Solving the Problem of Packing Objects of Complex Geometric Shape into a Container of Arbitrary Dimension*

Vladislav Chekanin[1][0000-0001-9448-0583]

[1] Moscow State University of Technology «STANKIN», Moscow, Russia
vladchekanin@rambler.ru

Abstract. The article is devoted to algorithms developed for solving the problem of placement orthogonal polyhedrons of arbitrary dimension into a container. To describe all free areas of a container of complex geometric shape is applied the developed model of potential containers. Algorithms for constructing orthogonal polyhedrons and their subsequent placement are presented. The decomposition algorithm intended to reduce the number of orthogonal objects forming an orthogonal polyhedron is described in detail. The proposed placement algorithm is based on the application of intersection operations to obtain the areas of permissible placement of each considered object of complex geometric shape. Examples of packing sets of orthogonal polyhedrons and voxelized objects into containers of various geometric shapes are given. The effectiveness of application of all proposed algorithms is presented on an example of solving practical problems of rational placement of objects produced by 3D printing technology. The achieved layouts exceed the results obtained by the Sinter module of the software Materialise Magics both in speed and density.

Keywords: Orthogonal Polyhedron, Voxelized Object, Packing Problem, 3D Printing.

1 Introduction

The problem of packing objects of irregular geometric shape has a large number of practical applications in various fields, including cutting of industrial materials, layout of spaces (spaces of aircraft, ships and etc.), covering problems, modeling the microstructure of materials, active electronically scanned arrays generation and other relevant problems [1–5]. All the packing problems including the classic orthogonal packing problem are NP-hard [6, 7] and require the use of heuristic [8, 9] or metaheuristic optimization algorithms [10–14]. Therefore, it is important to develop effective algorithms that provide good layout of objects at an acceptable time.

* Publication financially supported by RFBR grant №20-01-00547

The use of polygonal modeling in describing the geometric shape of objects requires the subsequent use of time-consuming algorithms for nonlinear programming when solving the problem of placing these objects using the hodograph of a vector function of dense placement [15, 16]. Due to the high computational complexity, nonlinear programming methods practically ineffective when increasing the number of packed objects. To solve the problem of packing objects of irregular geometric shape is proposed to present the objects as the orthogonal polyhedrons which combine non-overlapping orthogonal objects (rectangles or parallelepipeds in the two-dimensional or three-dimensional case, respectively) with a fixed position relative to each other [17–20]. This approach makes it possible to solve the problems of packing voxelized objects of complex geometric shape [21, 22].

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

2 Representation of Objects and Description of a Packing

2.1 Set-theoretic Operations for Working with Orthogonal Polyhedrons

To work with orthogonal polyhedron of arbitrary dimension, the set-theoretic operations of addition and intersection were implemented. When performing operation of addition of orthogonal objects, the following sets of objects are used: A – the initial set of objects, A^+ – set of objects to which the addition operation is applied, B – an intermediate set of objects, C – the resulting set of orthogonal objects. The size of a set of objects is indicated as $|A|$ (for a set A).

The operation of addition orthogonal objects (Figure 1) includes the following steps.

1. Place all objects from the set A into the set A^+. Create sets $B = \varnothing$, $C = \varnothing$.
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 $o_k \in A^+$ $(k = 2\ldots|A^+|)$ 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 $B \neq \varnothing$, then move all the objects from the set B to the set A^+, and then go to step 2.

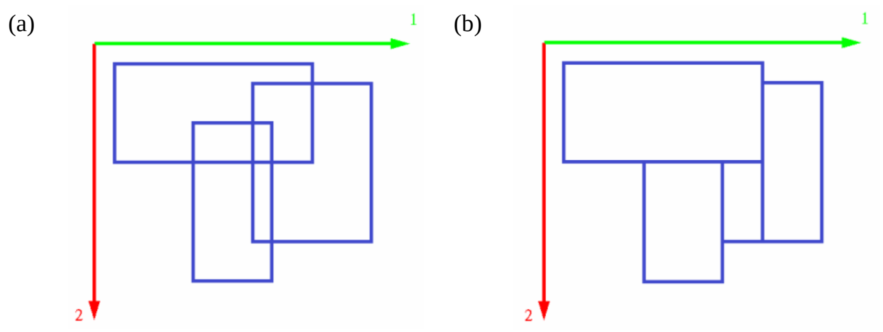

Fig. 1. The addition operation: (a) original objects; (b) result

The result of the intersection operation of orthogonal polyhedrons $O_1 \cap O_2$ is a new orthogonal polyhedron, the points of which occupy a space that belongs simultaneously to two original orthogonal polyhedrons (Figure 2).

