

Application of Algorithms for Placement of Orthogonal Polyhedrons for Solving the Problems of Packing Objects of Complex Geometric Shape

Vladislav Chekanin^{1,*} and Alexander Chekanin¹

¹Moscow State Technological University "STANKIN", RU-127055, Moscow, Russia

Abstract. The article is devoted to the development and research of algorithms for placing objects of complex geometric shapes. To solve the placement problem is proposed an approach which consists in transforming the shape of all objects and further application of the developed algorithm for placing orthogonal polyhedrons of arbitrary dimension to the resulting transformed objects. In the process of transforming the shape of the objects being placed, they are initially voxelized, after which the developed decomposition algorithm is applied to the resulting voxelized objects, which provides the formation of orthogonal polyhedrons consisting of the largest possible orthogonal objects. The proposed model of potential containers is used to describe the free space of containers as a set of orthogonal areas. The developed algorithm for the placement of orthogonal polyhedrons provides a fast solution to NP-hard problems of placing objects of complex geometric shapes without resorting to the use of time-consuming nonlinear programming methods. Examples of the practical application of the developed algorithms for modeling the dense layout of parts of complex geometric shapes on the platform of a 3D printer are given.

1 Introduction

The problem of packing objects of complex geometric shape has a large number of practical applications in various areas. Solving the problems of cutting industrial materials, rational usage of free spaces (for example, ships or aircraft), geometric surface coverage with specified shapes, modelling the microstructure of composite materials as well as a number of other optimization problems is reduced to packing problem [1-6].

All packing problems are NP-hard optimization problems for which there are no algorithms of polynomial complexity to solve them, which makes it urgent to develop algorithms that provide suboptimal solutions in an acceptable time. Modern methods for solving problems of packing objects of complex geometric shape are based on the use of the hodograph vector function of dense placement, which requires the subsequent application of nonlinear programming methods characterized by high computational complexity [7-12].

To reduce the complexity of the problem, we propose an approach that consists in packing of voxelized objects [13-15] by applying the developed algorithm of placement of orthogonal polyhedrons.

We will consider the problem of placement orthogonal polyhedrons in the general D -dimensional case. A container is specified in the form of D -dimensional parallelepiped with the dimensions

$\{W^1; W^2; \dots; W^D\}$ (the superscript in formulas means the number of the coordinate axis) as well as specified a set of n orthogonal polyhedrons $O_i, i \in \{1, \dots, n\}$, each of which consists of m_i orthogonal objects in the form of D -dimensional parallelepipeds $o_{i,k}, k \in \{1, \dots, m_i\}$ with the dimensions $\{w_{i,k}^1; w_{i,k}^2; \dots; w_{i,k}^D\}$, the position of which relative to each other is specified using vectors $\{z_{i,k}^1; z_{i,k}^2; \dots; z_{i,k}^D\}$ containing the coordinates of orthogonal objects in the local coordinate system associated with each orthogonal polyhedron O_i [16]. In the particular case, when all orthogonal polyhedrons consist of only one orthogonal object $m_i = 1 \forall i \in \{1, \dots, n\}$, the considered problem will be reduced to the classic D -dimensional orthogonal packing problem [1, 2].

2 Description of free container spaces

To describe a packing is used the developed model of potential containers [17-19]. Under a potential container (PC) placed in a container at some its point is understood an imaginary orthogonal object with the largest possible dimensions that can be placed at this point without overlapping with any packed into the container object and edges of the container. Each potential container h is described with a vector $\{p_h^1; p_h^2; \dots; p_h^D\}$ containing its

* Corresponding author: vladchekanin@rambler.ru

dimensions and a vector $\{x_h^1; x_h^2; \dots; x_h^D\}$ containing coordinates of its point which is nearest to the origin of the container containing it.

All existing free orthogonal spaces located in a container are described by a set of potential containers. When a new orthogonal object is put into a container it is necessary to verify the correctness of the placement. The model of potential containers guarantees the correct placement of an orthogonal object if it overlaps no borders of the potential container in which it is located. In this case when an object is put at some point of a container instead of checking on the intersection with all placed into the container objects is required to check only one condition of placement of this object entirely within the potential container, located at this point. This ensures a higher speed of formation the orthogonal packing.

