equivalent to $\pi$*: random selection of a block with at least two initial elements and random selection of a pair of elements meeting the conditions of Lemma 1. The permutation of the passive elements in the marked block is a new sequence $\pi'$ equivalent to $\pi$.

($\Lambda$2)—*Construction of solution neighboring $p$*: under the input data $\langle W; m; w; l\rangle$, we apply SubNF to the list $\pi'$, obtain $p' = \mathrm{RP_{NF}}((w, l), \pi')$ of length $L'$ and its block structure VBS$'$ for calculation of the coordinates of rectangles.

*Example 2.* The RP shown in Fig. 1a is taken as the initial packing $p$ under the source data of Example 1. The block structure $\mathrm{VBS} = ((1, 3, 6)\chi_1;\ (1, 6)\chi_2;\ (4, 2, 6)\chi_3;\ (2)\chi_4;\ (5)\chi_5)$ and $L = \chi_1 + \chi_2 + \chi_3 + \chi_4 + \chi_5 = l_1 + l_2 + l_5$ correspond to it. After elimination in the corteges of the repeated elements we obtain the sequence of reduced corteges $\mathrm{rBS} = ((1, 3, 6); (4, 2); (5))$, and after concatenation, the list $\pi = (1, 3, 6, 4, 2, 5)$. We generate the input information for the ROCS problem: $\langle W; m; \lambda = w; \beta = l\rangle$; apply NF to the list $\pi(p)$ and obtain the rectangular oriented linear cutting $\mathrm{ROLC_{NF}}((\lambda, \beta), \pi(p)) = ((1, 3, 6)\chi_1;\ (1, 6)\chi_2;\ (4, 2, 6)\chi_3;\ (2, 5)\chi_4)$ of length $\Lambda = \chi_1 + \chi_2 + \chi_3 + \chi_4 = l_1 + l_2$. We take the cortege no. 3 and permute elements 4 and 2 proceeding from it; the permutation of these elements in $\pi$ provides the list $\pi' = (1, 3, 6, 2, 4, 5)$ equivalent to $\pi$. We restore the input information for 2DSPP and apply the NF algorithm with the list $\pi'$. For the new packing $p'$, we have $\mathrm{BS}' = ((1, 3, 6)\chi_1;\ (1, 6)\chi_2;\ (2, 4, 6)\chi_3;\ (2, 5)\chi_4)$ and $L' = l_1 + l_2 = \Lambda < L$.

Repetition of the process of constructing the neighbor solutions provides the neighborhood $\Lambda(p)$ of the solution of the orthogonal packing $p$ which is called the $\Lambda$-neighborhood.

## 4. LOWER BOUND OF THE Λ-NEIGHBORHOOD

NF-active packings. The problem of one-dimensional bin packing (1DBP) where all pieces differ, that is, $\beta_i = 1$, $i = \overline{1, m}$, is a special case of the problem 1DCS of one-dimensional cutting. The permissible solution of 1DBPP is representable as the list $S = (S_1; S_2; \ldots; S_j, \ldots; S_N)$, where $S_j$ is a cortege (sequence of the numbers of the pieces put in the $j$th bin). The block $j$ with the use intensity $\chi_j = 1$ corresponds to the cortege $S_j$. The order $\pi$ of element packing can be obtained by concatenation of the reduced corteges $S_j$, $j = \overline{1, N}$.

For the source data of the 1DBP problem, we make use of its solution by the NF algorithm assuming at that the permutation $\pi$ establishing the order of packing the pieces is known. The resulting packing of the set of pieces $\{\lambda\}$ is denoted by $\mathrm{P_{NF}}(\lambda, \pi)$. We note that if one makes use of another strategy of packing, then the resulting packing usually will be different.

**Definition 2.** The permissible one-dimensional packing $p$ of the set $\{\lambda\}$ of $m$ pieces is called the NF-active packing for fixed permutation $\pi$ if $\mathrm{P_{NF}}(\lambda, \pi)$ coincides with $p$.

This definition was proposed by V.V. Zalyubovskii in the paper [10] devoted to the one-dimensional packings. Let $p$ be the permissible packing of a set of pieces $\{\lambda\}$ and $|p|$ be the number of occupied bins. Then, the following lemma is true.

**Lemma 2** (Zalyubovskii). *For any permissible packing $p$ of the set of pieces $\{\lambda\}$, there exists an NF-active packing $p'$ such that*

$$|p'| \leq |p|. \qquad (1)$$

At the same time, the 1DCS problem is just a generalization of the problem of packing. Namely, $\lambda_i$ is repeated $\beta_i$ times if $\beta_i > 1$, that is, 1DCSP is the 1DBP problem of packing with repeated pieces. Then, a modification of NF with permutation $\pi$ leads to an NF-active linear cutting. As for the ROCS problem, the Conditions (1°) and (2°) are readily realized within the framework of the aforementioned modifications.

Modification of the substitution algorithm with the NF strategy for solution of the ROCS problem. The modified next fit (MNF) strategy for solution of the ROCS problem consists of the following procedures:

($MNF1$) *Generation of the first cortege*: apply NF to pack the first bin with different elements (Condition (1°)) in the order of the list $\pi$ until it is possible. We assume that $\chi_1 = \min\limits_{i \in I_1} \beta_i$, where $I_1$ is the set of the numbers of the elements packed in the first bin; $\beta_i' = \beta_i - \chi_1$, $i \in I_1$; the element $i \in I_1$ such that $\beta_i = \chi_1$ is removed from the list $\pi$.

