Compaction algorithm for orthogonal packing problems

Vladislav A Chekanin¹ and Alexander V Chekanin¹

¹Department of Theoretical Mechanics and Strength of Materials, Moscow State University of Technology «STANKIN», 3a Vadkovsky lane, Moscow, 127055, Russian Federation

Abstract. The article is devoted to an algorithm for increasing the density of placement schemes in orthogonal packing problems. The proposed heuristic packing compaction algorithm is based on the principle of local replacement of placed objects located as far from the origin of a container as well as located near the local free spaces inside it. The algorithm uses a list of six rules which select objects from a container for deleting and subsequent reallocation of them into freed spaces of the container. The article contains results of computational experiments carried out on the standard strip packing problems.

1. Introduction

Solution of many different practical optimization problems, including resources saving problem, cutting of materials, problems in logistics and manufacturing, scheduling and planning comes down to the orthogonal packing problem [1–3]. This problem is a particular case from a wide class of resource allocation problems. All these problems are classical problems of the mathematical theory of operations research. The solution of these problems is to find the most rational ways of allocating resources (materials) of one type (called objects) among resources of another type (called containers). A detailed classification of resource allocation problems was proposed by G. Wascher, H. Haubner and H. Schumann in 2007 [4], in preparation of which 445 scientific articles were considered. The wide area of practical applications of resource allocation problems in industry and the economy makes the problem of developing effective algorithms and methods for solving these problems urgent [5], as evidenced by the presence of a large number of scientific publications [1, 4].

The orthogonal packing problem is the NP-complete problem in the strong sense [6]. To obtain exact solutions of such problems is necessary to use the resource-intensive optimization algorithms, which are ineffective on practice due to heavy spending of time resources. Therefore, to solve NP-complete packing problems approximate heuristic optimization algorithms are usually used. They provide obtaining of suboptimal solutions for an acceptable time [7–10].

In this article is proposed a packing compaction algorithm based on the idea of reallocation of placed into a container objects. Since the problem of developing of an iterative algorithm intended for compacting of the placement schemes was first introduced here, there is no practical possibility of comparing its effectiveness with other similar algorithms. The described algorithm is designed to improve the quality of suboptimal solutions (i.e. to increase the density of the placement schemes), which were previously found using various optimization heuristic and metaheuristic algorithms, the development of which is the goal of most researchers in the field of the cutting and packing.

2. Statement of the orthogonal packing problem

Consider the statement of $D$-dimensional orthogonal packing problem of $n$ objects into one container with dimensions $\{W^1, W^2, ..., W^D\}$. All objects and the container are presented in the form of $D$-
dimensional parallelepipeds with dimensions $[w_i^1, w_i^2, \ldots, w_i^D], i \in \{1, \ldots, n\}$. We denote the position of an object $i$ located inside the container by $(x_i^1, x_i^2, \ldots, x_i^D)$. The goal is to find the most density placement of all the objects into the container under the following conditions of correct placement:

- all edges of the objects are parallel to edges of the container;
- all packed objects do not overlap with each other:
  \[
  (x_i^d \geq x_j^d + w_j^d) \lor (x_j^d \geq x_i^d + w_i^d) \quad \forall d \in \{1, \ldots, D\}, \quad i, k \in \{1, \ldots, n\}, \quad i \neq k ;
  \]
- all packed objects are within the bounds of the container:
  \[
  (x_i^d \geq 0) \land (x_i^d + w_i^d \leq W^d) \quad \forall d \in \{1, \ldots, D\}, \quad \forall i \in \{1, \ldots, n\}.
  \]

When objects are placed into the container any rotation of them is not allowed.

A solution of any dimensional packing problem is represented by a so-called placement string (also known as a chromosome) which contains a sequence of objects selected for placing. To obtain a packing is necessary to decode the placement string for this is solved the problem of selecting the objects to be packed and free spaces of the container. The statement of $D$-dimensional orthogonal packing problem includes specifying of a load direction of the container as the priority selection list $\{L\} = \{L^1; L^2; \ldots; L^D\}$ of its coordinate axes, where $L^d \in [0;D] \forall d \in \{1, \ldots, D\}$, $L^d \neq L^j \forall i \neq j$. This list defines the order of selecting of free spaces of the container for placing the objects into them.