When an orthogonal object i with the dimensions $\{w_i^1; w_i^2; \dots; w_i^D\}$ is placed at a point $\{x_i^1; x_i^2; \dots; x_i^D\}$ it divides the potential container h into a set of smaller potential containers from two sets:

1) set of D potential containers with the dimensions $\{p_h^1; p_h^2; \dots; p_h^{d-1}; x_i^d - x_h^d; p_h^{d+1}; \dots; p_h^D\}$ located in the origin $\{x_h^1; x_h^2; \dots; x_h^d; \dots; x_h^D\}$ of the potential container h produced under the following conditions: $x_i^d > x_h^d$ and $x_i^d < x_h^d + p_h^d \quad \forall d \in \{1, \dots, D\}$;

2) set of D potential containers with the dimensions $\{p_h^1; p_h^2; \dots; p_h^{d-1}; x_h^d + p_h^d - x_i^d - w_i^d; p_h^{d+1}; \dots; p_h^D\}$ located at D points with coordinates $\{x_h^1; x_h^2; \dots; x_h^{d-1}; x_i^d + w_i^d; x_h^{d+1}; \dots; x_h^D\}$ produced under the following conditions: $x_i^d + w_i^d > x_h^d$ and $x_i^d + w_i^d < x_h^d + p_h^d \quad \forall d \in \{1, \dots, D\}$.

Figure 1 presents all new potential containers which are formed in a three-dimensional potential container after placing an orthogonal object inside it.

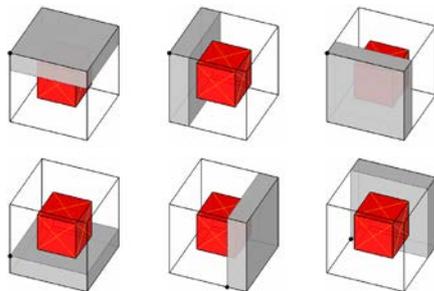


Fig. 1. New potential containers in a three-dimensional case.

To update a set of potential containers after placing an orthogonal polyhedron O_i at the point $\{X_i^1, X_i^2, \dots, X_i^D\}$ of a D -dimensional container, the free spaces of which are described by a set of potential containers Ω_0 , the following algorithm is performed.

Step 1. Create a set $\Omega'_0 \subset \Omega_0$ of potential containers $h: \exists d \in \{1, \dots, D\}: x_h^d \leq X_i^d + S_i^d$, where $\{S_i^1, S_i^2, \dots, S_i^D\}$ denotes the overall dimensions of a D -dimensional

parallelepiped bounding the orthogonal polyhedron O_i : $S_i^d = \max(z_{i,k}^d + w_{i,k}^d), \quad \forall d \in \{1, \dots, D\}, k \in \{1, \dots, m_i\}$.

Step 2. Place an orthogonal polyhedron O_i in the specified position $\{X_i^1, X_i^2, \dots, X_i^D\}$ of a new identical empty container, as a result of which a set of potential containers Ω will be formed in it. Placement of the orthogonal polyhedron is performed by sequentially placing of all its orthogonal objects $o_{i,k}, k \in \{1, \dots, m_i\}$.

Step 3. Apply the intersection operation (Figure 2) to sets of potential containers Ω'_0 and Ω to get a set of potential containers $\Omega''_0 = \Omega'_0 \cap \Omega$ that describes all free spaces of the original container in the area of the placed orthogonal polyhedron O_i . During the intersection operation, each set of potential containers is considered as an orthogonal polyhedron, consisting of orthogonal objects whose parameters coincide with the parameters of the corresponding potential containers.

Step 4. Replace in the set Ω_0 all potential containers that are also in the set Ω'_0 with potential containers from the set Ω''_0 .

This algorithm is shown in Figure 3.

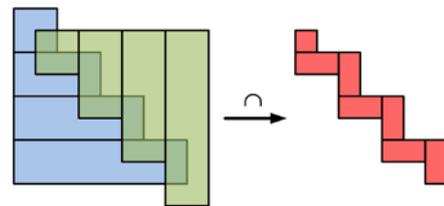


Fig. 2. Intersection operation.