($MNF2$) *Packing of the current $j$th bin* is carried out with allowance for Condition (2°) until $\pi \neq \emptyset$: we apply ($MNF1$) to the modified information about the input data $\beta$ and the list $\pi$, obtain the cortege $S_j$ and intensity of its use $\chi_j$; correct $\beta$ and $\pi$; and assume that $j = j + 1$.

($MNF3$) *End*. We obtain a ROLC coded by the block structure VBS $= (S_1\chi_1; S_2\chi_2; \ldots; S_j\chi_j; \ldots)$.

Therefore, MNF reproduces the ROLC for the set of elements $(\lambda, \beta)$ by packing them in the bins in the order defined by the list $\pi$ with observation of Conditions (1°) and (2°).

Now it is possible to go to the initial problem 2DSP.

Local lower bound of the $\Lambda$-neighborhood of the rectangular packing. Let the permissible packing $p \in \Upsilon$ of the set of pieces $\{w, l\}$ be known and the permutation $\pi(p)$ and $\mathrm{ROLC_{NF}}((\lambda, \beta), \pi(p))$ of length $\Lambda$ correspond to it.

The following lemma is a corollary to Lemma 2.

**Lemma 3.** *For any rectangular packing $p$ of the set of pieces $\{w, l\}$ and NF-active rectangular oriented linear cutting* $\mathrm{ROLC_{NF}}((\lambda, \beta), \pi(p))$, $\lambda = w$, $\beta = l$, *the inequality*

$$|\mathrm{ROLC_{NF}}((\lambda, \beta), \pi(p))| \leq |p(w, l)| \tag{2}$$

*is valid, no matter what algorithm was not used to reproduce $p$.*

**Definition 3.** By the lower local bound of the $\Lambda$-neighborhood of the packing $p \in \Upsilon$ is meant the number $h$ such that

$$h \leq |p| \tag{3}$$

for any packing $p$ from this neighborhood.

According to Definition 3, the main statement follows from Lemma 3.

**Theorem 1.** *The solution* $\Lambda = \mathrm{ROLC_{NF}}((\lambda, \beta), \pi(p))$ *of the problem* ROCS *with the initial information* $\langle W; m; \lambda = w; \beta = l\rangle$, $\pi(p)$ *is the local lower bound of the $\Lambda$-neighborhood of the packing $p$ of the set of pieces $\{w,\ l\}$.*

The solution $\Lambda$ of the ROCS problem with the initial data corresponding to 2DSPP will be called the $\Lambda$-*bound*.

**Definition 4.** If for some packing $p^*$ from the $\Lambda$-neighborhood of $p$ we get $|p^*| = \Lambda$, then $\Lambda$ is the local optimal minimum.

## 5. SEARCH OF THE BEST LOCAL OPTIMAL SOLUTION USING Λ-NEIGHBORHOODS

The aforementioned underlies a hybrid algorithm of searching an optimal solution with the use of the Λ-neighborhood (HSS-Λ). More than one strategy of using the Λ-neighborhood is possible. We dwell on one of them. The HSS-Λ strategy relies on the possibility of calculating the Λ-bound prior to constructing the current solution. The ROCS problem is solved for that purpose. Its solution is the initial local Λ-bound. For the list $\pi$ corresponding to ROLC, the initial packing $p = \mathrm{RP}$ is determined using SubNF. The neighborhood is constructed using the passive permutations in $\pi$, and a packing with the record value of length $L$ is found in it. Then, a new neighborhood with smaller $\Lambda$ is generated using the one-point evolutionary algorithm (1+1)-EA [3], and a packing with the record $L' < L$ is sought in it. We present one iteration of the HSS-Λ algorithm.

The method of sequential value correction (SVC) [11] is used to solve the ROCS problem. The SVC procedure is repeated until a solution superior to the preceding one is found or the runtime limit is exhausted. If $\Lambda$ improved, then the Λ-neighborhood is generated and the best packing is sought in it (line 15).

Algorithm 1. Search of the best solution in Λ-neighborhood

1 **Algorithm** HSS-Λ
2 **input:**
3 $W$, $m$, $w = (w_1, \ldots, w_m)$, $l = (l_1, \ldots, l_m)$ *//input information for 2DSPP*
4 $L$, $\lambda$, $\beta$ *//input information for ROCSP*
5 **variables:**
6 $d = (d_1, \ldots, d_m)$ *//pseudoestimates*
7 rolc *//block structure of solution of problem ROCS*
8 **INITIALIZATION OF VARIABLES:**
9 $L \leftarrow W$, $\lambda \leftarrow w$, $\beta \leftarrow l$
10 rolc $\leftarrow$ (); *//rolc initialization by empty list*
11 iter $\leftarrow$ 1;
12 **MAIN PART:**
13 **while not** TimeLimit
14 rolc $\leftarrow$ SVC(SubKP, iter, $\langle L; m; \lambda; \beta\rangle$, $d$); *// solution of ROCS problem, improvement of* $\Lambda$
15 SearchNeighbourhood(rolc); *//construction and search in neighborhood*
16 iter $\leftarrow$ iter + 1;
17 **END.**

Complexity of an iteration of the HSS-Λ algorithm is calculated as follows. Lines 10 and 11 are executed in a constant time $O(1)$. As will be shown below, solution of the ROCS problem (line 14) has complexity $O(W \times m^3)$. Search in the neighborhood (line 15) is done in $O(m \times \log m)$ steps. Cycle 13–16 is repeated until exhausting the time limit which is not dependent functionally on the problem size. Consequently, in the worst case the time complexity of one iteration of the algorithm is equal to $O(W \times m^3 + m \times \log m) = O(W \times m^3)$.