The quality of the placement is estimated by the criteria of length minimization of filled with the objects container’s part which is measured along the coordinate axis $l = L^D$, i.e. the minimized fitness function is calculated as $S = \max(x_i^l + w_i^l), \quad i = 1 \ldots n$ and it defines the position of the most remote object in the container.

To describe placement schemes of orthogonal objects in containers was chosen the previously developed packing representation model – the model of potential containers. The efficiency of this model for orthogonal packing problems is shown in paper [11]. This model describes the current state of the container with a set of so-named potential containers in form of orthogonal objects with the maximal dimensions which potentially can be placed in the container. A potential container $k$ is described by a vector $[p_k^1, p_k^2, \ldots, p_k^D]$ of its dimensions as well by a vector $[x_k^1, x_k^2, \ldots, x_k^D]$ which defines the position of the origin of the coordinate system of this potential container relative the origin of the coordinate system of the containing it container. The fast access to potential containers is provided by usage of the effective multilevel linked data structure described in papers [12, 13].

3. Compacting packing problem

In the basis of the proposed packing compaction algorithm lays usage of the rules for selecting the placed objects for deleting of them with subsequent more rational filling of the freed spaces in the container. The implementation of an algorithm for deleting objects designed especially for the model of potential container model is described in detail in [14]. The rules of selecting objects for reallocation of them are based on the methods given below.

Method AREA selects from the container all objects contacting with the most remote edges of an orthogonal object with the dimensions $V^d = W^d \forall d \neq l$, $V^l = \max(x_i^l + w_i^l), \quad i = 1 \ldots n$ which encloses the placement (figure 1).

Method LAST selects from the container all objects contacting with potential containers the edges of which along the coordinate axis $l$ match to the border of the container, i.e. the potential containers for which $x_l^k + p_l^k = W^l$ (figure 2).

Method MAX selects from the container all objects contacting with a potential container that has the maximal volume (area) and located inside the placement, i.e. the potential container for which $x_l^k + p_l^k < W^l$ (on figure 3 this potential container is represented by a rectangle without color filling).

Method DOUBLE selects from the container all objects contacting with two potential containers which have maximal volumes (areas) and located inside the placement (figure 4).
The methods of the first group (AREA and LAST) are intended to select objects located as far from the origin of the container. The methods of the second group (MAX and DOUBLE) are intended to select the objects located close to the largest free spaces in the container. Based on these methods, six object selection rules for local reallocation of placed objects were implemented (table 1).

| No. | Rule         | Method 1 | Method 2 | Description                                                                 |
|-----|--------------|----------|----------|-----------------------------------------------------------------------------|
| 1   | MAX_AREA     | AREA     | MAX      | Cooperative application of two methods.                                    |
| 2   | MAX_LAST     | LAST     | MAX      | Cooperative application of two methods.                                    |
| 3   | DOUBLE_AREA  | AREA     | DOUBLE   | Cooperative application of two methods.                                    |
| 4   | DOUBLE_LAST  | LAST     | DOUBLE   | Cooperative application of two methods.                                    |
| 5   | MAXMAX_AREA  | AREA     | MAX      | Application of the method AREA as well as application of the method MAX to the placement scheme obtained after deleting of all objects previously found by the method MAX. |
| 6   | MAXMAX_LAST  | LAST     | MAX      | Application of the method LAST as well as application of the method MAX to the placement scheme obtained after deleting of all objects previously found by the method MAX. |

The proposed packing compaction algorithm is a heuristic algorithm which sequentially applies each of the described rules from the special ordered list (this order is given in the section of computational experiments of the article). This algorithm works with a population of 10 chromosomes (the placement strings), composed as follows:

- chromosome 1 contains sequence of objects in the order as they recorded in the problem;
- chromosome 2 contains sequence of objects ordered by descending of their volumes (areas);
- chromosome 3 contains sequence of objects ordered by ascending of their volumes (areas);
- chromosome 4 contains sequence of objects ordered by descending of their dimensions measured along the coordinate axis \( l \);
- chromosome 5 contains sequence of objects ordered by ascending of their dimensions measured along the coordinate axis \( l \);
- chromosomes 6–10 contain pseudorandom sequences of objects.
If any chromosome provides obtaining of a placement scheme with the better density (i.e. with the smaller value of the fitness function), then the best found solution is saved, after which is performed an attempt to compact this placement again and for that are applied all the rules from the ordered list starting with the first rule. This process is repeated iteratively while the algorithm provides a better density of the placement or until the stop criteria will be performed, which is the achievement of a predefined number of iterations or the use of all rules without any improvement of the density.

4. Computational experiments
The effectiveness of the compaction algorithm is investigated on the standard two-dimensional strip packing problems (rectangular cutting problems) from the paper of O. Berkey and P. Wang [15] (classes of instances C1–C6) and from the paper of S. Martello and D. Vigo [16] (classes of instances C7–C10). Were solved 500 problem instances (10 classes of 50 instances each). Every class contains from 20 to 100 rectangular objects. The maximal number of iterations is chosen equal to 20.

At the first stage of the computational experiments, the effectiveness of applying of each the object selection rule was separately analyzed. In this case was used the simplified packing compaction algorithm which uses only one object selection rule and the transition to the next iteration proceeds only if more dense placement of deleted objects was obtained at the current iteration. The algorithm was applied for placements schemes previously obtained after decoding of placement strings which contain objects in order as they recorded in the considered test problems.

The quality of compacting is estimated by evaluation $\mu = (S_0 - S_1)/S_0 \times 100\%$, where $S_0$ is a value of the fitness function before compacting, $S_1$ – a value of the fitness function after compacting.

For each rule of selecting objects, a series of 10 computational experiments was implemented. The averaged results are summarized in table 2.

**Table 2.** The quality of compacting for different object selection rules.

| Rule           | Class of test instances | Average quality, % |
|----------------|-------------------------|---------------------|
|                | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| MAX_AREA       | 0.43 | 2.39 | 0.10 | 2.09 | 0.37 | 1.12 | 0.38 | 1.05 | 0.00 | 0.64 | 0.86 |
| MAX_LAST       | 0.47 | 3.96 | 0.67 | 3.43 | 0.45 | 2.68 | 0.38 | 1.41 | 0.00 | 0.96 | 1.44 |
| DOUBLE_AREA    | 0.56 | 3.50 | 0.27 | 2.24 | 0.63 | 1.88 | 0.60 | 1.48 | 0.03 | 1.11 | 1.23 |
| DOUBLE_LAST    | 0.65 | 4.39 | 0.74 | 3.80 | 0.79 | 3.31 | 0.66 | 1.77 | 0.09 | 1.30 | 1.75 |
| MAXMAX_AREA    | 1.21 | 3.37 | 0.41 | 2.96 | 0.76 | 2.37 | 0.65 | 1.91 | 0.15 | 1.38 | 1.52 |
| MAXMAX_LAST    | 0.94 | 4.88 | 1.06 | 4.29 | 0.87 | 4.01 | 0.67 | 2.22 | 0.15 | 1.62 | 2.07 |

The results of the computational experiments have showed that the MAXMAX_LAST rule ensures the greatest efficiency of compacting. This rule provides obtaining of most dense placement schemes in comparison with other rules for 8 test classes, however, when using the MAXMAX_LAST rule, the packing compaction algorithm spends the maximum time. The smallest effect of the applying of the algorithm is achieved after using the MAX_AREA rule, which provides the minimal quality of compacting, while it has the highest time efficiency.