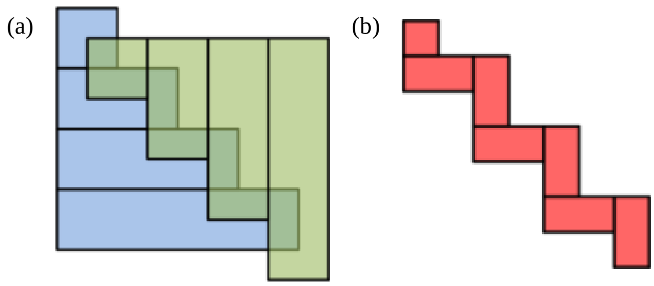

Fig. 2. The intersection operation: (a) original objects; (b) result

2.2 Model of Potential Containers

To describe a packing is used the developed model of potential containers [23, 24]. 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 $\{p_h^1; p_h^2; \ldots; p_h^D\}$ containing its dimensions and a vector $\{x_h^1; x_h^2; \ldots; x_h^D\}$ 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 $\{w_i^1; w_i^2; \ldots; w_i^D\}$ is placed at a point $\{x_i^1; x_i^2; \ldots; x_i^D\}$ it divides the potential container h into a set of smaller potential containers from two sets:

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

Figure 3 presents all new potential containers (gray) which are formed in a three-dimensional potential container after placing an orthogonal object inside it.

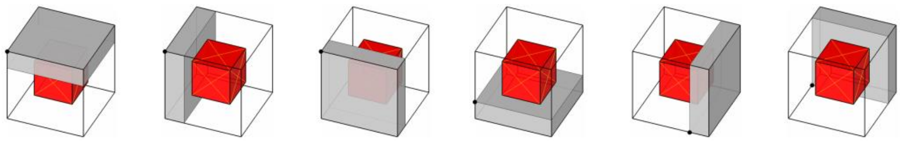

Fig. 3. New potential containers

To update a set of potential containers after placing an orthogonal polyhedron O_i at the point $\{X_i^1, X_i^2, \ldots, X_i^D\}$ 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 $h : \exists d \in \{1, \ldots, D\} : x_h^d \le X_i^d + S_i^d$, where $\{S_i^1, S_i^2, \ldots, S_i^D\}$ denotes the overall dimensions of a D-dimensional parallelepiped bounding the orthogonal polyhedron O_i: $S_i^d = \max\left(z_{i,k}^d + w_{i,k}^d\right)$, $\forall d \in \{1, \ldots, D\}$, $k \in \{1, \ldots, m_i\}$.

Step 2. Place an orthogonal polyhedron O_i in the specified position $\{X_i^1, X_i^2, \ldots, X_i^D\}$ 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 $o_{i,k}$, $k \in \{1,\ldots,m_i\}$.

Step 3. Apply the intersection operation 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''.

2.3 Decomposition of Orthogonal Polyhedron

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 $\{W_1^1, W_1^2, \ldots, W_1^D\}$ matching the dimensions of a D-dimensional parallelepiped bounding the original orthogonal polyhedron V ($W_1^d = S^d, \forall d \in \{1,\ldots,D\}$, 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 $\{w_{k_1}^1, w_{k_1}^2, \ldots, w_{k_1}^D\}$, $k_1 \in \Omega_1$ located at points $\{p_{k_1}^1, p_{k_1}^2, \ldots, p_{k_1}^D\}$ 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 ($W_2^d = W_1^d$, $\forall d \in \{1,\ldots,D\}$).

