Vol. 34, No. 4 (2024) | DOI: 10.37190/ord240409

OPEN ACCESS

Operations Research and Decisions

www.ord.pwr.edu.pl

OPERATIONS RESEARCH AND DECISIONS QUARTERLY

Graph models for identifying robot-packable patterns of pallet loading

Krzysztof Pieńkosz

[1] *Warsaw University of Technology, Institute of Control and Computation Engineering, Warsaw, Poland*

[*] *Email address: Krzysztof.Pienkosz@pw.edu.pl*

Abstract

In the logistics processes of trade business, goods delivered from warehouses to retailers are often transported on pallets. The paper concerns the issue of using a robot for automatic palletizing. In order to put an item in a given place on a partially loaded pallet, the robot must have free access to that place, i.e., the place must not be covered. A graph model and a formalized method for determining the sequence of putting goods on a pallet is proposed to avoid such collisions. It is also shown that not every packing pattern can be loaded by a robot. However, some pallet loading approaches have been identified that always guarantee a feasible robot packing.

Keywords: *distributor's pallet loading problem, robot packing, aggregated precedence graphs, extended topological sorting*

1. Introduction

Most goods delivered to warehouses and then to retail stores are transported on pallets. A pallet has a rectangular flat surface on which products are placed for storage and handling. Goods packed on the pallets are usually transported in bulk packaging, e.g. cartons, trays, etc. Most of them have rectangular shapes similar to cuboids, so later in the paper, we will call them boxes. It is assumed that boxes may be stacked on pallets only parallel to their edges. Such packing is denoted as orthogonal packing.

In warehouses, only one type of goods is stored on individual pallets. However, orders from retailers usually concern a diverse assortment. Therefore, pallets must be loaded with the required variety of goods in quantities consistent with individual orders. At the same time, it is desirable to ensure that the number of pallets used and transported is as small as possible.

To automate the palletizing process, three issues must be considered:

- a method of determining where individual goods should be placed on pallets,
- capability and flexibility of the robot used,

Received 14 November 2023, accepted 19 October 2024, published online 19 December 2024
ISSN 2391-6060 (Online)/© 2024 Authors

The costs of publishing this issue have been co-financed by the Department of Operations Research and Business Intelligence at the Faculty of Management, Wrocław University of Science and Technology, Wrocław, Poland

- an order of loading goods on pallets by the robot.

In the literature, most of papers deal only with the first issue, i.e., methods for finding the best pallet packing patterns.

This paper focuses on the problem of finding a feasible sequence of packing boxes on pallets assuming that a pallet packing pattern and the capability of the robot are known. For a given packing pattern, which specifies where individual goods should be placed on the pallet, it is not obvious in what order should they be picked and loaded by the robot. To pack a box on the pallet, the robot must have free access to the place where it is to be put. Depending on the functionality of the robot, its access requirements can be defined differently.

The main contributions of the paper are as follows.

- A formalized method for analyzing pallet packing patterns is proposed. It allows to identify whether a given packing pattern can be stacked using a robot and if the answer is positive, it determines the order of placing boxes taking into account the robot's capability.
- A new concept of graph models for representing packing patterns is proposed in the form of aggregated precedence graphs. It allows us to take into account the various manipulative capabilities of the loading robot.
- The topological sorting algorithm is generalized to include the robot's ability of loading boxes from each side of the pallet.
- Finally, the packing strategies that always create robot-packable patterns of pallet loading are identified.

The remainder of the paper is organized as follows. The next section provides a brief review of the existing approaches to the pallet loading problem and the box placement techniques that are used to determine packing patterns of the pallets. It also includes a presentation of the constraints that can be imposed in practice when loading and unloading pallets. In Sections 3 and 4 various capabilities of the robots used for packing boxes are considered. An algorithm is proposed, which taking into account the functionality of the robot, allows to check whether a given packing pattern can be loaded. If this pattern is feasible for the robot, an order of packing boxes is determined. The algorithm uses the new concepts of aggregated precedence graphs and extended topological sorting. Section 5 discusses the suitability of the box placement techniques described in Section 2 when using robots to load pallets. Finally, in Section 6 concluding remarks are presented.

2. Literature review

The pallet loading problems can be classified in two categories:

- the manufacturer's pallet loading problem (MPLP),
- the distributor's pallet loading problem (DPLP).

The manufacturer's pallet loading problem refers to the simplest form of palletizing. It occurs when there are identical boxes to be placed on a pallet to maximize space utilization. This is usually the case when goods are packed on pallets by manufacturers in warehouses. Assuming that the vertical orientation of the boxes is fixed, the MPLP can be decomposed into a two-dimensional problem. Then the task is to find a packing pattern for each layer of the pallet. A comprehensive review of various approaches for solving the MPLP is presented in [17].

More complex than the MPLP is the distributor's pallet loading problem where a set of different boxes are to be packed on the smallest number of pallets. Many different concepts of algorithms have been developed and an extensive overview of them can be found, for example, in [1, 9, 18, 20].

The DPLP is an extension of the Three-dimensional bin packing problem (3DBPP) and the container loading problem (CLP). The 3DBPP consists of packing a given set of rectangular-shaped boxes into the minimum number of identical three-dimensional containers (bins). In the CLP, only one container has to be loaded with the boxes, but in such a way that the total volume of packed boxes is maximized. In this case, not all boxes have to be loaded. A comparative review of various solution methods for the CLP and 3DBPP is presented in [1, 20].