On the basis of the obtained results, was chosen the following order of usage of the object selection rules is set for the packing compaction algorithm: MAXMAX_LAST, DOUBLE_LAST, MAXMAX_AREA, MAX_LAST, DOUBLE_AREA, MAX_AREA (the rules are listed in decreasing order of their quality of compacting). On the second stage of the computational experiments were obtained the averaged results of testing the packing compaction algorithm (table 3).
Table 3. Results of testing the compaction algorithm.

| Class of test instances | Average value of the fitness function before compacting | Average value of the fitness function after compacting | Average time, s | Quality of compaction, % |
|-------------------------|----------------------------------|----------------------------------|----------------|------------------------|
| 1                       | 203.98                           | 201.76                           | 1.06           | 1.53                   |
| 2                       | 69.34                            | 65.56                            | 1.29           | 6.72                   |
| 3                       | 577.88                           | 573.20                           | 1.15           | 1.55                   |
| 4                       | 229.82                           | 218.98                           | 1.75           | 6.65                   |
| 5                       | 1817.86                          | 1804.08                          | 1.19           | 1.34                   |
| 6                       | 608.02                           | 579.32                           | 1.97           | 5.68                   |
| 7                       | 1685.78                          | 1676.24                          | 1.07           | 0.78                   |
| 8                       | 1644.70                          | 1603.48                          | 1.48           | 3.53                   |
| 9                       | 3398.78                          | 3394.94                          | 1.04           | 0.26                   |
| 10                      | 1052.94                          | 1034.88                          | 1.36           | 2.17                   |
| Average                 | 1128.91                          | 1115.24                          | 1.34           | 3.02                   |

5. Conclusion
The compaction algorithm for the orthogonal packing problems has been proposed. The work of this algorithm is based on the idea of iterative reallocation of placed objects in a container. The effectiveness of the algorithm for different rules of selection objects has been investigated on the standard two-dimensional packing strip problems. The developed packing compaction algorithm is implemented in a general form, which makes it possible to use it not only for two-dimensional problems as well as for three-dimensional orthogonal packing problems.

6. References
[1] Bortfeldt A and Wascher G 2013 Eur. J. Oper. Res. 229(1) 1–20
[2] Lodi A, Martello S and Monaci M 2002 Eur. J. Oper. Res. 141(2) 241–52
[3] Martello S, Pisinger D and Vigo D 2000 Oper. Res. 48(2) 256–67
[4] Wascher G, Haubner H and Schumann H 2007. Eur. J. Oper. Res. 183(3) 1109–30
[5] Crainic T G, Perboli G and Tadei R 2008 Informs J. Comput. 20(3) 368–84
[6] Johnson D S 2012 Documenta Mathematica Extra Volume ISMP 359–76
[7] Kierkosz I and Luczak M 2014 Cent. Eur. J. Oper. Res. 22(4) 729–53
[8] Leung S C H, Zhang D F, Zhou C L and Wu T 2012 Comput. Oper. Res. 39(1) 64–73
[9] Chekanin V A and Chekanin A V 2014 Appl. Mech. Mater. 598 377–81
[10] Chekanin A V and Chekanin V A 2013 Comp. Math. Math. Phys. 53(10) 1457–65
[11] Chekanin V A and Chekanin A V 2015 Adv. Mech. Eng. 22 33–8
[12] Chekanin A V and Chekanin V A 2014 Adv. Mater. Res. 962–965 2868–71
[13] Chekanin V A and Chekanin A V 2014 Adv. Mater. Res. 945-949 3143–46
[14] Chekanin V A and Chekanin A V 2016 ARPN J. Eng. Appl. Sci. 11(13) 8436–46
[15] Berkey O and Wang P 1987 Journal of the Operational Research Society 38(5) 423–29
[16] Martello S and Vigo D 1998 Manage. Sci. 44 388–99

Acknowledgements
This work was carried out with the financial support of the Ministry of Education and Science of Russian Federation in the framework of the state task in the field of scientific activity of MSTU "STANKIN" (No. 1.7706.2017/8.9).