Delft University of Technology

Faculty of Aerospace Engineering

2-Dimensional Vertical Bin Packing Optimization

AE4446 - Airport and Cargo Operations

Professor Alessandro Bombelli

Group #25

Timotei

Zach

Date: May 5, 2023

Table of Contents

1. Introduction

In the context of air cargo, an optimal bin packing strategy may reduce the number of flights required to transport goods, reduce greenhouse gas emissions, and/or improve the safety of the transported products. The number of flights and emissions may be reduced by minimizing the excess space in each used container. The safety of the transported products may be improved by considering their attributes i.e. fragility, perishability and radioactivity.

In order to determine an optimal bin packing strategy a mixed linear integer program (MILP) approach may be used to model the given problem. Ideally, a MILP is developed considering all 3-dimensions of space and gravity, however, such a model is complex in nature and is computationally expensive to solve. Leading to infeasibility in air cargo operations due to inadequate computation time. Therefore, a simplification is made to consider only 2-dimensions and gravity. The objective is to develop an optimal packing strategy that ensures all items are packed while minimizing the cost of the used ULDs.

2. Mathematical Model

This section describes the sets, parameters, decision variables, objectives, and constraints of the implemented mathematical model. Some simplifications are made in addition to only considering a 2D case. These include ignoring weight restrictions and assuming that no cut is present in the LD3 container type, meaning both container types are rectangles.

2.1 Sets

A set of *n* rectangular boxes to be packed into a set of *m* available ULDs.

2.2 Parameters

The following parameters are known about the set of boxes: the number of boxes to be packed, the dimensions of each box, whether or not the boxes can be rotated by $\pi/2$, are fragile, perishable, and/or radioactive. The following parameters are known about the set of available ULDs: the dimensions and cost of each ULD. All of these values are assumed to be integers and defined as:

n	Total number of boxes to be packed,	
m	Total number of available ULDs,	
l_i x h_i	Length x height of box *i*,	$\forall i,$
v_i	Rotatability of box *i*,	$\forall i,$
f_i	Fragility of box *i*,	$\forall i,$
p_i	Perishability of box *i*,	$\forall i,$
r_i	Radioactivity of box *i*,	$\forall i,$
L_j x H_j	Length x height of container *j*,	$\forall j,$
C_j	Cost of container *j*,	$\forall j,$

$i \in \{1, \dots, n\}$, $j \in \{1, \dots, m\}$.

The following described parameters are binary and equal to one when the described case is true and zero when false. The rotatability parameter states box *i* is allowed to be rotated by $\pi/2$, the fragility parameter states box *i* is fragile meaning it cannot have any other boxes on top of it, the perishability parameter states box *i* is perishable and the radioactivity parameter states box *i* is radioactive. If an item is perishable it cannot be in the same container *j* as a radioactive item and vice versa.

Additional parameters that are required in the description of constraints are introduced as follows:

L Maximum length of all containers, max L_j ,

H Maximum height of all containers, max H_j ,

$j \in \{1, \dots, m\}$.

The parameters *L* and *H* are introduced for use in the definition of various constraints that use the big M method described in section 2.5.

2.3 Decision Variables

The coordinate system used to express the following variables considers the length L_j of the container *j* to lie on the x-axis and the height H_j to lie on the z-axis $\forall\, j \in \{1,\dots, m\}$. The origin lies on the bottom left corner of the containers.

$$p_{ij} = \begin{cases} 1 & \text{if box } i \text{ is in container } j, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i, j,$$

$$u_j = \begin{cases} 1 & \text{if container } j \text{ is used,} \\ 0 & \text{otherwise,} \end{cases} \qquad \forall j,$$