In DPLP, the primary objective is to use as few pallets as possible, but in practice, various constraints related to transport requirements must be taken into account. Bortfeldt and Wäscher [4] provide a comprehensive review of such types of constraints, namely box orientation, vertical stability, pallet weight limit, load balance, load-bearing strength of boxes, etc. Assuming only orthogonal packing patterns and that it is allowed to put the boxes on any of their sides, six box orientations are possible. However, it is often required that the boxes are placed vertically on a defined bottom side and then the admissible number of box orientations is restricted to only two. The vertical stability requires that most of the bottom area of each box must either lie directly on the pallet or be supported by other boxes placed underneath. In practice, support of 70% may be sufficient for pallet loading if the packed pallets are wrapped in plastic foil, but in many cases, full vertical support is required [4, 11]. The pallet weight limit specifies the maximum allowed total weight of the goods packed on it. Alternatively, the height of the load is being limited instead. The load balance constraints ensure that the weight of the boxes is distributed evenly over the entire pallet. Balanced loads reduce the risk of overturning while the pallet is transported. Finally, load-bearing strength defines the maximum pressure, i.e., weight per unit of area that the box can support on its top side, so as not to be damaged.

Most of these constraints should also be considered when loading goods into the cargo space of delivery vehicles [16, 19]. However, other requirements concerning loading problems in combination with vehicle routing may also arise. For example, the order in which goods are distributed to customers and unpacked must be taken into account. These types of constraints are referred to in the literature as unloading, visibility, LIFO or multi-drop constraints [2, 6, 15, 16, 19]. They ensure that products delivered to the customer can be directly taken out of the cargo space without moving other products destined for subsequent customers along a given transport route.

The two basic decisions and operations performed by the packing algorithms are: (i) the selection of the next box that will be put on the pallet or in the container, and (ii) the choice of the place of its loading [21]. The algorithms may differ in the order in which these two operations are carried out. The set of possible positions for placing the next box is usually limited. The way these places are determined and the rules for choosing the best one also differ in these algorithms.

Martello et al. [13] proposed to consider only corner point locations where an item can be placed into an existing packing. The first box is packed in the back, bottom, and left corner of the container, and the next ones are placed in the newly created corner points of the partial packing. The corner points are characterized by the fact that there are no other boxes in front of, right of or above them, but the lines up, forward, and right of these points are adjacent to sides of certain previously packed boxes or sides of the container. The black dots in Fig. 1 show corner points for an example of partial packing and two-dimensional case. These are the only possible positions to place the corner of the box inserted next.

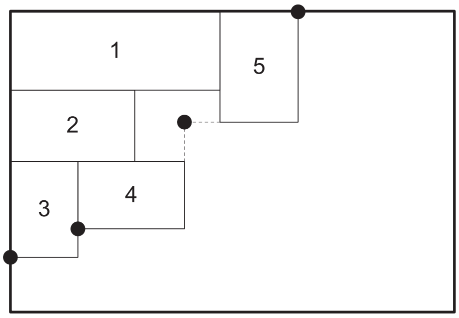


Figure 1. Corner points for 2D packing problem

Crainic et al. [7] extended this approach by introducing the concept of extreme points. These are points, which are projections of the corners of each new box added to an existing packing. Extreme points also include points inside the envelope of the existing packing and thus the number of potential placement positions of boxes is increased.

In [21], two other concepts for determining possible positions for placing a box are described. The space in the container (also called residual space) is represented by a set of cuboids adjacent to the already-packed boxes. Each subsequent box is placed only into one of these cuboids, on its bottom surface.

In the partition representation proposed by Bortfeldt et al. [3], the cuboids are interior-disjoint (i.e., non-overlapping). In the beginning, when the container is empty, we have only one cuboid of the size of this container. After packing the first box in the back, bottom, and left corner of the container, three new cuboids appear in place of the existing one – two on the side and one at the top of the box. Such a partition of residual space has a guillotine character, i.e., it is made through a sequence of cuts along the sides of the inserted box. Depending on the order in which these cuts are carried out, different variants of partition are obtained. In total, there are six such variants. Figure 2 shows the top view of the container after packing an box with the base (A, B, E, D). The upper cuboid of the partition will be adjacent to the top side of the box and, depending on the partition variant, can have the base (A, B, E, D), (A, C, F, D), (A, B, H, G) or (A, C, I, G). From above it is bounded by the height of the container. The residual space on the side of the box can be split in two ways. In the first variant, one cuboid has the base (B, C, F, E), and the other (D, F, I, G). In the second variant, one cuboid has the base (B, C, I, H), and the other (D, E, H, G). The height of these cuboids corresponds to the shape of the upper cuboid. When full vertical support of the boxes is required, only two variants of the empty space partition are considered. In this case the upper cuboid always has the base (A, B, E, D) like the box below. When using the partition representation, after inserting a new box, a clear choice from the above partition variants must be made. This decision may limit the possible locations for subsequent boxes.

In the cover representation proposed by Lim et al. [12], the residual space is represented by a set of overlapping cuboids. When a new box is placed, there is only one way to determine these cuboids. In Figure 2, the upper cuboid would adhere to the upper surface of the box and would have the base

(A, C, I, G). The two side cuboids would always have the base (B, C, I, H) and (D, F, I, G) respectively. The upper face of all three cuboids would be at the height of the upper side of the container. When full vertical support of boxes is required, the upper cuboid in the cover representation is reduced and has the base (A, B, E, D).