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 $w_i^d = w_{k_1}^d$ $\forall d \in \{1,\ldots,D\}$, $i = 1\ldots|\Omega_1|$. As a result in the container 2, a set of potential containers Ω_2 with the overall dimensions $\{w_{k_2}^1, w_{k_2}^2, \ldots, w_{k_2}^D\}$, $k_2 \in \Omega_2$, located at points $\{p_{k_2}^1, p_{k_2}^2, \ldots, p_{k_2}^D\}$, 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 $w_i^d = w_{k_2}^d$ $\forall d \in \{1,\ldots,D\}$, $i = 1\ldots|\Omega_2|$, 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.

Examples of decomposition of two-dimensional and three-dimensional orthogonal polyhedrons are presented in Figure 4.

To speed up the decomposition of a voxelized object, it is proposed to use a two-stage algorithm. At the first stage, the fast algorithm of objects clustering described below is applied, after which the developed decomposition algorithm is applied to the resulting orthogonal polyhedron.

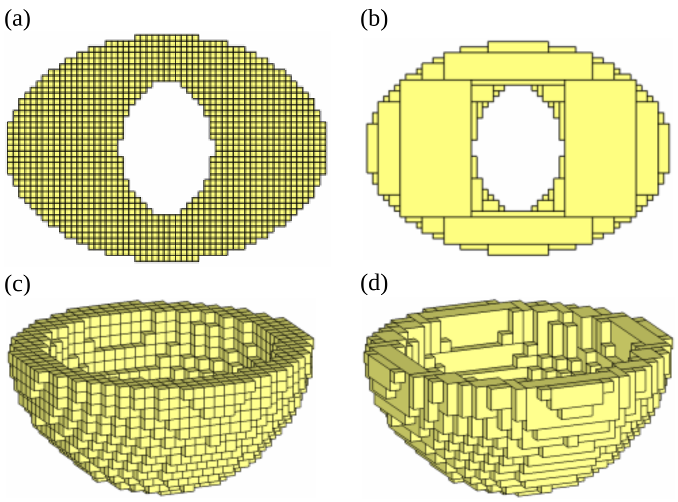

Fig. 4. Examples of decomposition of orthogonal polyhedrons: (a) original orthogonal polyhedron (1398 objects); (b) decomposed orthogonal polyhedron (72 objects); (c) original orthogonal polyhedron (2259 objects); (d) decomposed orthogonal polyhedron (308 objects)

Algorithm of clustering a voxelized object O_i includes three steps.

Step 1. Set current axis number $d := 1$.

Step 2. For each pair of orthogonal objects $o_{i,k} \in O_i$ and $o_{i,j} \in O_i$ ($k, j \in |O_i|$) create a new orthogonal object o with overall dimensions $\{w^1_{i,k} + w^1_{i,j}, w^2_{i,k} + w^2_{i,j}, \ldots, w^D_{i,k} + w^D_{i,j}\}$ located at a point $\{z^1_{i,k}, z^2_{i,k}, \ldots, z^D_{i,k}\}$ under the conditions $z^{d'}_{i,j} = z^{d'}_{i,k}$, $w^{d'}_{i,j} = w^{d'}_{i,k}$ $\forall d' \neq d$ and $z^d_{i,j} = z^d_{i,k} + w^d_{i,k}$. Replace in the orthogonal polyhedron O_i the objects $o_{i,k}$ and $o_{i,j}$ with the object o. If there are no objects $o_{i,k}$ and $o_{i,j}$ satisfying the above conditions in the orthogonal polyhedron O_i, then go to step 3, otherwise repeat step 2.

Step 3. Set $d := d + 1$. If $d \leq D$ then go to step 2.

Figure 5 presents an example of clustering and decomposition of a voxelized object. The voxelized object was clustered in 0.03 s (Figure 5, d). Two-stage algorithm decomposes it in 6.45 s (Figure 5, e) while decomposition algorithm without clustering spends 15.11 s. The computational experiments were carried out on a personal computer (Intel Core i5-8400 2.80 GHz, RAM 8.00 GB).