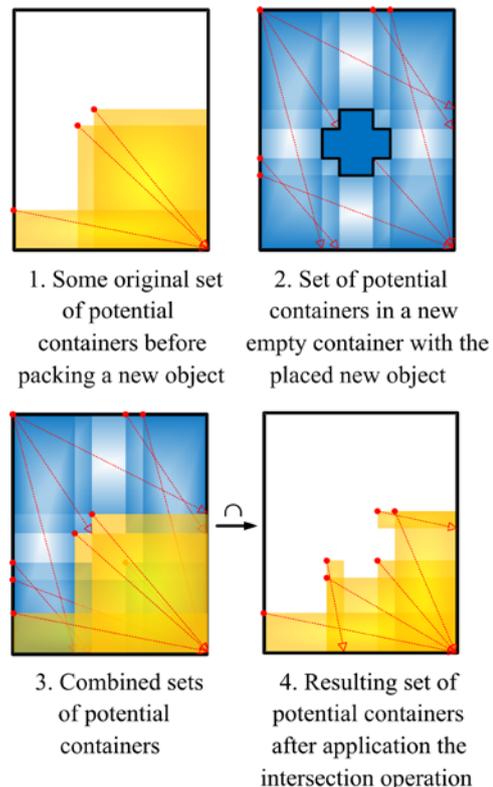


Fig. 3. Updating of potential containers.

3 Preparing objects for placement

As a result of converting objects to a voxel representation, orthogonal polyhedrons consisting of a large number of cubes are obtained. To reduce the number of orthogonal objects forming an orthogonal polyhedron, a decomposition algorithm has been developed. The algorithm provides the decomposition of a D -dimensional orthogonal polyhedron V into a set of large orthogonal objects includes steps 1–6.

Step 1. Create an empty D -dimensional orthogonal container 1 with the overall dimensions $\{W_1^1, W_1^2, \dots, W_1^D\}$ matching the dimensions of a D -dimensional parallelepiped bounding the original orthogonal polyhedron V ($W_1^d = S^d, \forall d \in \{1, \dots, D\}$), where S^d is the length of the packing measured along the axis d .

Step 2. Place the orthogonal polyhedron V into container 1.

As a result in the container 1, a set of potential containers Ω_1 with the overall dimensions $\{w_{k_1}^1, w_{k_1}^2, \dots, w_{k_1}^D\}$, $k_1 \in \Omega_1$ located at points $\{p_{k_1}^1, p_{k_1}^2, \dots, p_{k_1}^D\}$ will be formed. The set of potential containers Ω_1 describes the space of container 1, which does not belong to the placed orthogonal polyhedron V .

Step 3. Create an empty D -dimensional orthogonal container 2 with the overall dimensions that match the overall dimensions of the container 1 ($W_2^d = W_1^d, \forall d \in \{1, \dots, D\}$).

Step 4. Place in the container 2 a set of D -dimensional orthogonal objects with parameters matching the parameters of potential containers from the container 1 (when placing objects, their mutual overlap is allowing): $x_i^d = p_{k_1}^d$ and $w_i^d = w_{k_1}^d, \forall d \in \{1, \dots, D\}$, $i = 1 \dots |\Omega_1|$. As a result in the container 2, a set of potential containers Ω_2 with the overall dimensions $\{w_{k_2}^1, w_{k_2}^2, \dots, w_{k_2}^D\}$, $k_2 \in \Omega_2$, located at points $\{p_{k_2}^1, p_{k_2}^2, \dots, p_{k_2}^D\}$, will be formed. The set Ω_2 describes the space of container 2, which belongs to the orthogonal polyhedron V .

Step 5. Create a D -dimensional orthogonal polyhedron V' , consisting of orthogonal objects with parameters matching the parameters of potential containers from the container 2: $z_i^d = p_{k_2}^d$ and $w_i^d = w_{k_2}^d, \forall d \in \{1, \dots, D\}$, $i = 1 \dots |\Omega_2|$, so the orthogonal polyhedron V' will contain the set of all orthogonal objects into which it can be decomposed.