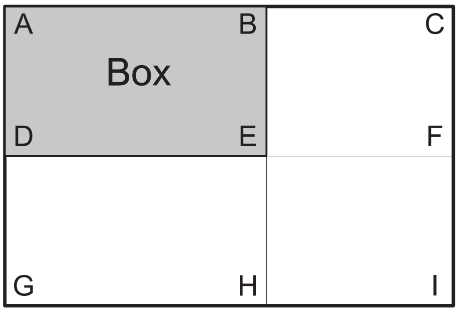


Figure 2. Residual space representation (top view)

The feasible packing of a pallet can be represented using graph models. The vertices in these graphs correspond to the individual boxes, while the undirected edges or arcs define their location relative to each other. In the graph model proposed by Fekete and Schepers [10] the three-dimensional packing patterns is represented by three undirected interval graphs. Each of these graphs corresponds to a single coordinate axis, and vertices representing boxes are connected by an edge if their projections overlap along this axis. Figure 3 illustrates such graphs for the two-dimensional packing pattern shown in Figure 1.

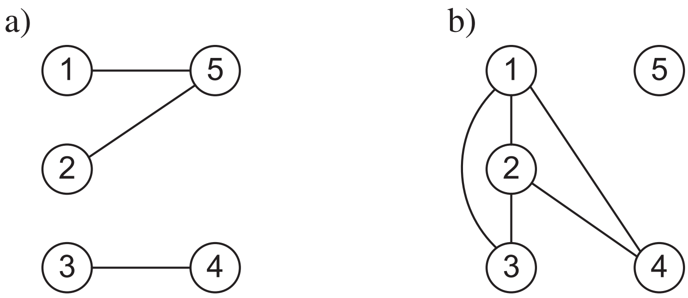


Figure 3. Interval graphs corresponding to 2D packing pattern shown in Fig 1:
a) for the horizontal projection, b) for the vertical projection

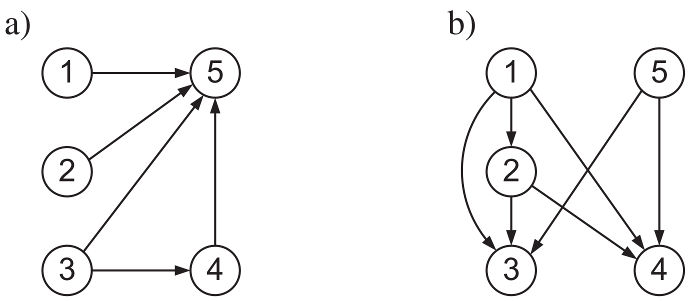


Figure 4. Comparability graphs corresponding to 2D packing pattern shown in Fig 1:
a) for the horizontal coordinate axis directed to the left, b) for the vertical coordinate axis directed downwards

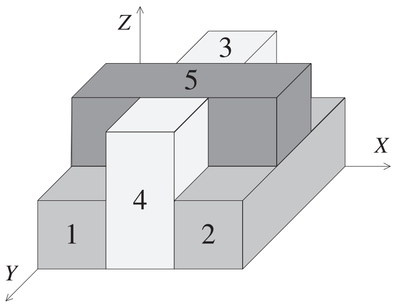
Zhu et al. [22] also used three graphs to represent three-dimensional packing pattern but slightly different ones. For each coordinate axis, they defined a comparability graph representing the mutual precedence of boxes. This is a directed graph, in which there is an arc from vertex v_i to v_j if box j lies entirely on the forward side of box i regarding the direction of a given coordinate axis. The comparability graphs for the two-dimensional packing pattern shown in Figure 1 are presented in Figure 4. It is assumed that the horizontal coordinate axis is directed to the left and the vertical axis is directed downward.

However, both of the above graph models are not adequate for determining the order in which boxes can be loaded on a pallet without causing collisions.

3. One-sided robot packings

In the case of automatic palletizing, in addition to the problem of building packing patterns, the issue of robot's ability to implement it is also important. To place a box on a pallet, the robot must have free access to the place where it is inserted. In [8], the concept of robot packing was introduced. It was defined as a packing pattern that can be created starting from the bottom, back, left corner of the pallet and placing successive boxes so that they are always above, in front of and to the right of the previously packed boxes. In this paper, such packing will be called a one-sided robot packing.

Let N be a set of the boxes packed on a pallet. The location of each box $i \in N$ will be identified by the coordinates (x_i, y_i, z_i) of its bottom, back, left corner in the coordinate system X, Y, Z, where X axis is directed to the right, Y axis forward, and Z axis upwards (as in Figure 5). The position $(0, 0, 0)$ is the bottom, back, left corner of the pallet.



Box	Dimensions l_i	w_i	h_i	Coordinates (x_i, y_i, z_i)
1	1	3	1	(0, 0, 0)
2	1	3	1	(2, 0, 0)
3	1	1	2	(1, 0, 0)
4	1	1	2	(1, 2, 0)
5	3	1	1	(0, 1, 1)

Figure 5. An example packing pattern (boxes 1 and 2 are identical, as well as boxes 3 and 4)

Suppose we consider placing a box at location (a, b, c) on a partially packed pallet. Let octant $O(a, b, c)$ be the set of points (x, y, z) such that $x > a$, $y > b$ and $z > c$. We will say that location (a, b, c) is accessible from the front of the pallet if octant $O(a, b, c)$ is disjoint from any previously loaded box.

Definition 1. The one-sided robot packing is a packing pattern that can be created by placing consecutive boxes at locations which are accessible from the front of the pallet, so that these boxes are above, in front of and to the right of the previously loaded boxes.

The one-sided robot packing assumes that in order to place another box on a pallet and avoid collisions, the robot must have free access, i.e., empty space, from the top, from the front of the pallet and from the right side of this box. It turns out that satisfaction of this condition can be not possible for some packings. Figure 5 shows an example that is not the one-sided robot packing. In the further part of the paper, we will present a method that allows to formally determine whether a given packing is a one-sided robot packing. If this is the case, this method also determines the order of packing boxes for a robot.