(x_i , z_i) location of the bottom left corner of box *i*, $\forall i,$

(x'_i , z'_i) location of the top right corner of box *I*, $\forall i,$

$$r_{iab} = \begin{cases} 1 & \text{if the side } b \text{ of box } i \text{ is along the } a\text{-axis,} \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i,$$

$$x^p_{ik} = \begin{cases} 1 & \text{if box } i \text{ is on the right of box } k (x'_k \leq x_i), \\ 0 & \text{otherwise} (x_i < x'_k), \end{cases} \qquad \forall i,$$

$$z^p_{ik} = \begin{cases} 1 & \text{if box } i \text{ is above } k (z'_k \leq z_i), \\ 0 & \text{otherwise } (z_i < z'_k), \end{cases} \qquad \forall i,$$

$$i, k \in \{1, \dots, n\},\ j \in \{1, \dots, m\},\ \ a, b \in \{1,\ 3\}.$$

The variables (x_i ,z_i) and (x'_i , z'_i) are assumed to be integers describing the position of box *i* inside the container. Considering some boxes are allowed to rotate by $\pi/2$, the variables r_{iab} are used to describe the orientation of box *i* inside the container. The index *a* indicates the axis, $a \in \{x := 1, z := 3\}$ and *b* indicated the side of the box, $b \in \{l := 1, h := 3\}$. These variables specify which side of box *i* is along which axis.

The variables x^p_{ik} (z^p_{ik}) are introduced to prevent overlap of boxes. They represent the relative position of two boxes, being equal to 1 if the box *i* is on the right side (or above) of box *k*. These two variables are defined later on by constraints but only partly, for example if $x^p_{ik} = 1$, then we know that $x'_k \leq x_i$, however, if $x^p_{ik} = 0$, we have no information.

For use in vertical stability, fragility, and compatibility constraint definitions, the following variables are introduced:

$$g_i = \begin{cases} 1 & \text{if box } i \text{ is on the ground } (z_i = 0), \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i,$$

$$h_{ik} = \begin{cases} 0 & \text{if box } k \text{ has the suitable height to support box } i \ (z_i = z'_k), \\ 1 & \text{otherwise,} \end{cases} \qquad \forall i, k,$$

$$o_{ik} = \begin{cases} 0 & \text{if the projections on the } XY \text{ plane of the boxes} \\ & i \text{ and } k \text{ have a nonempty intersection,} \\ 1 & \text{otherwise,} \end{cases} \qquad \forall i, k,$$

$$s_{ik} = \begin{cases} 1 & \text{if box } k \text{ supports box } i \text{ and are in the same bin,} \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i, k,$$

$$\eta^1_{ik} = \begin{cases} 0 & \text{if } x_k \leq x_i, \\ 1 & \text{otherwise,} \end{cases} \qquad \forall i, k,$$

$$\eta^3_{ik} = \begin{cases} 0 & \text{if } x'_i \leq x'_k, \\ 1 & \text{otherwise,} \end{cases} \qquad \forall i, k,$$

$$\beta^l_{ik} = \begin{cases} 1 & \text{if the vertex } l \text{ is supported by box } k, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i, k, l,$$

$$v_{ik} = \left| z_k' - z_i \right| \qquad \forall i, k,$$

$$m_{ik} = \begin{cases} 1 & \text{if } z_k' \geq z_i, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i, k,$$

$$c_j = \begin{cases} 1 & \text{if container } j \text{ is compatible with radioactive items,} \\ 0 & \text{otherwise,} \end{cases} \qquad \forall j,$$

$i, k \in \{1, \dots, n\},\ l \in \{1, 2\}$.

Variables s_{ik} and o_{ik} are useful for the definition of fragility constraints in section 2.5.4.

2.4 Objective

The objective function consists of packing all items $\in n$ while minimizing the cost of the used ULDs $\subseteq m$. This function is mathematically described as:

$$\sum_{j=1}^{m} u_j C_j \qquad \forall j$$

2.5 Constraints

The constraints of the model are split into geometric, orientation, vertical stability, fragility, and compatibility constraints.