Solution of ROCS problem. The SVC method for the problem of one-dimensional cutting was proposed by E.A. Mukhacheva [11], and its latest modification is described in [12]. It represents an evolutionary algorithm with the mutation operator realizing a permutation in $\pi$ with preference to the maximal element norms. The method is based on the idea of filling the bins in compliance with the solution of the binary knapsack problem with simultaneous maximization of the total pseudoestimate of the elements which is similar in a sense to the objectively conditioned L.V. Kantorovich estimates. Introduction of the pseudoestimates enables one to realize the idea of accounting for the resource consumption for each element in each bin on the basis of the previous solutions.

The computational complexity of the SVC algorithm depends on the heuristics H used in the method to generate the cuttings. The decoder with the substitution knapsack (SubKP) strategy [12] was used as H, its time complexity is $O(m^3)$. Complexities of the rest of SVC procedures do not exceed $O(m^3)$. Therefore, complexity of the entire SVC procedure is $O(W \times m^3)$.

Search in neighborhood. The Λ-neighborhood is constructed and the best solution is sought in it as follows. The rectangular oriented linear cutting rolc is used as the input information. One packing from the Λ-neighborhood is constructed at each iteration, and its length is calculated. Reduction of rolc is done in lines 11 and 12, and the neighborhood cardinality $\prod_k N_k!$ is calculated, where $N_k$ is the number of elements in the $k$th column of the reduced rolc. If the neighborhood cardinality does not exceed the given number max_iter (line 13), then the full search in the neighborhood (line 21) can be carried out. Otherwise, $N$ elements of the Λ-neighborhood are randomly enumerated. For that, each column of the reduced rrolc is shuffled (line 17), and concatenation of the new rrolc is performed, that is, a new list $\pi$ (line 18) is constructed, after which the packing is reproduced by the SubNF algorithm (line 19) using the resulting list. Therefore, the ss block structure includes a packing from the neighborhood. The best packing restored by the NF algorithm using the list $\pi$ from Λ-neighborhood is the result of the SearchNeighbourhood procedure.

ALGORITHM 2. Construction of Λ-neighborhood with search of the best solution

```
1  procedure SearchNeighbourhood(rolc)
2  input:
3    W, m, w = (w_1, ..., w_m), l = (l_1, ..., l_m) //input information for 2DSPP
4  parameters:
5    max_iter                          //max. neighborhood cardinality for complete enumeration
6    N                                 //max. number of elements in neighborhood
7  variables:                          //looked through at random enumeration
8    rrolc                             //reduced rolc
9    power                             //cardinality of rolc neighborhood
10 BEGIN
11   rrolc ← Reduce_rolc(rolc);        //reduction of rolc
12   power ← Calc_power(rrolc);
13   if power > max_iter               //if neighborhood cardinality exceeds max_iter,
14     iter ← 1;                       //it is enumerated randomly
15     while iter < N
16       for each slice i in rrolc
17         rrolc_i ← Random_shuffle(rrolc_i)
18       π ← π(rrolc);
19       ss ← SubNF(⟨L; m; w; l⟩, π);
20       iter ← iter + 1;
21   else CompleteSearch(rrolc, power); //otherwise, entire neighborhood may be enumerated
22 END.
```

Complexity of calculating the reduced rolc is $O(m)$. In the worst case, the number of columns of the reduced rolc is equal to the number of elements. Therefore, the complexity of calculations of the neighborhood cardinality is $O(m)$. For the same reason, the calculational complexity of lines 16–18 is $O(m)$. Complexity of the SubNF algorithm is $O(m \times \log m)$ [7]. Therefore, the time complexity of lines 14–20 is $O(m \times \log m)$. The full search in the CompleteSearch neighborhood (line 21) has an exponential complexity. Yet, since full search is carried out only if the cardinality of the enumerated set does not exceed the given number $N$, the algorithm's computational complexity in the worst case is $O(m \times \log m)$.

## 6. GLOBAL LOWER BOUNDS

The well-known lower bound 1 algorithm (LB1) based on linear programming [5] is used to estimate effectiveness of the proposed HSS algorithm. Here we propose an improved lower bound LB2 which can be introduced using another method to code packings proposed by V.M. Kartak.

Horizontal coding of packing. Let packing RP be given. The method to code packing by block structures (Section 2) requires vertical cuts through the right sides of the rectangles. The sequence of the resulting blocks is called the *vertical block structure* (VBS). In a similar way we make the horizontal cuts along the lower sides of the rectangles. The sequence of the horizontal blocks is called the *horizontal block structure* (HBS).

The rectangular oriented problem of linear cutting (ROCS) defined in Section 2 is called below the *vertical cutting stock problem* (VCSP). Its permissible solution is called the *vertical linear cutting* (VLC). Similarly, we define the *horizontal cutting stock problem* (HCSP) for the horizontal decomposition of packing as the corresponding HBS problem 1DCS with the constraints (1°) and (2°). The source data for it are $\langle L; m; \lambda; \beta \rangle$, where $\lambda_i = l_i$, $\beta_i = w_i$, $L$ is the length of the occupied RP strip. The permissible solution is represented by the *horizontal linear cutting* (HLC).