For a given packing, let us define X-comparability graph $G_X = (V_X, E_X)$ similarly as in [22]. It is a directed graph in which:

- Every box $i \in N$ corresponds to one vertex $v_i \in V_X$.
- There is an arc from vertex v_i to v_j if and only if $x_i + l_i \leq x_j$, where l_i is the size of box i along X axis.

Graph G_X shows the relative positions of the packed boxes along the X axis. The arc from the vertex v_i to v_j in this graph implies that box i lies entirely to the left of box j. In the same manner, we define comparability graphs G_Y and G_Z for the Y and Z axes, respectively. Figure 6 shows the comparability graphs for the example packing in Figure 5.

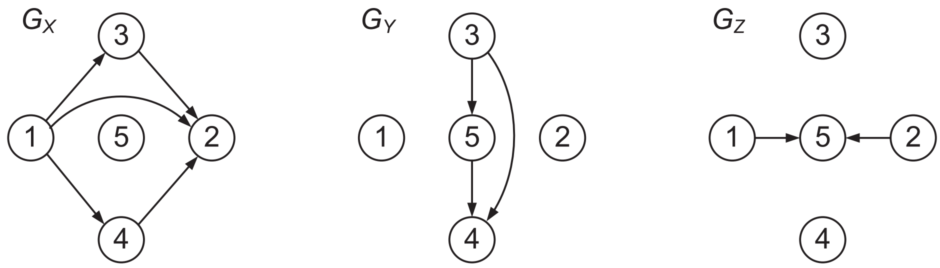


Figure 6. Comparability graphs for the example presented in Figure 5

Graphs G_X, G_Y, and G_Z are directed graphs. They have the same set of vertices but can differ in arcs. Note that each of them will always be an acyclic graph, i.e., it will have no directed cycle. For these graphs, we introduce the new concept of a aggregated precedence graph denoted as $G_X + G_Y + G_Z$. It is a directed graph in which:

- The set of vertices is the same as in G_X, G_Y and G_Z.
- There is an arc from vertex v_i to v_j if and only if in at least one of the graphs G_X, G_Y or G_Z there is an arc from v_i to v_j and in none of these graphs there is an arc directed reversely, i.e., from v_j to v_i.

The aggregated precedence graph determines the required mutual sequence in which the robot should pack each pair of boxes. Unlike the models proposed by Fekete and Schepers [10] as well as by Zhu et al. [22], the packing pattern is represented here by only one graph not three. Note that if there is an arc from v_i to v_j in graph $G_X + G_Y + G_Z$, then box j is either above, or in front of, or to the right of box i, and therefore must be packed later than i. On the other hand, if in one of the graphs G_X, G_Y or G_Z there is an arc from vertex v_i to v_j, and an opposite arc in the other, then the boxes i and j in fact do not interfere with each other, and therefore in the aggregated precedence graph there will be no arc between such vertices. For example, in the case of the packing shown in Figure 1, there would be the arc from v_4 to v_5 in graph G_X, and the opposite arc from v_5 to v_4 in graph G_Y. Note that the boxes 4 and 5, do not cover each other along the X and Y axes.

Theorem 1. A pattern is a one-sided robot packing if and only if the aggregated precedence graph $G_X + G_Y + G_Z$ for that pattern is acyclic.

Proof. Suppose that there is a directed cycle in graph $G_X + G_Y + G_Z$ and let the vertices v_i and v_j belong to this cycle. This means that box i must be packed earlier than box j and at the same time box j would have to be packed earlier than box i, which is impossible. Thus, graph $G_X + G_Y + G_Z$ for the one-sided robot packing cannot contain a directed cycle.

On the he other hand, if $G_X + G_Y + G_Z$ is a directed acyclic graph, its vertices can be numbered so that for each arc going from v_i to v_j in this graph, vertex v_i will be assigned a smaller number than v_j.

The vertex numbers will then determine the sequence in which the robot can load the individual boxes as to avoid collisions. □

Graph $G_X + G_Y + G_Z$ will not always be an acyclic graph. In computer science, ordering the vertices according to the orientation of the arcs, and at the same time checking whether a directed graph is acyclic, is called topological sorting [5]. A directed graph can be topologically sorted if and only if it is acyclic.

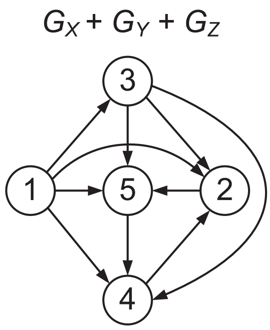


Figure 7. Graph $G_X + G_Y + G_Z$
for the example presented in Figure 5

Figure 7 shows the aggregated precedence graph $G_X + G_Y + G_Z$ for the packing pattern presented in Figure 5 and comparability graphs G_X, G_Y, G_Z depicted in Figure 6. Note that in this graph, vertices 2, 5 and 4 form a directed cycle. Thus, the pattern in Figure 5 is not a one-sided robot packing. The cycle reflects the following situation. Box 5 must be placed later than box 2, because it lies on it, and earlier than box 4, which is in front of box 5. Thus, box 4 would have to be placed later than box 2. However, this is impossible because box 2 is covered by box 4 from the right.

4. Multi-sided robot packings

The one-sided robot packing refers to a situation in which boxes are packed by a robot operating from only one side of the pallet. Starting from the left corner in the back of the pallet, it can place each next box either in front, on the right or on top of the previously placed ones. However, there are also more flexible robot designs with the arm above the pallet that allow packing from either side of the pallet. Taking into account the capabilities of such robots, we introduce the concept of a multi-sided robot packing.