Step 6. Apply the addition operation to all orthogonal objects that are part of the orthogonal polyhedron V' . As a result, an orthogonal polyhedron O will be obtained, consisting of the largest orthogonal objects that do not overlap each other.

The operation of addition orthogonal objects (Figure 4) includes the following steps.

1. Place all objects from the set A into the set A^+ . Create sets $B = \emptyset, C = \emptyset$.
2. Sort the set A^+ in descending order of volumes (areas) of objects.
3. Include the first object $o_1 \in A^+$ in the set C .
4. For each current object $o_k \in A^+ (k = 2 \dots |A^+|)$ check for overlapping with the object o_1 . If the objects do not overlap each other, then place the current object o_k in the set B ; otherwise, perform the cutting off procedure for all objects, and then place the obtained objects in the set B .
5. Clear the set A^+ . If the set $B \neq \emptyset$, then move all the objects from the set B to the set A^+ , and then go to step 2.

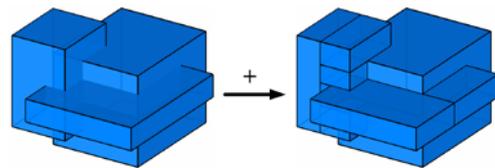


Fig. 4. Addition operation.

Unlike decomposition algorithms developed by other authors [20–22], the proposed algorithm is described and programmatically implemented invariantly with respect to the dimension of the problem to be solved.

The decomposition significantly increases the speed of placing objects. Figures 5 as an example shows original voxelized ball as well as the obtained decomposed orthogonal polyhedron. In this figure also shown time which was spent to place the orthogonal polyhedrons (here and later in this paper, the results of computational experiments performed on a personal computer with CPU Intel Core i5-8400, 2,8 GHz and 8 GB RAM are presented). An example of the decomposition of a complex geometric shape is shown in the Figure 6.

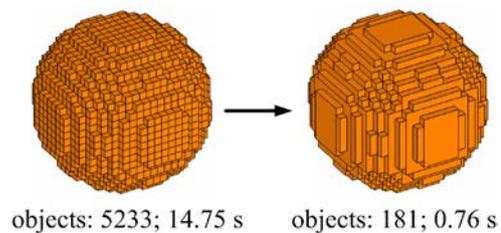


Fig. 5. Decomposition of a voxelized ball.

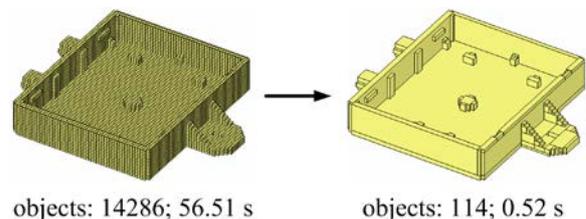


Fig. 6. Decomposition of a voxelized object of a complex geometric shape.

4 Placement of orthogonal polyhedrons

Figure 7 presents the developed algorithm for packing an orthogonal polyhedron (OP) O_i of arbitrary dimension D into a container.