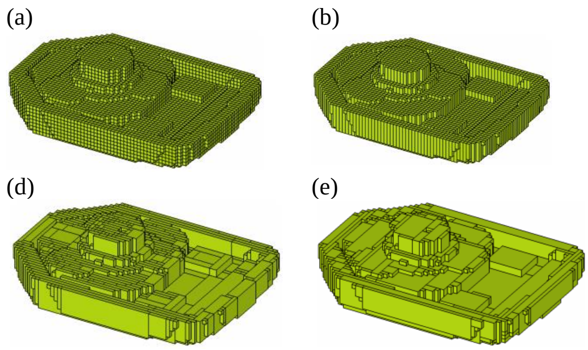

Fig. 5. Two-stage algorithm of decomposition of a voxelized object: (a) original orthogonal polyhedron (13238 objects); (b) clustered orthogonal polyhedron when $d=1$ (2503 objects); (c) clustered orthogonal polyhedron when $d=2$ (800 objects); (d) clustered orthogonal polyhedron when $d=3$ (521 objects); (e) decomposed orthogonal polyhedron (483 objects)

3 Placement of Orthogonal Polyhedrons

Figure 6 presents the developed algorithm for packing orthogonal polyhedrons of arbitrary dimension into a container.

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 6 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 in presented in the paper [25].

The steps for sequentially determining the region of possible 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 possible placement: $U_i = U_{i,1} \cap U_{i,2} \cap U_{i,3} \cap U_{i,4}$.