Definition 2. The multi-sided robot packing is a packing pattern that can be created by placing consecutive boxes at locations which are accessible from at least one side of the pallet.

When stacking multi-sided robot packing, it is assumed that the robot can place a box from any side of the pallet. However, it must have empty space on that side of the pallet, as well as on the top and one of the sides of the box, so that it doesn't have to push it in between other boxes. Note that if a place is accessible from a given side of the pallet, this place is not covered from that side of the pallet, as well as from its right side. Thus, the box can then be loaded by the robot from any of these sides. Of course, any one-sided robot packing is also a multi-sided robot packing.

It turns out that the graph models proposed in the previous section can also be used to analyze the multi-sided robot packings. Let $\overline{G}_X$ be a directed graph that differs from the comparability graph G_X in that all the arcs are oppositely directed. Thus, this graph is also a comparability graph, but in opposite direction to the X axis, and describes the acceptable packing order of boxes from this side. Similarly, let us define the comparability graph $\overline{G}_Y$. To take into account the robot's capability to pack boxes from each of the four sides of the pallet, let us create the following four aggregated precedence graphs: $G_X + G_Y + G_Z, G_X + \overline{G}_Y + G_Z, \overline{G}_X + G_Y + G_Z, \overline{G}_X + \overline{G}_Y + G_Z$.

For a set of directed graphs $\mathcal{G}$ having the same set of vertices V, let us define the algorithm of the extended topological sorting which will be referred to as Algorithm 1.

Require: set $\mathcal{G}$ of directed graphs with $|N|$ vertices
  $k \leftarrow |N|$
  while $k > 0$ **do**
    Find a vertex v with no outgoing arc in at least one of the graphs $G \in \mathcal{G}$
    if such a vertex exists **then**
      Assign the number k to this vertex
    else
      stop {there are cycles in each graph $G \in \mathcal{G}$ so it is impossible to number all the vertices}
    end if
    Remove vertex v from all graphs $G \in \mathcal{G}$ and all arcs entering or leaving this vertex
    $k \leftarrow k - 1$
  end while
  return

Algorithm 1. Extended topological sorting of directed graph set $\mathcal{G}$

Note that when performing extended topological sorting of graph set $\mathcal{G} = \{G_X + G_Y + G_Z,\ G_X + \overline{G}_Y + G_Z,\ \overline{G}_X + G_Y + G_Z,\ \overline{G}_X + \overline{G}_Y + G_Z\}$ we are in fact determining how to unload a pallet, removing in each step one box that is not covered from the top and two adjacent sides of the pallet. So, the pallet loading can be done in the reverse order. The numbers assigned to the vertices during extended topological sorting will therefore specify the sequence of packing the boxes by the robot.

Theorem 2. A pattern is a multi-sided robot packing if and only if it is possible to number all vertices by performing extended topological sorting of graph set $\mathcal{G} = \{G_X + G_Y + G_Z,\ G_X + \overline{G}_Y + G_Z,\ \overline{G}_X + G_Y + G_Z,\ \overline{G}_X + \overline{G}_Y + G_Z\}$.

Proof. The numbering of all vertices is not possible during extended topological sorting if in a certain iteration we have directed cycles in each of the graphs $G \in \mathcal{G}$. This means that it is impossible for the robot to load any subsequent box from either side of the pallet. Thus, the packing pattern represented by the set of graphs $\{G_X + G_Y + G_Z, G_X + \overline{G}_Y + G_Z, \overline{G}_X + G_Y + G_Z, \overline{G}_X + \overline{G}_Y + G_Z\}$ cannot then be a multi-sided robot packing.

On the other hand, if the extended topological sorting is successfully completed, the numbers assigned to the vertices will determine the feasible loading sequence of a multi-sided robot packing. □

Let us note that the extended topological sorting of only one graph reduces to the simple topological sorting. In particular, if the set $\mathcal{G}$ consists only of graph $G_X + G_Y + G_Z$, then Algorithm 1 will verify the one-sided robot packing.

Analyzing the packing pattern in Figure 5, it can be seen that it is not a multi-sided robot packing. Comparability graphs G_X, G_Y and G_Z for this packing are shown in Figure 6, while the corresponding aggregated precedence graphs $G_X + G_Y + G_Z$, $G_X + \overline{G}_Y + G_Z$, $\overline{G}_X + G_Y + G_Z$ and $\overline{G}_X + \overline{G}_Y + G_Z$ in

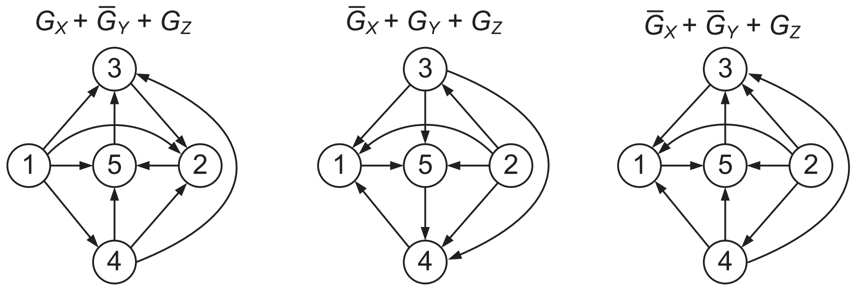


Figure 8. The aggregated precedence graphs for the example presented in Figure 5

Figure 7 and 8. Note that in all these four aggregated precedence graphs each vertex has an outgoing arc. Thus, Algorithm 1 will stop in the first iteration because it will not find any vertex to assign the number 5 and, as a result, the vertices will not be numbered.