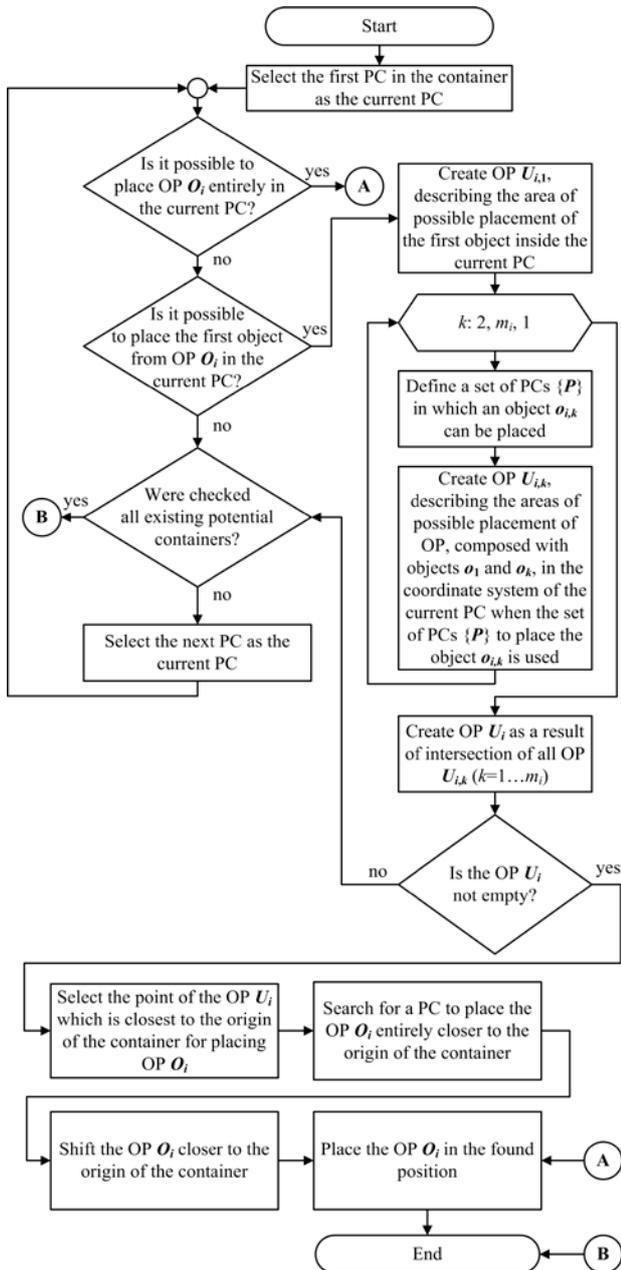


Fig. 7. Algorithm for placing an orthogonal polyhedron.

This algorithm is based on the creation of orthogonal polyhedrons that determine the possible placement areas for each object of complex shape. To reduce the complexity of the block diagram, in Figure 7 is presented a part of the algorithm which provides placing of an orthogonal polyhedron inside only one current container. Algorithm for packing orthogonal polyhedrons in a set of containers is presented in the paper [23].

The steps for sequentially determining the region of admissible placement of an orthogonal polyhedron in a given potential container are shown in Figure 7. The final orthogonal polyhedron is obtained after application

the intersection operation to all orthogonal polyhedrons describing areas of admissible placement.

Figure 9 presents an example of placement orthogonal polyhedrons into a container.