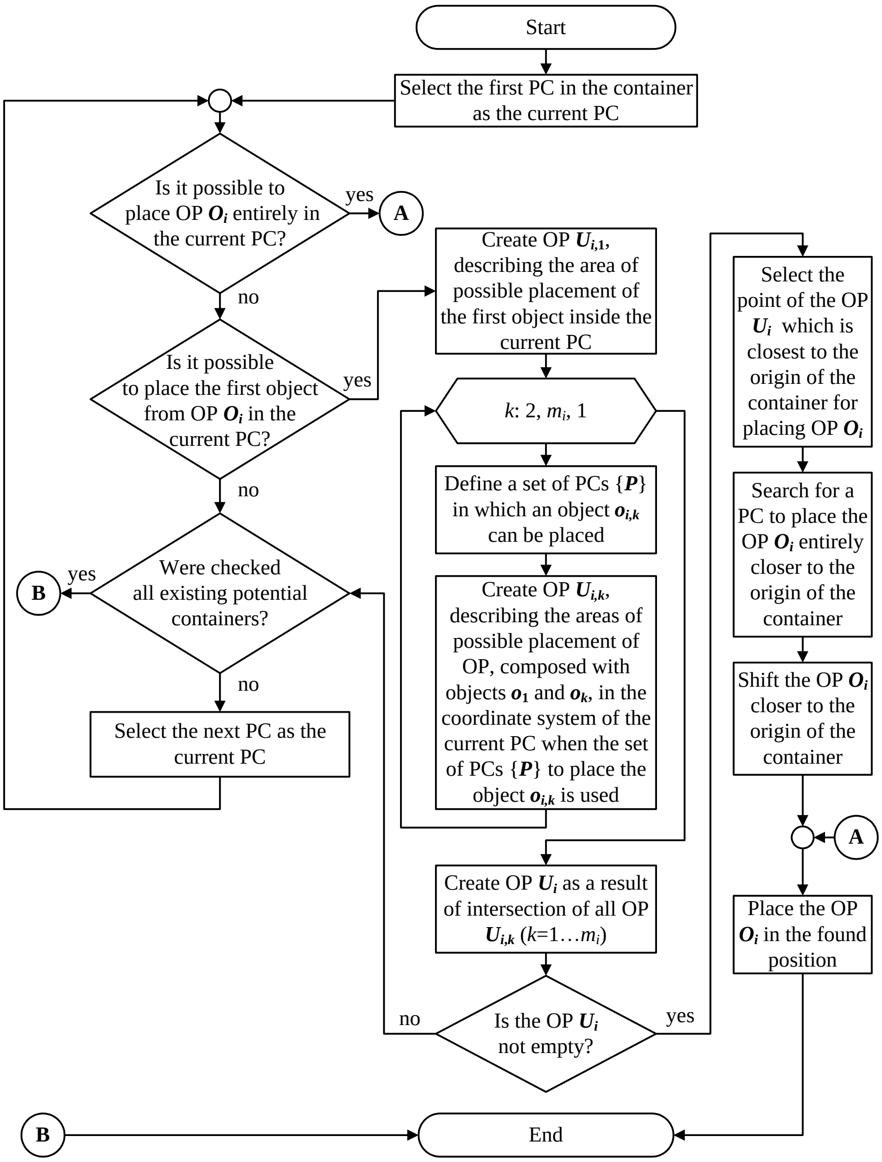

Fig. 6. Algorithm of placing an orthogonal polyhedron (OP) O_i in a container

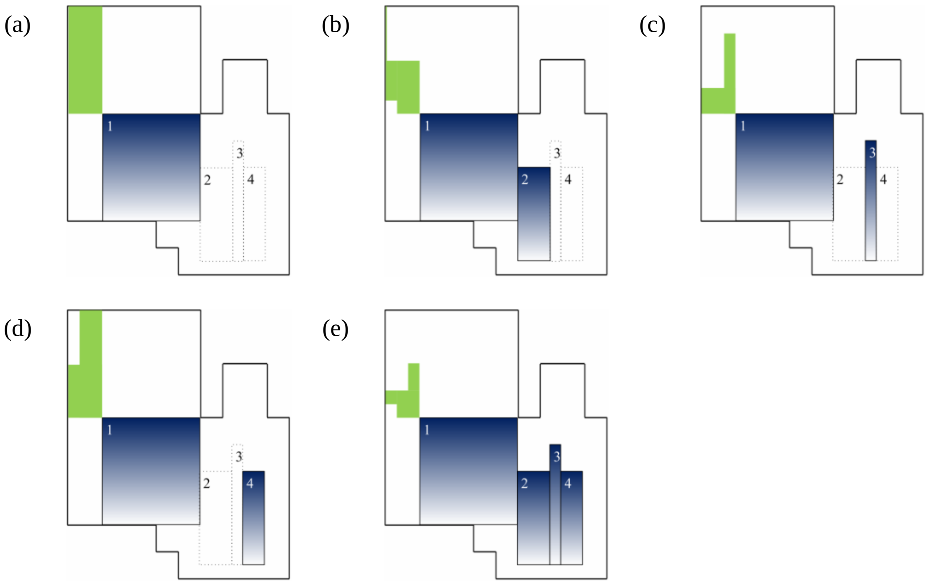

Fig. 7. Determining the area of possible placement of an orthogonal polyhedron (filled with a gradient fill) for the current potential container located in the upper left corner of the container: (a) $U_{i,1}$; (b) $U_{i,2}$; (c) $U_{i,3}$; (d) $U_{i,4}$; (e) resulting placement area U_i

To convert a container represented in the form of a D -dimensional parallelepiped into an orthogonal polyhedron, it is proposed to place a set of fictitious orthogonal objects in it that are combined into one orthogonal polyhedron of geometric constraints. Figure 8 shows a set of constraints that transform a rectangular container into an ellipse, as well as the result of packing various objects into it.

Examples of placement of orthogonal objects and orthogonal polyhedrons inside containers of complex geometric shapes are presented in Figure 9.

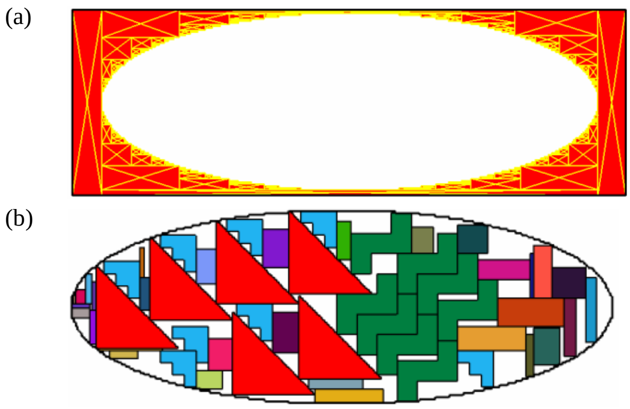

Fig. 8. Container in the form of ellipse: (a) original container with geometric constraints; (b) placement of objects in the container