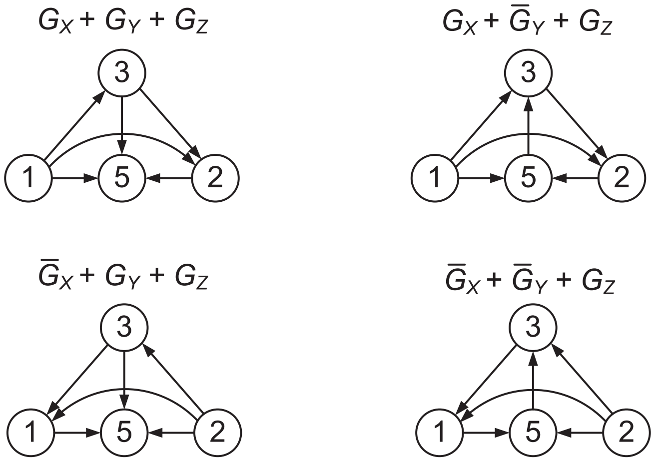


Figure 9. The aggregated precedence graphs without vertex 4

Let us now consider the case when there is no box 4 in the packing shown in Figure 5. Then there will be no vertex 4 in the aggregated precedence graphs and these graphs will be reduced to the form depicted in Figure 9. If we apply Algorithm 1 to the set of these four graphs, then the process of finding a loading sequence for a robot will be as follows.

- In the first iteration, vertex 5 will be selected and assigned number 5 because it has no outgoing arc in graphs $G_X + G_Y + G_Z$ and $\overline{G}_X + G_Y + G_Z$ shown in Figure 9. This means that box 5 will be packed as the last one by the robot. Since the horizontal components of graphs $G_X + G_Y + G_Z$ and $\overline{G}_X + G_Y + G_Z$ are graphs G_Y, G_X and $\overline{G}_X$, box 5 can be loaded from the front side of the pallet, as well as from its left or right side.
- In the second iteration of Algorithm 1, vertex 1 can be selected because, after removing vertex 5, it will have no outgoing arcs in graphs $\overline{G}_X + G_Y + G_Z$ and $\overline{G}_X + \overline{G}_Y + G_Z$. Thus, vertex 1 can be packed just before vertex 5 from the left side of the pallet or from its front or back side.
- In the third iteration, vertex 2 can be selected, then having no outgoing arcs in the graphs $G_X + G_Y + G_Z$ and $G_X + \overline{G}_Y + G_Z$. It can be placed from the right side of the pallet, as well as from its front or back side.
- In the last iteration, only vertex 3 will remain and therefore it will be packed first by the robot from any side of the pallet.

Since it is possible to number all the vertices, the aggregated precedence graphs shown in Figure 9 represent the multi-sided robot packing. The resulting box packing sequence is 3-2-1-5. It can be seen that the packing pattern considered above is not only a multi-sided robot packing, but even a one-sided robot packing. We can check this by applying Algorithm 1 to only one graph $G_X + G_Y + G_Z$ shown in Figure 9. In that case, vertex 5 will be numbered first, then vertex 2, next vertex 3 and finally vertex 1. Thus, the resulting box packing order will be 1-3-2-5 which can be realized entirely from the front side of the pallet.

The proposed technique of analyzing packing patterns for its feasibility can also be applied to various other access requirements that may exist when packing boxes by a robot. Consider, for example, the case where the robot needs to have access from the top and only from the front side of the pallet (Y axis), but is able to insert a box between other placed boxes. As with visibility constraints [2, 15], it is only required that the box is visible from above and from the loading area, not necessarily having an empty space on the side. To determine the packing order in such a situation, the aggregate precedence graph can also be used, but it needs to be created from a slightly modified comparability graphs.

Let H_Y denotes the graph formed from G_Y by removing from it the arcs between those vertices that are connected by any arc in graph G_X. Thus, there are arcs in the graph H_Y only between those vertices that represent boxes that block each other along the Y coordinate axis. This graph can be seen as an oriented version of the interval graph proposed by Fekete and Schepers [10].

To find the feasible packing sequence, this time aggregated preceding graph $H_Y + G_Z$ should be created and the extended topological sorting applied to it. For the packing pattern shown in Figure 5, graph H_Y is the same as G_Y and $H_Y + G_Z$ has the form as in Figure 10. The only vertex with no outgoing arc in is vertex 4, so this vertex will be assigned number 5 in the first iteration of Algorithm 1. After removing vertex 4 and all arcs entering it, vertex 5 will be chosen in the second iteration. In the third iteration, there will be three vertices without any arcs, so their numbering can be continued in any order. Thus, graph $H_Y + G_Z$ can be topologically sorted and the packing pattern shown in Figure 5 would be feasible for such a robot. One possible order of packing the boxes is 1-2-3-5-4.

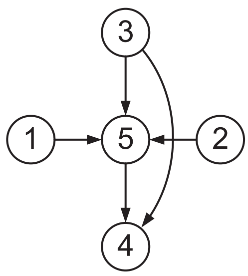


Figure 10. Graph $H_Y + G_Z$ for the example presented in Figure 5

On the other hand, if the robot were capable to place boxes from any side of the pallet, having only free access from that side and from the top, then the extended topological sorting would have to be applied to the set consisting of graph $H_Y + G_Z$ and three additional aggregated precedence graphs corresponding to the other sides of the pallet.