2.5.1 Geometric

Geometric constraints ensure each box lies within only one ULD and does not overlap with other boxes.

$$\sum_{j=0}^{m} p_{ij} = 1 \qquad \forall i \qquad (1)$$

Box i is constrained to a single bin *j*.

$$\sum_{j=0}^{m} l_i p_{ij} \geq x'_i \qquad \forall i \qquad (2)$$

The horizontal position of the right side of the box *i* is constrained by the length of the bin *j*.

$$\sum_{j=0}^{m} h_i p_{ij} \geq z'_i \qquad \forall i \qquad (3)$$

The vertical position of the upper side of the box *i* is constrained by the height of the bin *j*.

$$x'_i - x_i = r_{i00} l_i + r_{i01} h_i \qquad \forall i \qquad (4)$$

The length of item *i* is either the length or height of item *i*.

$$z'_i - z_i = r_{i10} l_i + r_{i11} h_i \qquad \forall i \qquad (5)$$

The height of item *i* is either the length or height of item *i*.

$$\sum_{a=0}^{1} r_{iab} \qquad \forall i, b \qquad (6)$$

The side *b* of box *i* can only be along one axis *a*.

$$\sum_{b=0}^{1} r_{iab} \qquad \forall i, a \qquad (7)$$

The axis *a* can only be along 1 side of the box *i*.

$$x^p_{ik} + x^p_{ki} + z^p_{ik} + z^p_{ki} \geq (p_{ij} + p_{kj}) - 1 \qquad \forall i, k, j \qquad (8)$$

Box *i* and *k* cannot overlap if they’re placed in the same bin.

$$x'_k \leq x_i + (1 - x^p_{ik})L \qquad \forall i, k \tag{9}$$

If box *i* is to the right of box *k* then the left vertex of *i* must be greater or equal to the right vertex of box *k*. When $x^p_{ik} = 1$ and the positions of boxes *i* and *k* satisfy the inequality, the value of x^p_{ik} is reinforced. Dummy constraint imposed when $x^p_{ik} = 0$ (always true).

$$x_i + 1 \leq x'_k + x^p_{ik}L \qquad \forall i, k \tag{10}$$

If box *i* is not to the right of box *k* then the left vertex of *i* must be less or equal to the right vertex of box *k*. When $x^p_{ik} = 0$ and the positions of boxes *i* and *k* satisfy the inequality, the value of x^p_{ik} is reinforced. Dummy constraint imposed when $x^p_{ik} = 1$ (always true).

$$z'_k \leq z_i + (1 - z^p_{ik})H \qquad \forall i, k \tag{11}$$

If box *i* is above box *k* then the lower vertex of *i* must be greater or equal to the upper vertex of box *k*. When $z^p_{ik} = 1$ and the positions of boxes *i* and *k* satisfy the inequality, the value of z^p_{ik} is reinforced. Dummy constraint imposed when $z^p_{ik} = 0$ (always true).

$$z_i + 1 \leq z'_k + z^p_{ik}H \qquad \forall i, k \tag{12}$$

If box *i* is not above box *k* then the lower vertex of *i* must be less or equal to the upper vertex of box *k*. When $z^p_{ik} = 0$ and the positions of boxes *i* and *k* satisfy the inequality, the value of z^p_{ik} is reinforced. Dummy constraint imposed when $z^p_{ik} = 1$ (always true).

2.5.2 Orientation

Orientation constraints are needed to ensure that only boxes which are allowed to be rotated by $\pi/2$ can be rotated by the model.

$$r_{i31} \leq v_i \qquad \forall i \tag{13}$$

The length of the box *i* can only be along the axis *z* if and only if box *i* is allowed to rotate. If the parameter v_i is set to one the box can rotate, if it is set to zero the box is not allowed to rotate.