Relation between VCS and HCS. Let for some RP $\langle W; m; w; l \rangle$ the length of the occupied strip part $L$ be known.

**Theorem 2** (criterion for existence of a permissible packing). *The following conditions must be satisfied for existence of a permissible orthogonal packing of length not greater than the number $L$ of the bins used in* VLC*:*

(1) *there exist permissible solutions* VLC *and* HLC *with the numbers of used bins not exceeding $L$ and $W$, respectively,*

(2) *if there exists block $S_j$ of the* VLC *block structure including two elements $p$ and $q$, then no block where both elements occur exists in the block structure* HLC.

Theorem 2 is proved in the Appendix.

To calculate the RP lower bound, we verify the conditions of Theorem 2. For that, we assign to each block $S_j$ of the VLC block structure an $m$-dimensional binary vector $a^j = (a_1^j, \ldots, a_m^j)^{\mathrm{T}}$, where $a_i^j = 1$ if there is the element $i$ in the block $j$; otherwise, $a_i^j = 0$.

Bortfeld [5] proved that the following linear programming (LP) problem is the global lower bound for RP

$$z = \sum_{j=1}^{N} x_j \longrightarrow \min, \tag{4}$$

on the set of vectors $x$ satisfying the conditions

$$Ax = \beta, \quad x \in R_+^N, \tag{5}$$

where $A$ is a matrix consisting of all feasible binary vectors $a^j$ and $N$ is the number of such vectors. This problem solves VCL disregarding (2°). We denote by $Z(\mathrm{VCS})$ the optimal solution of problem (4) and (5).

Let us define a similar LP problem for HCS

$$z = \sum_{j=1}^{M} y_j \longrightarrow \min, \tag{6}$$

on the set of vectors $y$ meeting the conditions

$$By = \lambda, \quad y \in R_+^M, \tag{7}$$

where $B$ is a matrix consisting of all feasible binary vectors corresponding to the HBS blocks and $M$ is the number of such vectors.

The optimal value of the solution of problem (6) and (7) is denoted by $Z$(HCS). If packing is done in a strip of length not greater than $L$, then $Z$(HCS) is the minimal width of this strip.

We denote by $L_d$ the current lower bound for RP. Obviously, if $Z(\text{HCS}) > W$ is true for the HCS $\langle L_d; m; \beta; \lambda \rangle$, then the initial set of rectangles cannot by packed in a strip of length $L_d$ and width at most $W$, that is, the first condition of Theorem 2 in this case is not satisfied and, consequently, $L_d$ must be increased. Since the rectangles have integer dimensions and the length of the occupied part of strip can have only an integer value, $L_d$ may be increased by one. On this basis, the following algorithm to calculate the initial lower bound can be suggested.

Algorithm 3. Calculation of the initial value of the lower bound

```
Algorithm FLB
input:
  W, w = (w_1, ..., w_m), l = (l_1, ..., l_m)   //input information for 2DSPP
variables:
  L_d                                           // lower bound
MAIN PART:
  L_d ← ⌈Z(VCS)⌉                                // initial value for the lower bound
  HCS ← (L_d, m, w, l)                          // determination of data for HCS
  if Z(HCS) > W
    L_d = L_d + 1; goto 8                       // increase of the lower bound
END.
```

To satisfy the second condition of Theorem 2, we introduce into problem (4) additional constraints forbidding the use of the blocks containing a fixed pair of elements $(p, q)$ in the solutions of both problems (see also [13]):

$$z = \sum_{j=1}^{N} x_j \longrightarrow \min, \tag{8}$$

provided that the following conditions are satisfied:

$$\begin{cases} Ax = \beta, & x \in R_+^N \\ x_i = 0, & i \in \{j \mid a_p^j = 1 \text{ and } a_q^j = 1\}. \end{cases}$$

This LP problem considers only the vectors $a^j$ corresponding to the blocks having no elements numbered $p$ and $q$. We denote by $Z_{(p,q)}$(VCS) the optimal solution of problem (8).

Similarly, we determine $Z_{(p,q)}$(HCS) for the following problem:

$$z = \sum_{j=1}^{M} x_j \longrightarrow \min, \tag{9}$$

provided that the following conditions are satisfied:

$$\begin{cases} By = \lambda, & y \in R_+^M \\ y_i = 0, & i \in \{j \mid b_p^j = 1 \text{ i } b_q^j = 1\}. \end{cases}$$

**Lemma 4.** *Let for some value $L$ there be a pair $(p, q)$ such that $Z_{(p,q)}(\text{VCS}) > L$ and $Z_{(p,q)}(\text{HCS}) > W$. Then, there is no packing* RP *whose occupied part of the strip is shorter than $L$.*

Lemma 4 is proved in the Appendix.

Consequently, if the condition of Lemma 4 is satisfied for some value of the lower bound $L_d$, then there exists no packing that fits in a strip of length at most $L_d$. Therefore, one must increase the lower bound $L_d$ by one and again verify satisfaction of the conditions of Lemma 4. The algorithm to calculate the improved lower bound is as follows.