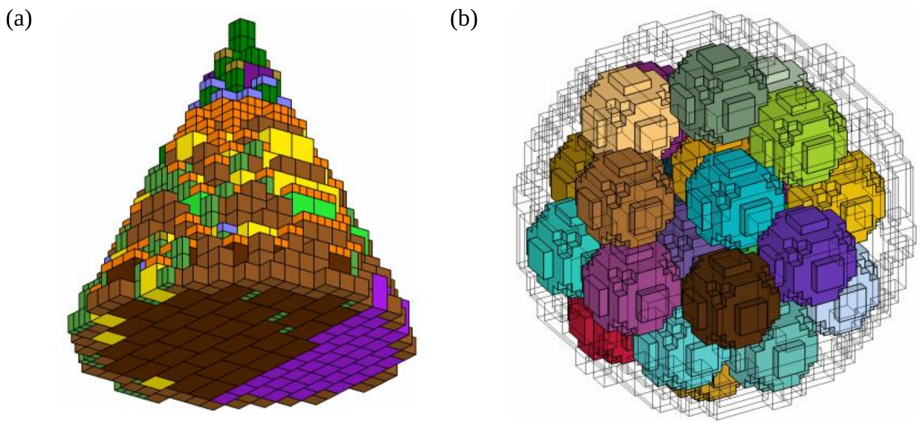

Fig. 9. Packing of objects in a container of complex geometric shape: (a) waste-free packing of parallelepipeds in a cone; (b) compact placement of balls in a sphere

4 Solving the Layout Problem

The developed algorithm for packing orthogonal polyhedrons was applied to solve practical problems of rational placement of objects produced by 3D printing.

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 [26]. 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. Objects can be rotated by multiples of 90° when placed. Figure 10 (a) shows the best solution obtained by the Sinter module of the software Materialise Magics, version 23.0.1.19 (packing height: 318.0 mm, solution time: 300 seconds). The packing consists of 100 objects of 8 different types specified in STL format. To solve this problem, using the developed placement algorithm, all objects were voxelized and decomposed. Figure 10 (b) presents the denser placement of orthogonal polyhedrons found by the developed software Packer [27] (packing height: 298.4 mm, solution time: 96 seconds). Based on the found solution, a 3D printing layout was built, it is shown in Figure 10 (c).

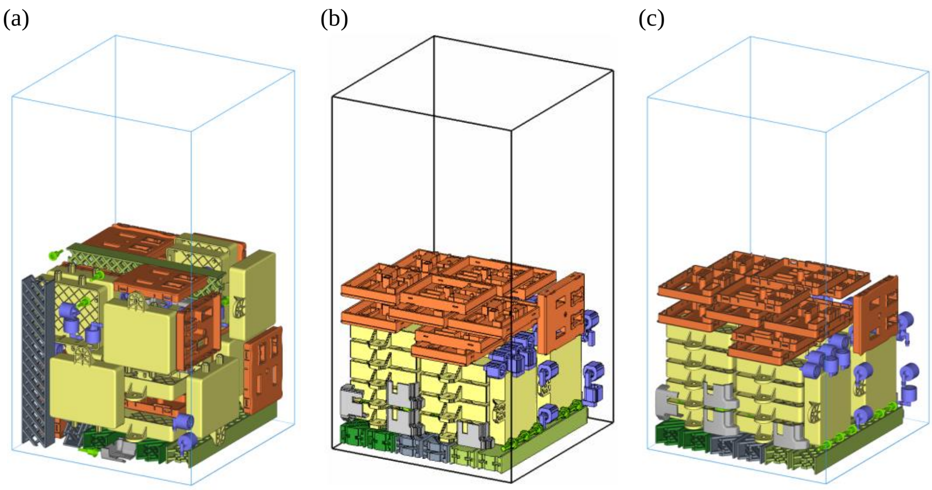

Fig. 10. Placement of irregular objects at a given minimum distance from each other: (a) the best placement obtained by Materialise Magics; (b) dense placement of orthogonal polyhedrons (decomposed voxelized objects) obtained by the developed algorithms; (c) layout for 3D printing in Materialise Magics based on the found placement of orthogonal polyhedrons

5 Conclusion

Using the developed model of potential containers to describe the free spaces inside a container, it is possible to obtain the best area for placing irregular objects inside geometrically complex containers. This model allows switch from use the time-consuming nonlinear programming methods (which are applied for placing objects specified by polygonal modeling, the practical application of which is limited only by the small size of the problem) to methods developed for placing voxelized objects.