2.5.3 Vertical Stability

Vertical stability constraints ensure that the bottom of each box is supported by the top of another box or by the container floor to ensure that there are no floating boxes. To do this, we need to determine whether a box is on the ground and whether its bottom vertices are supported correctly. A correctly supported lower vertex of box *i* relies on there being another box *k* that has a suitable height and the two box's projections having an overlap on the *x*-axis.

$$\sum_{l=1}^{2}\sum_{k=1}^{n} \beta_{i,k}^{l} + 2g_i = 2 \qquad \forall i \in n \tag{14}$$

The vertices one (bottom-left) and two (bottom-right) are both supported by single or two distinct boxes when the first term of the left-hand side is equal to two. Both vertices are supported by the ground when the second term of the left-hand side is equal to two.

$$z_i \le (1 - g_i)H \qquad \forall i \tag{15}$$

Box *i* is on the container floor when g_i equals one.

$$z'_k - z_i \le v_{ik} \qquad \forall i, k \tag{16}$$

$$z_i - z'_k \le v_{ik} \qquad \forall i, k \tag{17}$$

$$v_{ik} \le z'_k - z_i + 2H(1 - m_{ik}) \qquad \forall i, k \tag{18}$$

$$v_{ik} \le z_i - z'_k + 2Hm_{ik} \qquad \forall i, k \tag{19}$$

Variable v_{ik} is defined as the absolute value $\left|z'_k - z_i\right|$ using variable m_{ik} .

$$h_{ik} \le v_{ik} \qquad \forall i, k \tag{20}$$

$$v_{ik} \le h_{ik}H \qquad \forall i, k \tag{21}$$

Variable h_{ik} is defined using variable v_{ik} . If the gap between the top of box *k* and the bottom of box *i* is zero, then box *k* has a suitable height to support box *i*.

$$o_{ik} \le x_{ik}^{p} + x_{ki}^{p} \le o_{ik} \qquad \forall i, k \tag{22}$$

Boxes *i* and *k* share a part of their orthogonal projection on the *x*-axis if $x_{ik}^{p} + x_{ki}^{p} = 0.$

$$(1 - s_{ik}) \le h_{ik} + o_{ik} \le 2(1 - s_{ik}) \qquad \forall i, k \tag{23}$$

If box *k* has a suitable height to support box *i* and their orthogonal projections on the *x*-axis overlap, then box *k* supports box *i*.

$$p_{ij} - p_{kj} \le 1 - s_{ik} \qquad \forall i, j, k \tag{24}$$

$$p_{kj} - p_{ij} \le 1 - s_{ik} \qquad \forall i, j, k \tag{25}$$

Box *i* can only be supported by box *k* if they are in the same container *j*.

$$\beta_{ik}^{l} \leq s_{ik} \qquad \forall i, k, l \qquad (26)$$

Vertex *l* of box *i* is supported by box *k* only if box *i* is supported by box *k* and they are in the same container.

$$\eta_{ik}^{1} \leq 2(1 - \beta_{ik}^{1}) \qquad \forall i, k \qquad (27)$$

Vertex 1 of box *i* can only be supported by box *k* if $x_k \leq x_i$.

$$\eta_{ik}^{3} \leq 2(1 - \beta_{ik}^{2}) \qquad \forall i, k \qquad (28)$$

Vertex 2 of box *i* can only be supported by box *k* if $x'_i \leq x'_k$.

$$x_k \leq x_i + \eta_{ik}^{1} L \qquad \forall i, k \qquad (29)$$
$$x'_i \leq x'_k + \eta_{ik}^{3} L \qquad \forall i, k \qquad (30)$$

Variables η_{ik}^{1} and η_{ik}^{3} are defined. These constraints are similar to constraint (10) above.
$i, k \in \{1, \dots, n\}$, $j \in \{1, \dots, m\}$, $l \in \{1, 2\}$.

2.5.4 Fragility

Some boxes are fragile due to the nature of their contents, meaning that they cannot support boxes on top of them.

$$\sum_i s_{ik} \leq n(1 - f_k) \qquad \forall k \in \{1, \dots, n\}. \qquad (31)$$

The left-hand side of the inequality represents the number of boxes supported by box *k*. If box *k* is fragile, this must be zero.