Algorithm 4. Calculation of the lower bound

**Algorithm** Lower Bound 2
**input:**
  $W$, $w = (w_1, \ldots, w_m)$, $l = (l_1, \ldots, l_m)$ *//input information for 2DSPP*
**variables:**
  $L_d$ *// lower bound*
**MAIN PART:**
  $L_d \leftarrow$ FLB() *// initial value of the lower bound*
  for each (p,q) *// enumeration of all pairs of elements in VROLC*
    if $Z_{(p,q)}(VCS) > L_d$ and $Z_{(p,q)}(HCS) > W$ then $L_d = L_d + 1$
**END.**

## 7. NUMERICAL EXPERIMENT

The experiment was run on the Intel Celeron M 1.40 GHz, 768 Mb. The software was compiled in the MS Visual Studio 2005 environment. The experiment was carried out using the Berkey–Wang problems [14] (classes of examples 1–6 in the table) and Martello–Vigo problems [15] (classes of examples 7–10 in the table). The library contains 500 problems divided into 10 classes of 50 problems each, and each class in conventionally divided into 5 subclasses according to the number of rectangles equal to 20, 40, 60, 80, and 100; the strip varies from 10 to 300.

Also, the HSS-$\Lambda$ algorithm and two genetic algorithms SPGAL and GGSub were compared for efficiency. The SPGAL algorithm proposed by Bortfeld [5] is a genetic algorithm based on the level technology. The genetic algorithm with greedy decoder GGSub developed by A.S. Filippova and R.R. Shirgazin relies on the block technology [8].

The mean values of the objective function $L$ for two 60-second runs of the problem are compiled as the results of the HSS-$\Lambda$ algorithm. The results of the SPGAL algorithm are the mean value of $L$ for ten 95 second runs each. The GGSub algorithm was run twice for 60 seconds.

The results of calculations are compiled in the table consisting of thirteen columns representing four groups of indices. The first column shows the classes and subclasses of problems. The first group consists of two columns showing the mean values of the global bounds LB1 and LB2 for each subclass, class, and experiment as a whole. The remaining three groups are intended for the mean results of problem solutions, respectively, by the algorithms HSS-$\Lambda$, SPGAL, and GGSub. All three groups have columns with the mean values of the length $L$ of the occupied part of the strip and its deviations gap1 from the global lower bound LB1. For the HSS-$\Lambda$ algorithm, the following indices are introduced and located in the fourth, fifth, sixth, and ninth columns: the mean value $\Lambda$ of the local $\Lambda$-bound; the mean deviations gap1($\Lambda$) and gap2($\Lambda$) of $\Lambda$, respectively from LB1 and LB2, and the mean deviation gap2 of $L$ from the global lower bound LB2. The deviation gap($L$) for each problem was calculated as $100 \times (L - LB)/L$.

The results of the numerical experiment are as follows:

- the new global bound LB2 in 35 of 50 subclasses proved to be the best bounds LB1,
- the local bound $\Lambda$ in seven subclasses coincided with the global bound LB2,
- the best mean values for $L$ were given by the HSS-$\Lambda$ algorithm which was the best in 5 of 10 subclasses and in 24 of 50 subclasses,

Results for the examples of Berkey–Wang (classes 1–5) and Martello–Vigo (classes 6–10)