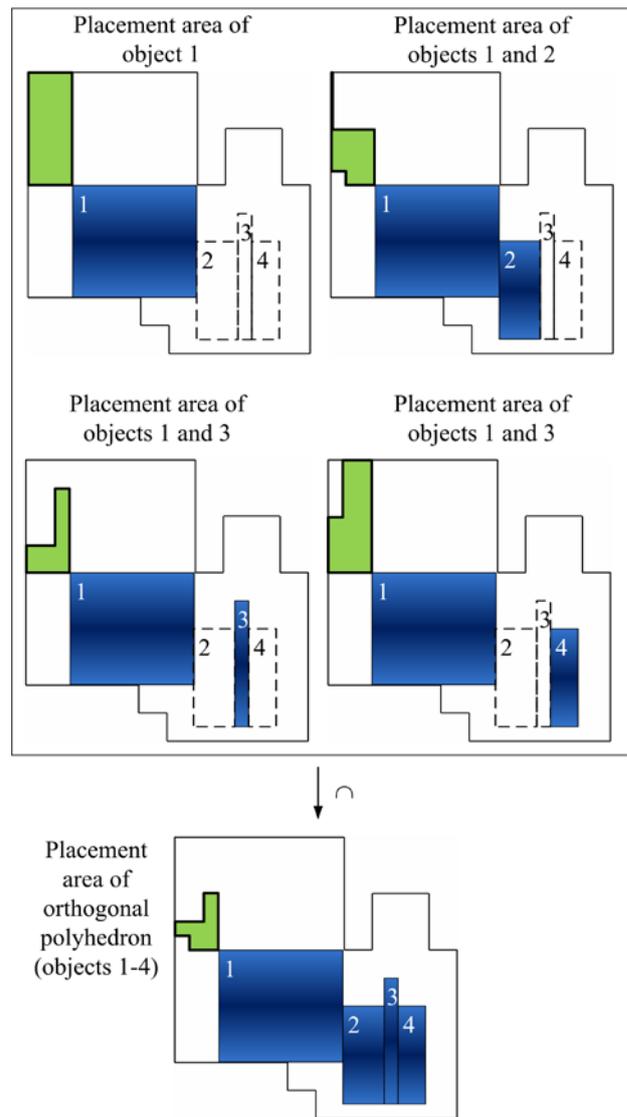


Fig. 8. Finding the admissible area of placement of an orthogonal polyhedron (filled with a gradient fill) in a potential container located in the upper left corner of the container.

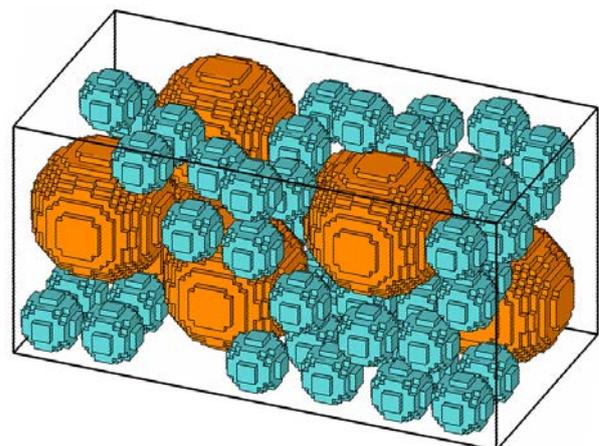


Fig. 9. Placement of voxelized balls into a parallelepiped.

5 Practical application

As an example of the practical application of the developed algorithms for the formation and placement of orthogonal polyhedrons, the problem of parts (details) layout on the platform of a 3D printer is considered.

Solving the problem of optimizing the layout of objects is of great importance in additive technologies, since it can significantly reduce the consumption of material, reduce the time spent on preparing the layout of objects inside the platform (container) of a 3D printer, as well as the time spent on their direct manufacture. The use of denser layouts additionally leads to a reduction in energy costs in the process of manufacturing objects, and also reduces the equipment depreciation used.

At present, the richest functionality for solving the problems of additive manufacturing is provided by the software Materialise Magics (Materialise NV, Lovaine, Belgium), which is used by the world's leading manufacturers of equipment for 3D printing [24]. The widespread recognition of this tool in the global market for additive technologies explains its choice for a comparative analysis of the obtained results.

We consider a three-dimensional container with the following parameters: length: 340 mm, width: 340 mm, height: 620 mm, the gap from the bottom of the container is 9 mm, the gap from the side faces of the container is 10 mm, minimum distance between placed objects is 6 mm. Parameters of details are given in Figure 10 (200 items of three types). All details can be rotated by multiples of 90° when placed.

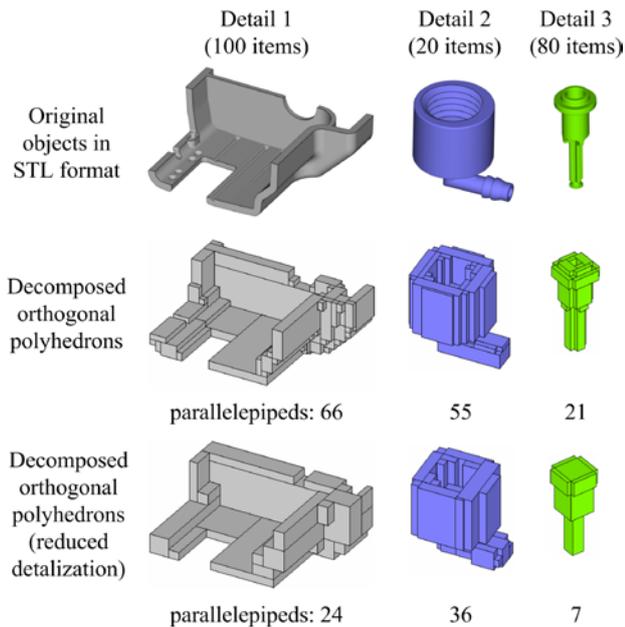


Fig. 10. Parameters of details.