2.5.5 Compatibility

Compatibility constraints include defining whether or not a specific container is used and not allowing certain boxes to be in the same container as each other. Some boxes are perishable and some are radioactive, these two types of boxes must be in separate containers.

$$p_{ij} - u_j \leq 0 \qquad \forall i, j \qquad (32)$$

Container *j* is used as soon as any box *i* is placed inside of it.

$$r_i p_{ij} \leq n c_j \qquad \forall i, j \qquad (33)$$

If there exist no radioactive boxes in bin j then there is no restriction on what items can be placed in the bin.

$$p_i p_{ij} \leq n(1 - c_j) \qquad \forall i, j \tag{34}$$

If there exists a radioactive box in bin j then there cannot be any perishable items in the bin.

3. Packing Results

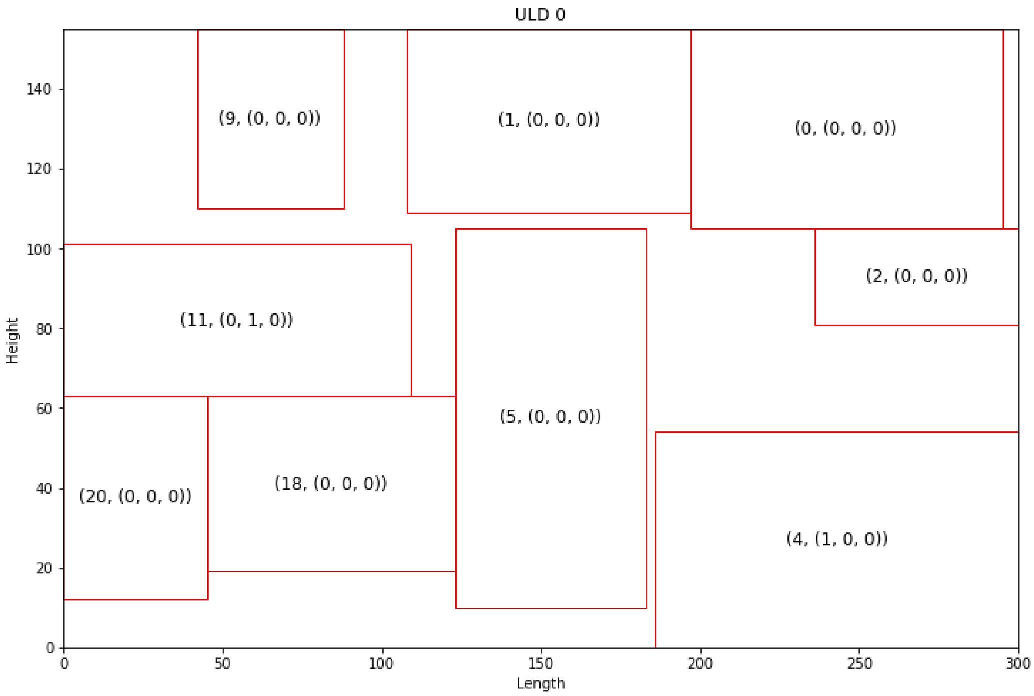

Figure 1: resulting packing strategy for Unit Load Device (ULD) 0.