Unlike decomposition algorithms developed by other authors [28–30], the proposed algorithm is described and programmatically implemented invariantly with respect to the dimension of the problem to be solved. The developed algorithm provides decomposition of orthogonal polyhedrons of arbitrary shape, including those with holes or internal cavities. The developed decomposition algorithm of the orthogonal polyhedrons provides an increase in the speed of placing voxelized objects by an average of two orders of magnitude.

The practical application of the developed algorithms is demonstrated by the example of solution the layout problem of objects produced by 3D printing. The found solution exceeds the best solution obtained by the software Materialise Magics both in terms of speed and packing dense.

References

1. Wäscher, G., Haußner, H., Schumann, H.: An improved typology of cutting and packing problems. European Journal of Operational Research **183**(3), 1109–1130 (2007).
2. Bortfeldt, A., Wäscher, G.: Constraints in container loading - A state-of-the-art review. European Journal of Operational Research **229**(1), 1–20 (2013).
3. Chekanin, V.A., Chekanin, A.V.: Algorithms for management objects in orthogonal packing problems. ARPN Journal of Engineering and Applied Sciences **11**(13), 8436–8446 (2016).
4. Crainic, T.G., Perboli, G., Tadei, R.: Recent advances in multi-dimensional packing problems. In: Volosencu, C. (ed.) New Technologies–Trends, Innovations and Research, pp. 99–110. InTech (2012).
5. Mailloux, R.J., Santarelli, S.G., Roberts, T.M., Luu, D.: Irregular polyomino-shaped subarrays for space-based active arrays. International Journal of Antennas and Propagation (2009). https://doi.org/10.1155/2009/956524.
6. Garey, M, Johnson, D.: Computers Intractability: a Guide to the Theory of NP-completeness. W.H.Freeman, San Francisco (1979).
7. Johnson, D.S.: A brief history of NP-completeness, 1954–2012. Documenta Mathematica **Extra Volume ISMP**, 359–376 (2012).
8. Oliveira, J.F., Neuenfeldt, Júnior A., Silva, E., Carravilla, M.A.: A survey on heuristics for the two-dimensional rectangular strip packing problem. Pesquisa Operacional **36**(2), 197–226 (2016).
9. Chen, D., Liu, J., Fu, Y., Shang, M.: An efficient heuristic algorithm for arbitrary shaped rectilinear block packing problem. Computers & Operations Research **37**(6), 1068–1074 (2010).
10. Gao, Y.Q., Guan, H.B., Qi, Z.W., Hou, Y., Liu, L.: A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. Journal of Computer and System Sciences **79**(8), 1230–1242 (2013).
11. Martinez, M.A.A., Clautiaux, F., Dell'Amico, M., Iori, M.: Exact algorithms for the bin packing problem with fragile objects. Discrete Optimization **10**(3), 210–223 (2013).
12. Kierkosz, I., Luczak, M.: A hybrid evolutionary algorithm for the two-dimensional packing problem. Central European Journal of Operations Research **22**(4), 729–753 (2014).
13. Chekanin, V.A., Chekanin, A.V.: Development of the multimethod genetic algorithm for the strip packing problem. Applied Mechanics and Materials **598**, 377–381 (2014).
14. Chekanin, V.A., Chekanin, A.V.: Design of library of metaheuristic algorithms for solving the problems of discrete optimization. In: Evgrafov, A. (eds.) Advances in Mechanical Engineering. Lecture Notes in Mechanical Engineering, pp. 25–32. Springer, Cham (2018).
15. Stoyan, Y., Romanova, T., Pankratov, A., Chugay, A.: Optimized object packings using quasi-phi-functions . Optimized packings with applications. In: Fasano, G. Pintér, J. (eds.) Optimized Packings with Applications, pp. 265–293. Springer, Cham (2015).
16. Ma, Y., Chen, Z., Hu, W., Wang, W.: Packing irregular objects in 3D space via hybrid optimization. Computer Graphics Forum **37**(5), 49–59 (2018).
17. Alvarez-Valdes, R., Carravilla, M.A., Oliveira, J.F.: Cutting and packing. In: Martí, R., Panos, P., Resende, M. (eds) Handbook of Heuristics, pp. 1–46. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-07153-4_43-1.
18. Chekanin, V.A., Chekanin, A.V.: Development of Algorithms for the Correct Visualization of Two-Dimensional and Three-Dimensional Orthogonal Polyhedrons. In: Radionov, A., Karandaev, A. (eds) Advances in Automation. RusAutoCon 2019. Lecture Notes in Electrical Engineering, vol. 641, pp. 891–900. Springer, Cham (2020).