To implement the proposed method for identifying robot-packable patterns of pallet loading, we must first create the comparability graphs G_X, G_Y, G_Z based on the location of the boxes and their dimensions. Next, depending on the functionality of the robot, an appropriate set of aggregated precedence graphs should be built. Finally, the extended topological sorting of these graphs is performed. The generation of aggregated precedence graphs requires comparing the position on the pallet of each pair of boxes, so it takes $O(|N|^2)$ time. The algorithm of the extended topological sorting runs in $O(|N|^2)$ time, since it consists in vertex search and arcs removal operations. Thus, the running time of the complete algorithm for determining the feasible order in which the robot could place the boxes on a pallet is $O(|N|^2)$.

5. Pallet loading strategies for robot packing

In the previous sections it is shown how to check whether a given packing pattern can be stacked on a pallet by a robot. From a practical point of view, it may be important to know which of the packing strategies discussed in Section 2 always create patterns that are feasible for the robot.

It was noted in [8] that the strategies for placing elements at corner points which were used, for example, in the algorithm described in [13], give us always a one-sided robot packing. This is because in such packing, no already packed box is positioned in front of, right of, or above the destination of the currently placed box.

The extreme points idea extends the corner points concept. However, by placing elements at the extreme points, we may not get the one-sided or even multi-sided robot packing. Let us consider the packing pattern shown in Figure 5 with an additional box 6 filling the empty space under box 5. The dimensions of box 6 are $1 \times 1 \times 1$ and its placement coordinates are $(1, 1, 0)$. Such an extended packing pattern can be formed by placing boxes at the extreme points, for example, in order: 1-3-6-2-5-4 (without box 6 there would be no extreme point to insert box 5). Note that, like the packing in Figure 5, this extended packing with additional box 6 would not be the one-sided and even multi-sided robot packing.

It is shown in [14] that every guillotine-cuttable packing is also a one-sided robot packing. Since packing strategy based on partition representation of residual space give us only guillotine-cuttable packing patterns, so using this technique we always get a one-sided robot packing. However, it is worth noting at this point that not all one-sided robot packings are guillotine-cuttable. An example of a pattern that is one-sided robot pacing but is not guillotine-cuttable is shown in [14].

The cover representation of the residual space offers the greatest freedom in choosing where to place the box on a pallet. However, using this technique, unfortunately, we can get a pattern that is neither the one-sided robot packing nor even multi-sided robot packing. An example of such a situation is the packing pattern depicted in Figure 5. As shown earlier, this is not the multi-sided robot packing. On the other hand, it can be obtained using the cover representation if the boxes are placed in the order 1-3-2-5-4.

We will show below that if there is an additional practical requirement for full vertical support of the stacked boxes when loading the pallet, then the cover representation will nevertheless always give us the one-sided robot packing. A box is fully supported when its entire bottom surface lies directly on the pallet or on some other boxes.

Let us first consider the two-dimensional case of packing only one layer of boxes on a rectangular surface, that is, including only those boxes that directly lie on such a surface. We will call such packing a layer packing of boxes.

Theorem 3. Every layer packing of boxes is a one-sided robot packing.

Proof. Graphs G_X, G_Y and H_Y are acyclic, and there are no arcs between the vertices in graph G_Z for the layer packing. Let T be a set of vertices with no outgoing arcs in oriented interval graph H_Y and t be a vertex with no outgoing arc in graph $G_X|T$, i.e., subgraph of G_X induced by the vertices from set T. Note that then no arc can come out from vertex t in graph $G_X + G_Y + G_Z$. So this vertex can be assigned the largest number in topological sorting. Removing vertex t from graphs G_X, G_Y, G_Z and all arcs entering it, we can continue topological sorting in this way and number all the vertices in decreasing order. So, graph $G_X + G_Y + G_Z$ of the layer packing is always acyclic and it follows from Theorem 1 that this is a one-sided packing. □

Theorem 4. The pallet loading strategy based on cover representation in full vertical support variant always leads to a one-sided robot packing.

Proof. It follows from Theorem 3 that boxes lying directly on the pallet form a one-side robot packing and can be loaded in a topological order $b_1, b, \ldots, b_n$. In the case of full vertical support, every other box is entirely above one of these lower boxes. So the robot with one-sided ability can first place box b_1 on a pallet and the individual layers of boxes lying above it in the order determined recursively, then box b_2 and the layers of boxes located above it, and so on. The following pseudo-code illustrates this recursive topological ordering method.

```
SEQUENCING(L, k)
repeat
  box ← NextTopological(L)
  k ← k + 1
  Assign the number k to box
  U ← UpperLayer(box)
  if U ≠ ∅ then
    SEQUENCING(U, k)
  end if
until box is the last one in layer L
return k
```

At the beginning of this algorithm, $k = 0$ and L is the set of boxes lying directly on the pallet. Function $UpperLayer(box)$ identifies the set of boxes (box layer) that lie directly on the top of a given box, while function $NextTopological(L)$ gives consecutive box according to the topological loading order of such a layer. By placing the boxes on the pallet in the order determined by the above recursive procedure, the robot will always have free access to their locations from the top, front and right side. □

Note that Theorem 4 can be generalized to any packing method in full vertical support variant with the restriction that each box can lie either at the bottom of the pallet or entirely on only one other box. This also applies to the extreme point insertion technique.

Corollary 1. The pallet loading strategy based on extreme point insertion in full vertical support variant with the restriction that each box can lie entirely only on just one box, always leads to a one-sided robot packing.