Figure 11 shows the best solution obtained by the Sinter module of the software Materialise Magics, version 23.0.1.19 (layout height: 141.0 mm, solution time: 300 seconds). Figure 12 presents the denser placement of orthogonal polyhedrons found by the designed software Packer which implements all

developed algorithms [25, 26] (height: 122.4 mm, solution time: 107 seconds).

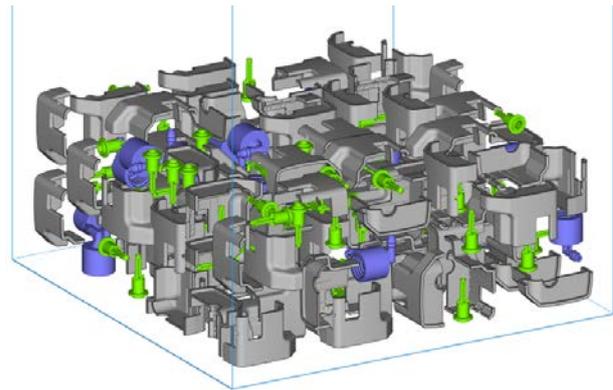


Fig. 11. The best placement obtained by Materialise Magics.

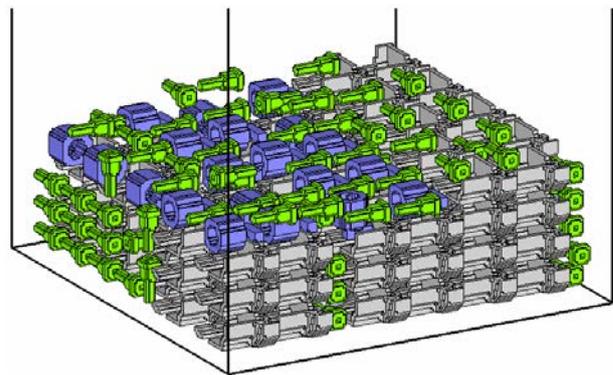


Fig. 12. Placement of orthogonal polyhedrons (decomposed voxelized objects) obtained by the developed algorithms.

Reducing the detalization on voxelization results in a significant increase in the placement speed with a slight loss of placement density. The resulting layout of details with reduced detalization is shown in Figure 13 (height: 124.4 mm, solution time: 18 seconds).

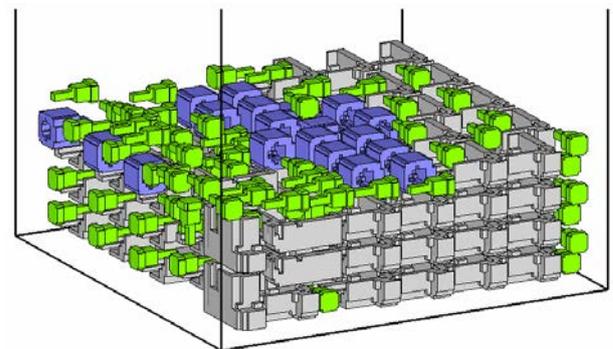


Fig. 13. Placement of orthogonal polyhedrons (decomposed voxelized objects with reduced object detalization) obtained by the developed algorithms.

6 Conclusion

The article proposes a method for solving the problem of placing objects of complex geometric shapes, which

consists in transforming it to the problem of placing orthogonal polyhedrons.

All considered algorithms are developed and implemented independently from the dimension of the problem being solved, which makes them applicable for solving arbitrary types of cutting and packing problems.

The proposed algorithm for placing an orthogonal polyhedron of arbitrary dimension solves the problem of obtaining the area closest to the origin of the container which can be used for the placement of all orthogonal objects that are part of the considered orthogonal polyhedron. As a result of analyzing the set of all existing free orthogonal areas of the container, the most suitable area of the container is determined for each orthogonal polyhedron when forming a dense placement of objects.