19. Aldana-Galván, I., Álvarez-Rebollar, J.L.,,, Catana-Salazar, J.C., Marín-Nevárez, N., Solís-Villarreal, E., Urrutia, J., Velarde, C.: Beacon coverage in orthogonal polyhedra. In: 29th Canadian Conference on Computational Geometry (CCCG 2017), pp. 166–171. Ottawa (2017).
20. Bournez, O., Maler, O., Pnueli, A.: Orthogonal polyhedra: Representation and computation. In: Vaandrager, F.W., van Schuppen, J.H. (eds) Hybrid Systems: Computation and Control. HSCC 1999. Lecture Notes in Computer Science, vol. 1569, pp. 46–60. Springer, Berlin, Heidelberg (1999).
21. De Korte, A.C.J., Brouwers, H.J.H.: Random packing of digitized particles. Powder technology **233**, 319–224 (2013).
22. Tolok, A.V., Tolok, N.B.: Mathematical Programming Problems Solving by Functional Voxel Method. Automation and Remote Control **79**(9), 1703–1712 (2018).
23. Chekanin, V.A., Chekanin, A.V.: Deleting objects algorithm for the optimization of orthogonal packing problems. In: Evgrafov, A. (eds.) Advances in Mechanical Engineering. Lecture Notes in Mechanical Engineering, pp. 27–35. Springer, Cham (2017).
24. Chekanin, V.A., Chekanin, A.V.: An efficient model for the orthogonal packing problem. In: Evgrafov, A. (eds.) Advances in Mechanical Engineering. Lecture Notes in Mechanical Engineering, pp. 33–38. Springer, Cham (2015).
25. Chekanin, V.A., Chekanin, A.V.: Algorithm for the Placement of Orthogonal Polyhedrons for the Cutting and Packing Problems. In: Evgrafov, A. (eds.) Advances in Mechanical Engineering. Lecture Notes in Mechanical Engineering, pp. 41–48. Springer, Cham (2020).
26. Hällgren, S., Pejryd, L., Ekengren, J.: 3D data export for additive manufacturing-improving geometric accuracy. Procedia CIRP **50**, 518–523 (2016).
27. Chekanin, V.A., Chekanin, A.V.: Implementation of packing methods for the orthogonal packing problems. Journal of Theoretical and Applied Information Technology **88**(3), 421–430 (2016).
28. Durocher, S., Mehrabi, S.: Computing conforming partitions of orthogonal polygons with minimum stabbing number. Theoretical Computer Science **689**, 157–168 (2017).
29. Biedl, T., Derka, M., Irvine, V., Lubiw, A., Mondal, D., Turcotte, A.: Partitioning Orthogonal Histograms into Rectangular Boxes. In: Bender M., Farach-Colton M., Mosteiro M. (eds.) LATIN 2018: Theoretical Informatics. LATIN 2018. Lecture Notes in Computer Science, vol. 10807, pp. 146–160. Springer, Cham (2018)
30. Floderus, P., Jansson, J., Levcopoulos, C., Lingas, A., Sledneu, D.: 3D rectangulations and geometric matrix multiplication. Algorithmica **80**(1), 136–154 (2018).