It is worth noting that if we allow that boxes can be supported not only by one, but several other boxes, then using extreme points technique, we may not even get the multi-sided robot packing. The packing pattern from Figure 5 extended by additional box 6 filling the empty space under box 5 meets the conditions of full vertical support, with box 5 being fully supported by three boxes: 1, 6 and 2. As mentioned earlier, we can get this pattern by loading the boxes at the extreme points, but it is not a multi-sided robot packing.

6. Conclusions

In the paper a formalized method for analyzing pallet packing patterns is proposed. It allows to identify whether a given packing pattern can be stacked using a robot and if the answer is positive, it determines the order of placing boxes taking into account the robot's capability. This method uses a graph representation of packing patterns and is flexible enough to take into account various manipulative abilities of the loading robot. Consequently, it can be uniformly applied to an arbitrarily defined concept of robot packing, resulting from the required free space for the robot's moves when stacking successive goods on a pallet.

It is also shown which packing strategies and their variants leads to the robot packings. The corner point insertion technique and partition representation give us always a one-sided robot packing. The extreme point insertion technique and cover representation do not guarantee this in general. In practical cases, when full vertical support is required, the use of cover representation provides the one-sided robot packing. The same applies to the extreme points insertion technique in the case when boxes must be fully supported by only one other box or lie directly on the pallet.

References

[1] Ali, S., Ramos, A. G., Carravilla, M. A., and Oliveira, J. F. On-line three-dimensional packing problems: A review of off-line and on-line solution approaches. *Computers & Industrial Engineering 168* (2022), 108122.

[2] Bonet Filella, G., Trivella, A., and Corman, F. Modeling soft unloading constraints in the multi-drop container loading problem. *European Journal of Operational Research 308*, 1 (2023), 336–352.

[3] Bortfeldt, A., Gehring, H., and Mack, D. A parallel tabu search algorithm for solving the container loading problem. *Parallel Computing 29*, 5 (2003), 641–662.

[4] Bortfeldt, A., and Wäscher, G. Constraints in container loading – a state-of-the-art review. *European Journal of Operational Research 229*, 1 (2013), 1–20.

[5] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to Algorithms*. The MIT Press, 2009.

[6] Côté, J.-F., Gendreau, M., and Potvin, J.-Y. An exact algorithm for the two-dimensional orthogonal packing problem with unloading constraints. *Operations Research 62*, 5 (2014), 1126–1141.

[7] Crainic, T. G., Perboli, G., and Tadei, R. Extreme point-based heuristics for three-dimensional bin packing. *INFORMS Journal on Computing 20*, 3 (2008), 368–384.

[8] den Boef, E., Korst, J., Martello, S., Pisinger, D., and Vigo, D. Erratum to the three-dimensional bin packing problem: Robot-packable and orthogonal variants of packing problems. *Operations Research 53*, 4 (2005), 735–736.

[9] Elhedhli, S., Gzara, F., and Yildiz, B. Three-dimensional bin packing and mixed-case palletization. *INFORMS Journal on Optimization 1*, 4 (2019), 323–352.

[10] Fekete, S. P., and Schepers, J. A combinatorial characterization of higher-dimensional orthogonal packing. *Mathematics of Operations Research 29*, 2 (2004), 353–368.

[11] Gzara, F., Elhedhli, S., and Yildiz, B. C. The pallet loading problem: three-dimensional bin packing with practical constraints. *European Journal of Operational Research 287*, 3 (2020), 1062–1074.

[12] Lim, A., Rodrigues, B., and Wang, Y. A multi-faced buildup algorithm for three-dimensional packing problems. *Omega 31*, 6 (2003), 471–481.

[13] Martello, S., Pisinger, D., and Vigo, D. The three-dimensional bin packing problem. *Operations Research 48*, 2 (2000), 256–267.

[14] Martello, S., Pisinger, D., Vigo, D., den Boef, E., and Korst, J. Algorithm 864: General and robot-packable variants of the three-dimensional bin packing problem. *ACM Transactions on Mathematical Software 33*, 1 (2007), 7.

[15] Martínez, D. A., Alvarez-Valdes, R., and Parreño, F. A grasp algorithm for the container loading problem with multi-drop constraints. *Pesquisa Operacional 35*, 1 (2015), 1–24.

[16] Pollaris, H., Braekers, K., Caris, A., Janssens, G., and Limbourg, S. Vehicle routing problems with loading constraints: state-of-the-art and future directions. *OR Spectrum 37*, 2 (2015), 297–330.

[17] Silva, E., Oliveira, J., and Wäscher, G. The pallet loading problem: a review of solution methods and computational experiments. *International Transactions in Operational Research 23*, 1-2 (2016), 147–172.

[18] Silva, E. F., Toffolo, T. A. M., and Wauters, T. Exact methods for three-dimensional cutting and packing: A comparative study concerning single container problems. *Computers & Operations Research 109* (2019), 12–27.

[19] Wang, F., Tao, Y., and Shi, N. A survey on vehicle routing problem with loading constraints. In *2009 International Joint Conference on Computational Sciences and Optimization* (Sanya, China, 2009), vol. 2, IEEE, pp. 602–606.

[20] Zhao, X., Bennell, J. A., Bektas, T., and Dowsland, K. A comparative review of 3D container loading algorithms. *International Transactions in Operational Research 23*, 1-2 (2016), 287–320.

[21] Zhu, W., Oon, W., Lim, A., and Weng, Y. The six elements to block-building approaches for the single container loading problem. *Applied Intelligence 37* (2012), 431–445.

[22] Zhu, W., Zhang, Z., Oon, W.-C., and Lim, A. Space defragmentation for packing problems. *European Journal of Operational Research 222*, 3 (2012), 452–463.