| | | | HSS-Λ | | | | | | SPGAL | | GGSub | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Example | LB1 | LB2 | $\Lambda$ | gap1($\Lambda$) | gap2($\Lambda$) | $L$ | gap1($L$) | gap2($L$) | $L$ | gap1($L$) | $L$ | gap1($L$) |
| C01 | | | | | | | | | | | | |
| 1–10 | 60.30 | 60.80 | 61.38 | 1.76 | 0.31 | 61.60 | 2.06 | 1.39 | 61.60 | 2.15 | **61.3** | **1.53** |
| 11-20 | 121.60 | 121.70 | 121.93 | 0.27 | 0.02 | 121.90 | 0.29 | 0.17 | 122 | 0.36 | **121.8** | **0.2** |
| 21–30 | 187.40 | 188.5 | 188.47 | 0.57 | 0.11 | **188.70** | **0.68** | 0.11 | 189 | 0.83 | **188.7** | **0.68** |
| 31–40 | 262.20 | 262.6 | 262.65 | 0.17 | 0.02 | **262.70** | **0.19** | 0.19 | 262.8 | 0.23 | 262.8 | 0.23 |
| 41–50 | 304.40 | 304.8 | 304.9 | 0.16 | 0.03 | **305** | **0.19** | 0.06 | 305 | 0.19 | 305.3 | 0.3 |
| Mean value | 187.18 | 187.68 | 187.86 | 0.59 | 0.1 | 187.98 | 0.68 | 0.36 | 188.08 | 0.75 | **187.98** | **0.59** |
| C02 | | | | | | | | | | | | |
| 1–10 | 19.70 | 19.70 | 19.81 | 0.56 | 0 | **19.80** | **0.56** | 0.56 | 20.5 | 3.91 | 19.90 | 0.99 |
| 11–20 | 39.10 | 39.10 | 39.1 | 0 | 0 | **39.10** | **0** | 0 | **39.1** | **0** | 39.40 | 0.65 |
| 21–30 | 60.10 | 60.10 | 60.1 | 0 | 0 | **60.1** | **0** | 0 | **60.1** | **0** | 60.90 | 1.28 |
| 31–40 | 83.20 | 83.20 | 83.2 | 0 | 0 | **83.2** | **0** | 0 | 83.3 | 0.11 | 83.80 | 0.72 |
| 41–50 | 100.50 | 100.5 | 100.5 | 0 | 0 | **100.5** | **0** | 0 | 100.7 | 0.2 | 101.40 | 0.86 |
| Mean value | 60.52 | 60.52 | 60.54 | 0.11 | 0 | **60.54** | **0.11** | 0.11 | 60.74 | 0.84 | 61.08 | 0.90 |
| C03 | | | | | | | | | | | | |
| 1–10 | 157.40 | 161 | 164.81 | 4.5 | 0.08 | 164.70 | 4.57 | 2.32 | 166.7 | 5.72 | **164.5** | **4.46** |
| 11–20 | 328.80 | 331.6 | 334.35 | 1.66 | 0.18 | **334.60** | **1.83** | 1.04 | 335.4 | 2.08 | **334.50** | **1.83** |
| 21–30 | 500 | 503.4 | 507.1 | 1.4 | 0.38 | 508.50 | 1.77 | 1.04 | 509.8 | 2.02 | **507.20** | **1.52** |
| 31–40 | 701.70 | 707.2 | 709.85 | 1.15 | 0.28 | 711.80 | 1.43 | 0.67 | 712.5 | 1.53 | **710.80** | **1.29** |
| 41–50 | 832.70 | 835 | 840.09 | 0.88 | 0.3 | 841.90 | 1.18 | 0.90 | 842.6 | 1.23 | **840.90** | **1.04** |
| Mean value | 504.12 | 507.64 | 511.24 | 1.92 | 0.24 | 512.30 | 2.16 | 1.19 | 513.4 | 2.52 | **511.58** | **2.03** |
| C04 | | | | | | | | | | | | |
| 1-10 | 61.40 | 61.9 | 63.67 | 3.56 | 0.45 | **64** | **3.99** | 3.25 | 66.3 | 7.53 | 64.40 | 4.65 |
| 11–20 | 123.90 | 123.9 | 125.77 | 1.49 | 0.73 | **126.70** | **2.21** | 2.21 | 127.1 | 2.58 | 128.20 | 3.35 |
| 21–30 | 192 | 193 | 195.7 | 1.38 | 0.48 | 196.70 | 1.85 | 1.85 | **196.6** | **1.83** | 199.2 | 3.01 |
| 31–40 | 267.20 | 267.2 | 270.42 | 1.19 | 0.58 | **272** | **1.76** | 1.76 | 272.2 | 1.84 | 275.20 | 2.91 |
| 41–50 | 322 | 322 | 325.39 | 1.04 | 0.48 | **327** | **1.52** | 1.52 | 327.3 | 1.62 | 330.50 | 2.57 |
| Mean value | 193.50 | 193.6 | 196.19 | 1.73 | 0.54 | **197.28** | **2.27** | 2.12 | 197.9 | 3.08 | 199.46 | 3.30 |
| C05 | | | | | | | | | | | | |
| 1–10 | 512.20 | 530.2 | 535.72 | 4.39 | 0.84 | 540 | 5.19 | 2.17 | 536.6 | 4.42 | **535** | **4.14** |
| 11–20 | 1053.80 | 1069.1 | 1075.64 | 2.03 | 0.32 | 1077.70 | 2.35 | 0.88 | 1081.4 | 2.6 | **1074.30** | **1.97** |
| 21–30 | 1614 | 1635.4 | 1650.48 | 2.21 | 0.04 | 1649 | 2.25 | 0.89 | 1650.8 | 2.34 | **1647.5** | **2.12** |
| 31–40 | 2268.40 | 2283.8 | 2289 | 0.9 | 0.23 | 2293.80 | 1.13 | 0.44 | 2299.5 | 1.38 | **2291.2** | **1** |
| 41–50 | 2617.40 | 2634.9 | 2654.46 | 1.4 | 0.16 | 2656.40 | 1.55 | 0.87 | 2666.9 | 1.93 | **2652.7** | **1.40** |
| Mean value | 1613.16 | 1630.68 | 1641.06 | 2.19 | 0.32 | 1643.38 | 2.49 | 1.05 | 1647.04 | 2.53 | **1640.14** | **2.13** |
| C06 | | | | | | | | | | | | |