The efficiency of application of the developed and software-implemented algorithms for the formation and placement of orthogonal polyhedrons is investigated by the example of solving the problem of modeling the layout of objects of complex geometric shape on the platform of a 3D printer. It is shown that the algorithms presented in the work provide an average of 10% denser layouts in a several times shorter than the layouts generated by Materialize Magics software which is the market leader in software solutions for additive technologies.

References

1. G. Wäscher, H. Haußner, H. Schumann, Eur. J. Oper. Res., **183**(3), 1109-1130 (2007)
2. A. Bortfeldt, G. Wäscher, Eur. J. Oper. Res., **229**(1), 1-20 (2013)
3. T.G. Crainic, G. Perboli, R. Tadei, INFORMS J. Comput., **20**(3), 368-384 (2008)
4. R.J. Mailloux, S.G. Santarelli, T.M. Roberts, D. Luu, Int. J. Antenn. Propag., **956524** (2009)
5. V.A. Chekanin, A.V. Chekanin, Comp. Math. Math. Phys+, **53**(10), 1457-1465 (2013)
6. V.A. Chekanin, A.V. Chekanin, Applied Mechanics and Materials, **598**, 377-381 (2014)
7. Y. Stoyan, T. Romanova, A. Pankratov, A. Chugay, Springer Optimization and Its Applications, **105**, 265-293 (2015)
8. N. Chernov, Y. Stoyan, T. Romanova, Computational Geometry, **43**(5), 535-553 (2010)
9. J. Bennell, G. Scheithauer, Y. Stoyan, Ann. Oper. Res., **179**(1), 343-368 (2010)
10. E.G. Birgin, L.H. Bustamante, H.F. Callisaya, J.M. Martínez, Int. T. Oper. Res., **20**(3), 365-389 (2013)
11. E.G. Birgin, R.D. Lobato, J.M. Martínez, J. Global. Optim., **65**(4), 709-743 (2016)
12. M. Verkhoturov, A. Petunin, G. Verkhoturova, K. Danilov, D. Kurennov, IFAC-PapersOnLine, **49**(12), 1-5 (2016)
13. A.V. Tolok, N.B. Tolok, Automat. Rem. Contr+, **79**(9), 1703-1712 (2018)
14. A.C.J. de Korte, H.J.H. Brouwers, Powder Technol., **233**, 319-324 (2013)
15. T. Byholm, M. Toivakka, J. Westerholm, Powder Technol., **196**(2), 139-146 (2009)
16. V.A. Chekanin, A.V. Chekanin, Lecture Notes in Electrical Engineering, **641**, 891-900 (2020).
17. V.A. Chekanin, A.V. Chekanin, *Advances in Mechanical Engineering. Lecture Notes in Mechanical Engineering*, 33-38 (Springer, Cham, 2015)
18. V.A. Chekanin, A.V. Chekanin, *Advances in Mechanical Engineering. Lecture Notes in Mechanical Engineering*, 27-35 (Springer, Cham, 2017)
19. V.A. Chekanin, A.V. Chekanin, *Advances in Mechanical Engineering. Lecture Notes in Mechanical Engineering*, 87-92 (Springer, Cham, 2016)
20. S. Durocher, S. Mehrabi, Theor. Comput. Sci., **689**, 157-168 (2017)
21. T. Biedl, M. Derka, V. Irvine, A. Lubiw, D. Mondal, A. Turcotte, Lecture Notes in Computer Science, **10807**, 146-160 (2018)
22. P. Floderus, J. Jansson, C. Levcopoulos, A. Lingas, D. Sledneu, Algorithmica, **80**(1), 136-154 (2018)
23. V.A. Chekanin, A.V. Chekanin, *Advances in Mechanical Engineering. Lecture Notes in Mechanical Engineering*, 41-48 (Springer, Cham, 2020)
24. S. Hällgren, L. Pejryd, J. Ekengren, Procedia CIRP, **50**, 518-523 (2016)
25. V.A. Chekanin, A.V. Chekanin, Journal of Theoretical and Applied Information Technology, **88**(3), 421-430 (2016)
26. V.A. Chekanin, A.V. Chekanin, Advances in Intelligent Systems Research, **123**, 379-382 (2015)