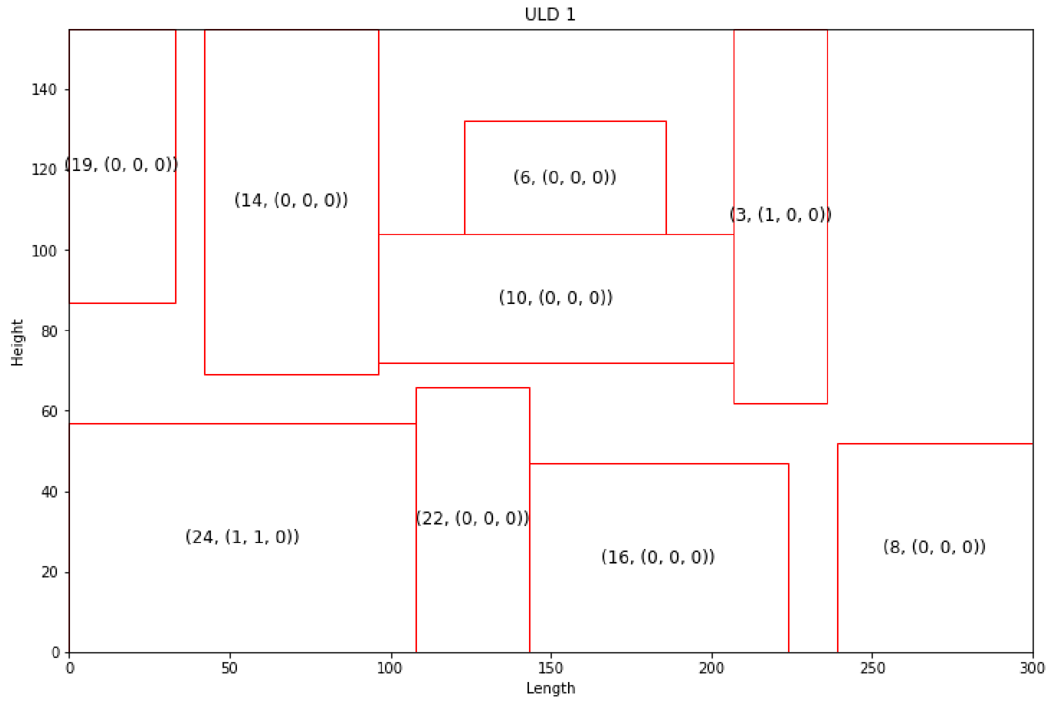

Figure 2: resulting packing strategy for Unit Load Device (ULD) 1.