| 1–10 | 159.90 | 162.8 | 167.84 | 4.73 | 0.56 | **170.80** | **6.46** | 4.61 | 179.1 | 10.85 | 172.9 | 7.65 |
| 11–20 | 323.50 | 323.5 | 332.14 | 2.6 | 0.51 | **333.80** | **3.10** | 3.10 | 337 | 4.1 | 340 | 4.92 |
| 21–30 | 505.10 | 505.1 | 516.81 | 2.27 | 0.57 | **519.80** | **2.82** | 2.82 | 519.8 | 2.84 | 525.30 | 3.86 |
| 31–40 | 699.70 | 699.7 | 713.95 | 2 | 0.51 | **717.60** | **2.49** | 2.49 | 719.4 | 2.74 | 726.80 | 3.74 |
| 41–50 | 843.80 | 843.8 | 860.06 | 1.89 | 0.45 | **863.90** | **2.33** | 2.33 | 868.1 | 2.81 | 873.70 | 3.43 |
| Mean value | 506.40 | 506.98 | 518.16 | 2.7 | 0.52 | **521.18** | **3.44** | 3.07 | 524.68 | 4.67 | 527.74 | 4.72 |
| C07 | | | | | | | | | | | | |
| 1–10 | 490.40 | 500.6 | 500.77 | 2.07 | 0.38 | 501.90 | 2.45 | 0.30 | 502.7 | 2.62 | **501.90** | **2.44** |
| 11–20 | 1049.70 | 1054.7 | 1060.62 | 1.03 | 0.1 | 1059.90 | 1.03 | 0.44 | 1059.4 | 0.99 | **1059** | **0.94** |
| 21–30 | 1515.90 | 1525 | 1529.67 | 0.9 | 0.03 | 1530.20 | 0.93 | 0.35 | 1529.7 | 0.9 | **1529.60** | **0.89** |
| 31–40 | 2206.10 | 2220.8 | 2221.65 | 0.7 | 0.06 | 2222.80 | 0.76 | 0.09 | 2222.9 | 0.76 | **2222.10** | **0.72** |
| 41–50 | 2627 | 2642.7 | 2643.66 | 0.63 | 0.07 | 2645.80 | 0.7 | 0.11 | 2648.8 | 0.81 | **2644.40** | **0.65** |
| Mean value | 1577.82 | 1588.76 | 1591.27 | 1.07 | 0.13 | 1592.12 | 1.17 | 0.26 | 1592.7 | 1.22 | **1591.40** | **1.13** |
| C08 | | | | | | | | | | | | |
| 1–10 | 434.60 | 439.1 | 461.1 | 5.75 | 0.03 | 463.50 | 6.14 | 5.26 | 465.9 | 6.6 | **460.20** | **5.52** |
| 11–20 | 922 | 924.7 | 954.35 | 3.39 | 0.06 | **955.20** | **3.44** | 3.17 | 956.2 | 3.58 | 956.60 | 3.61 |
| 21–30 | 1360.90 | 1360.9 | 1400.84 | 2.85 | 0.32 | 1405.30 | 3.16 | 3.16 | **1398.9** | **2.73** | 1408 | 3.35 |
| 31–40 | 1909.30 | 1911.4 | 1960.87 | 2.63 | 0.08 | **1962.40** | **2.7** | 2.61 | 1967.3 | 2.93 | 1972 | 3.17 |
| 41–50 | 2362.80 | 2363 | 2418.92 | 2.32 | 0.13 | 2422.50 | 2.45 | 2.44 | **2422.3** | **2.44** | 2435.10 | 2.95 |
| Mean value | 1397.92 | 1399.82 | 1439.22 | 3.39 | 0.12 | **1441.78** | **3.58** | 3.33 | 1442.12 | 3.66 | 1446.38 | 3.72 |
| C09 | | | | | | | | | | | | |
| 1–10 | 1106.80 | 1106.8 | 1106.8 | 0 | 0 | **1106.80** | **0** | 0 | **1106.8** | **0** | **1106.80** | **0** |
| 11–20 | 2189.20 | 2190 | 2190.81 | 0.07 | 0.01 | 2190.80 | 0.08 | 0.04 | 2191.2 | 0.11 | **2190.60** | **0.07** |
| 21–30 | 3410.40 | 3410.4 | 3410.4 | 0 | 0 | **3410.40** | **0** | 0 | 3417.5 | 0.19 | **3410.40** | **0** |
| 31–40 | 4578.60 | 4584.4 | 4587.69 | 0.2 | 0 | **4588.10** | **0.2** | 0.08 | **4588.10** | **0.2** | **4588.10** | **0.2** |
| 41–50 | 5430.50 | 5431.2 | 5435.06 | 0.08 | 0 | **5434.90** | **0.08** | 0.06 | **5434.90** | **0.08** | **5434.90** | **0.08** |
| Mean value | 3343.10 | 3344.56 | 3346.15 | 0.07 | 0 | **3346.2** | **0.07** | 0.04 | 3347.7 | 0.12 | **3346.16** | **0.07** |
| C10 | | | | | | | | | | | | |
| 1–10 | 337.80 | 347.1 | 352.06 | 4.05 | 0.06 | 351.60 | 4.24 | 1.60 | 354.2 | 5.11 | **351.10** | **4.10** |
| 11–20 | 642.80 | 654.6 | 666.39 | 3.54 | 0.12 | 670.10 | 4.14 | 2.50 | **664.7** | **3.29** | 665.90 | 3.51 |
| 21–30 | 911.10 | 919.5 | 936.13 | 2.67 | 0.2 | 938.90 | 2.69 | 2.19 | **932.6** | **2.28** | 938.6 | 2.97 |
| 31–40 | 1177.60 | 1186 | 1210.94 | 2.75 | 0.35 | 1215.20 | 3.09 | 2.47 | **1207.4** | **2.43** | 1211.70 | 2.80 |
| 41–50 | 1476.50 | 1480.7 | 1514.02 | 2.48 | 0.26 | 1517.90 | 2.73 | 2.49 | **1507.8** | **2.09** | 1518.50 | 2.79 |
| Mean value | 909.16 | 917.58 | 935.91 | 3.1 | 0.2 | 938.80 | 3.44 | 2.25 | **933.34** | **3.04** | 937.16 | 3.24 |
| C01–C10 | | | 1042.76 | 1.69 | | **1044.16** | **1.94** | 1.38 | 1044.77 | 2.24 | 1044.91 | 2.18 |