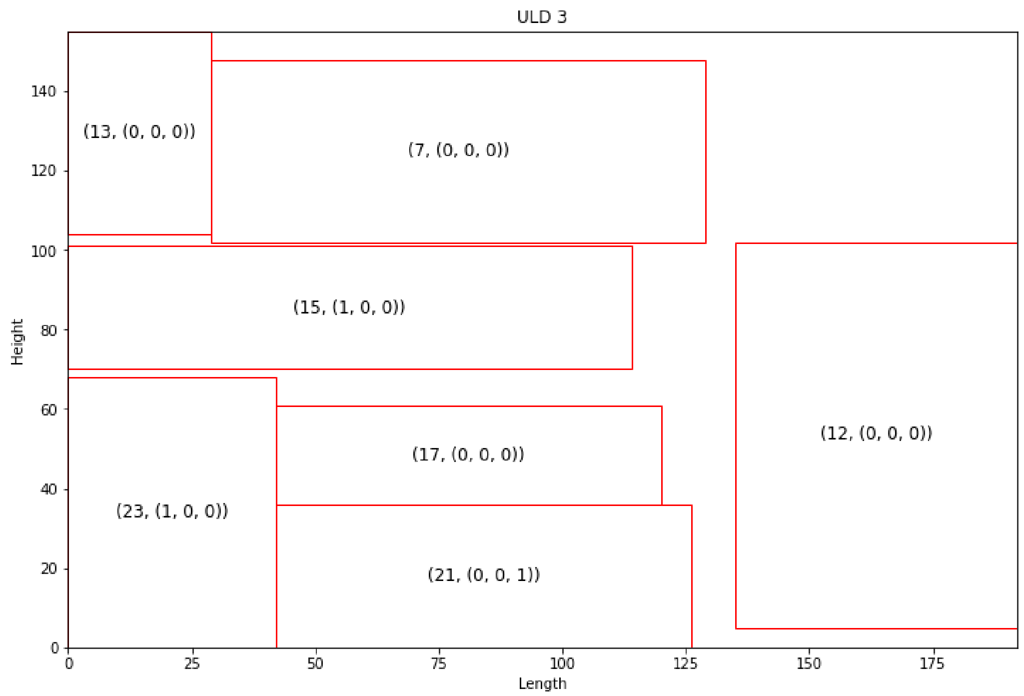

Figure 3: resulting packing strategy for Unit Load Device (ULD) 2.

Based on the observation of Figure 1-3, all geometric constraints have been satisfied. All boxes are confined to the dimensions of their respective bins, no bins overlap each other, and all boxes are constrained to a single bin. The compatibility constraint between perishable and radioactive boxes has been satisfied, hence bin 2 does not contain any perishable boxes. All bins which contain boxes have been declared used by the decision variable u_j. The fragility constraint is unsatisfied as boxes have been placed on top of fragile items. Not all stability constraints have been satisfied. The constraint that a minimum of two vertices must be supported by box k unless supported by the ground is not satisfied as there exist unsupported lower vertices. In addition, not all boxes have suitable heights to ensure vertical stability. The orientation constraint has not been satisfied, hence box 22 was rotated to a vertical position in bin 1.

```
[0, 1, 3]
```

Figure 4: the Unit Load Devices (ULDs) used.

```
{0: [0, 1, 2, 4, 5, 9, 11, 18, 20], 1: [3, 6, 8, 10, 14, 16, 19, 22, 24], 3: [7, 12, 13, 15, 17, 21, 23]}
```

Figure 5: Dictionary mapping for each used bin, the items that were allocated to that bin.

```
{0: [197.0, 105.0, 98.0, 50.0], 1: [108.0, 109.0, 89.0, 46.0], 2: [236.0, 81.0, 64.0, 24.0], 3: [207.0, 62.0, 29.0, 93.0], 4: [186.0, 0.0,
114.0, 54.0], 5: [123.0, 10.0, 60.0, 95.0], 6: [123.0, 104.0, 63.0, 28.0], 7: [29.0, 101.99999999999994, 100.0, 46.0], 8: [239.0, 0.0, 61.
0, 52.0], 9: [42.0, 110.0, 46.0, 45.0], 10: [96.0, 72.0, 111.0, 32.0], 11: [0.0, 63.0, 109.0, 38.0], 12: [135.0, 4.999999999999943, 57.0,
97.0], 13: [0.0, 104.0, 29.0, 51.0], 14: [42.0, 69.0, 54.0, 86.0], 15: [0.0, 69.99999999999994, 114.0, 31.0], 16: [143.0, 0.0, 81.0, 47.
0], 17: [42.0, 36.0, 78.0, 25.0], 18: [45.0, 19.0, 78.0, 44.0], 19: [0.0, 87.0, 33.0, 68.0], 20: [0.0, 12.0, 45.0, 51.0], 21: [41.99999999
999999, 4.440892098500626e-15, 84.0, 36.0], 22: [108.0, 0.0, 35.0, 66.0], 23: [0.0, 0.0, 42.0, 68.0], 24: [0.0, 0.0, 108.0, 57.0]}
```

Figure 6: Dictionary mapping for each box the horizontal coordinate and vertical coordinate of the lower-left vertex, and the horizontal and vertical extension of the box.

4. Algorithmic Results

The solution is not optimal, after reaching the set two-hour time limit the gap optimality was 41.7% with the best bound being 280 and the best incumbent being 480. This means that with some constraints being relaxed, the solution was able to fit all items in containers 0 and 1, which were both of type 0 and cost 140 each. With all constraints being enforced, the only feasible solution required the use of three containers, the two type 0 containers with one type 1 container.

Even with setting a higher time limit and allowing the model to explore more nodes, it would not be geometrically possible to put all items into only two containers due to the total area of the boxes being greater than the total area of the containers. This would require the overlapping constraint or some other geometric constraint to be relaxed.

It is not feasible to package a number of the boxes hence the center of gravity of these bins is not vertically supported meaning they would fall if placed in that position. Other boxes are supported by their center of gravity however their vertices are not. The reduced area supporting these boxes means they are subject to higher load, which may lead to damage or deformation.

5. Managerial Considerations

Apart from the obvious two-dimensional simplification, there are numerous additional assumptions that limit the model's ability to be implemented in real-world cargo scenarios. This model is simplified in many regards to an already simplified three-dimensional model.

Weight restrictions are ignored to simplify the model, however, removing weight constraints severely limits the model's ability to be implemented. Weight limits and distribution are major limiting factors in aircraft cargo operations, having impacts on safety and overall feasibility.

The model assumes that all the items are rigid and can be packaged tightly against their neighboring surfaces. In reality, packages do not have perfectly flat edges and therefore may not fit as intended by the model. This may result in larger loads on the items, which may cause deformation or damage. The ULDs considered in the model are also perfectly rectangular without the cuts that are common in reality.

The model does not account for the horizontal stability of items. The movement of the ULDs results in horizontal forces that may topple items over inside the ULD if the packing strategy is only accounting for vertical stability. This would be problematic, especially for fragile items.

The model can only be optimized once a given set of items and available containers are known. In reality, air cargo companies experience shipment cancellations and/or last-minute shipments. In such events, these items would not be accounted for unless the model is updated and re-optimized, which may be infeasible due to too long computation times.

The model does not consider the end destination of the items being packed. In reality, different items have different destinations and should be packed accordingly. Ideally, every item inside of one ULD should have the same destination so that they do not have to be repacked at every stopover that a cargo aircraft has.

The model only considers one objective which is minimizing the cost of the used containers, however, this may not be the only or most important objective in air cargo packing situations. Other important objectives may be to maximize the number of items packed, maximize the relative value of items packed, or maximize the safe transport of items. In reality, the model must also adhere to all relevant regulatory requirements either imposed by authorities or the relevant companies.

6. Personal Reflection

Zach:

This assignment applies the formulations discussed in lectures to a practical problem that relates to air cargo operations and general optimization concepts. I value that the solution is relatable and can be visually interpreted. However, it can be challenging to debug your code even if you understand the concepts. In the process of developing the mixed integer program formulation, I learned to simplify the problem as much as possible while keeping enough complexity when trying to detect the reasons for the error(s) in the solution.

In further iterations of the assignment, I recommend limiting the number of decision variables and constraints as debugging is challenging to trace due to the complexity of the model. I'd suggest adding optimization exercises in some lectures as this course does not prepare you to program optimization algorithms.

Timotei:

Overall this assignment was challenging and I learned a lot from it. I have no background in optimization or modeling in the way that was required by this assignment, that could be due to the fact that I am an exchange student. The best part of this assignment was the opportunity to develop these skills and apply them to an aerospace-related application. As mentioned multiple times in lectures, this bin packing problem and a lot of other problems related to airport and cargo operations are very complex and this assignment definitely confirmed that for me. Even this hugely simplified problem was very difficult. The worst part of the assignment was debugging. It was very difficult to find the reason for errors in the code and this took up most of the time spent on the project. I learned that using smaller toy cases and working on constraints one-by-one is the only way to code such a model.

If I were to redesign the assignment I would drop an additional constraint to further simplify the problem. Coding and debugging the main constraints of overlap, stability, and fragility is already a challenge. I would add the possibility to explore other optimization techniques and more coded examples either in class or provided as examples. I would keep the simplifications already given and the reference paper. Lastly, I would improve the group formation process by assigning groups. It is difficult to find a group, especially as an exchange student, and a smaller group significantly adds to the work required of each group member.