• as compared with HSS-$\Lambda$ and GGSub, the SPGAL algorithm was the worst in all classes, except for the last one,

• the global lower bound LB2 enables us to draw the following conclusions about the solutions by the HSS-$\Lambda$ algorithm: the optimal solutions were obtained in six of the 50 subclasses, the mean deviation from the optimum does not exceed 0.1% in 10 subclasses. and in 25 subclasses the mean deviation from the optimum does not exceed 1%.

## 8. CONCLUSIONS

The one-point method of local search of the optimum was proposed and realized. The record permissible solutions are determined in the $\Lambda$-neighborhood and improved at passing to another $\Lambda$-neighborhood. At that, the NF-strategy is realized for various lists $\pi$ differing in the permutations of the passive elements at the current iteration of the evolutionary algorithm. Numerical results of comparing the proposed algorithm with other effective algorithms are presented. In the majority of classes, HSS-$\Lambda$ proved to be the best. At that, the paths to improving it are seen in the application of the methods with prohibitions at seeking in the neighborhood, use of other metaheuristics, genetic algorithm, and the method of annealing. Additionally, the crossbreeding of the precise method and heuristics seems to be promising. An improved global lower bound LB2 was developed. The mean deviation from it of the solution is 1.38%, that is, smaller than the deviation 1.94% from LB1 obtained by the same algorithm.

## *APPENDIX*

**Proof of Theorem 2.** Let us assume that condition (1) is not true, which means that there exists no RP such that it fits the strip of width $W$ and occupies a length not longer than $L$. If such packing existed, then there would exist its corresponding solutions of the problems VCL and HSL with the number of bins used not exceeding $W$ and $L$.

Let RP be given such that there are in it two element $p$ and $q$ such that they are present in the block $S_j$ of the vertical block structure VLC, and there exists block $S_i$ in the horizontal block structure HLC where these elements are present as well. This means that the vertical block $S_j$ and the horizontal block $S_i$ intersect the rectangles numbered $p$ and $q$, which is possible only in the case of intersecting rectangles $p$ and $q$, that is, RP is impermissible.

**Proof of Lemma 4.** Let there exist RP occupying a strip of length equals at most to $L$. The inequality $Z_{(p,q)}(\text{VCS}) > L$ implies that there exists no solution of VCS without a vertical block with the elements $p$ and $q$ and the number of bins used smaller than $L$. Consequently, existence of such block is necessary to satisfy condition (1).

Similarly, $Z_{(p,q)}(\text{HCS}) > W$ implies that there should exist a horizontal block with elements $p$ and $q$. In this case, condition (2) of Theorem 2 is not satisfied. Consequently, there is no such RP.

## REFERENCES

1. Lodi, A., Martello, S., and Vigo, D., Recent Advance on Two-dimensional Bin Packing Problems, *Discret. Appl. Math.*, 2002, vol. 123, no. 1–3, pp. 379–396.

2. Mukhacheva, E.A. and Valeeva, A.F., Method of Dynamic Enumeration in the Problem of Two-dimensional Packing, *Inform. Tekhnol.*, 2000, no. 5, pp. 30–37.

3. Borisovskii, P.A. and Eremeev, A.V., Comparison of Certain Evolutionary Algorithms, *Autom. Remote Control*, 2004, no. 3, pp. 357–362.

4. Loris, F., An Application of Simulated Annealing to the Cutting Stock Problem, *Eur. J. Oper. Res.*, 1999, vol. 114, pp. 814–837.

5. Bortfeld, A., A Genetic Algorithm for the Two-dimensional Strip Packing Problem with Rectangular Pieces, *Eur. J. Oper. Res.*, 2006, vol. 172(3), pp. 814–837.

6. Mukhacheva, E.A., Mukhacheva, A.S., and Chiglintsev, A.V., Genetic Block Algorithm in Two-dimensional Packing, *Inform. Tekhnol.*, 1999, no. 11, pp. 13–18.

7. Mukhacheva, E.A., Nazarov, D.A., and Filippova, A.S., Design of Rectangular Packings Using Decoders of Block Structure, *Autom. Remote Control*, 2006, no. 6, pp. 977–988.

8. Mukhacheva, E.A. and Mukhacheva, A.S., Problem of Orthogonal Packing: Methods of Local Search of Optimum on the Basis of Block Structures, *Autom. Remote Control*, 2004, no. 2, pp. 248–257.

9. Zhitnikov, V.P. and Filippova, A.S., Problem of Orthogonal Packing into Half-infinite Strip: Search of Solution in the Neighborhood of the Local Lower Bound, *Inform. Tekhnol.*, 2007, no. 5, pp. 55–62.

10. Zalyubovskii, V.V., On Permutation-based Representation of the Permissible Solutions of the One-dimensional Problem of Bin Packing, in *Proc. XIII Baikal. Workshop*, Irkutsk, 2005, vol. 1, pp. 461–468.

11. Mukhacheva, E.A. and Zalgaller, V.A., Linear Programming Cutting Problems, *Int. J. Software Engin. Knowledge Engin.*, 1993, vol. 3(4), pp. 463–476.

12. Belov, G., Scheithauer, G., and Mukhacheva, E.A., One-dimensional Heuristics Adapted for Two-dimensional Rectangular Strip Packing, *J. Oper. Res. Soc., Advance Online Publ.*, 2007.

13. Kartak, V.M., Sufficient Conditions for the Integer Round-Up Property to Be Violated for the Linear Cutting Stock Problem, *Autom. Remote Control*, 2004, no. 3, pp. 407–412.

14. Berkey, J.O. and Wang, P.Y., Two-dimensional Finite Bin-packing Algorithms, *J. Oper. Res. Soc.*, 1987, vol. 38(5), pp. 423–429.

15. Martello, S. and Vigo, D., Exact Solution of the Finite Two-dimensional Bin Packing Problem, *Manag. Sci.*, 1998, vol. 44, pp. 388–399.

*This paper was recommended for publication by P.Yu. Chebotarev, a member of the Editorial Board*