1 General description

In this assignment, you will play the role of a team of air cargo analysts, part of the Operations department, working on a packing problem.

A colleague from the Revenue Management department has just given you the set of items $\mathcal{I}$ that were accepted to be transported in the belly space of a passenger aircraft that is departing tomorrow. Given the aircraft type and expected luggage demand, you know the remaining available volume for cargo and, as a consequence, the set $\mathcal{B}$ of available ULDs that can be loaded in the belly space.

Your goal is to model and optimize a packing strategy such that all items $\in \mathcal{I}$ are loaded while minimizing the cost of the used ULDs $\subseteq \mathcal{B}$. While the problem is formally a fully 3D Bin Packing Problem (3DBPP), in the context of this assignment a simplified 2D Bin Packing Problem (2DBPP) will be solved. More specifically, we will neglect the depth of both items and ULDs and will focus on a vertical 2DBPP where a horizontal and vertical direction are considered.

To solve the problem, the following paper should be used as a reference: `https://onlinelibrary.wiley.com/doi/full/10.1111/itor.12111`. As this paper deals with a 3DBPP, you need to neglect all y decision variables (out-of-plane position variables) and focus only on x and z variables when defining the position of an item in a ULD. In a similar fashion, items can (potentially) only be rotated by $\frac{\pi}{2}$, as this is the only rotation allowed in the x-z plane.

Apart from the constraints that are inherited directly from a "horizontal" 2DBPP, such as non-overlapping between items, horizontal and vertical bounds of the ULD, assignment of each item to a single ULD, there are some additional constraints you need to consider due to the "vertical" setting. In particular

1. items cannot float in the ULD. Both lower corners of each item must be supported. This can be done either by the ground, by another item, or by the lower-left cut in case the item is assigned to an LD3 container (see Fig. 1)

Figure 1: Example of LD3 container. From `https://www.air-cargo-products.de/ld3-container.html`.

2. some items might not be turned by $\frac{\pi}{2}$ as they contain objects that might get damaged otherwise
3. some items might be fragile and, as such, no other box can be stacked on top of them

In addition, we will consider two specific types of items, namely *perishable* and *radioactive* items. Because of incompatibility constraints, a ULD containing perishable items cannot accommodate any radioactive items (and vice versa).

2 Input data

Your group will receive two main inputs, namely the set of ULDs $\mathcal{B}$ and the set of items $\mathcal{I}$, stored as Python **pickle** files. Both data structures are dictionaries. The set of ULDs $\mathcal{B}$, once loaded, looks like this

```
B
Out[7]:
{0: (0, [200, 150, 2, 100, 0, 0, 0]),
 1: (0, [200, 150, 2, 100, 0, 0, 0]),
 2: (1, [180, 120, 1, 130, 1, -5, 40])}
```

where the key represents the index of the bin, the first number in the values is the bin type, and the list with seven values contains the following information: the length of the bin, the height of the bin, how many bins of this type are available (not really needed as the same information can be retrieved counting the number of bins characterized by the same bin type), the cost of the bin, whether the bin has a cut (1) or not (0) and, in the former case, the slope and intercept representing the cut. Using the example above, we have 2 bins of type 0. Each of them, if used, costs 100 and they do not have a cut. We do have a single bin of type 1 available, which is more costly (130) and has a cut instead. The set of items $\mathcal{I}$, once loaded, looks like this

```
I
Out[17]:
{0: (98, 50, 1, 0, 0, 0),
 1: (89, 46, 1, 0, 0, 0),
 2: (64, 24, 1, 0, 0, 0),
 3: (93, 29, 1, 1, 0, 0),
 4: (114, 54, 1, 1, 0, 0),
 5: (95, 60, 1, 0, 0, 0),
 6: (63, 28, 1, 0, 0, 0),
 7: (100, 46, 0, 0, 0, 0),
 8: (52, 61, 1, 0, 0, 0),
 9: (45, 46, 1, 0, 0, 0),
 10: (111, 32, 1, 0, 0, 0),
 11: (109, 38, 1, 0, 1, 0),
 12: (97, 57, 1, 0, 0, 0),
 13: (51, 29, 1, 0, 0, 0),
 14: (86, 54, 1, 0, 0, 0),
 15: (114, 31, 1, 1, 0, 0),
 16: (81, 47, 1, 0, 0, 0),
 17: (78, 25, 1, 0, 0, 0),
 18: (78, 44, 1, 0, 0, 0),
 19: (68, 33, 1, 0, 0, 0),
 20: (51, 45, 1, 0, 0, 0),
 21: (84, 36, 1, 0, 0, 1),
 22: (66, 35, 0, 0, 0, 0),
 23: (68, 42, 1, 1, 0, 0),
 24: (108, 57, 1, 1, 1, 0)}
```

where each key is the index of the associated item, and the six values are, in sequence, the original length of the item, the original height of the item, a binary indicator that is unitary if the item can be rotated by $\frac{\pi}{2}$ and zero otherwise, a binary indicator that is unitary if the item is fragile and zero otherwise, a binary indicator that is unitary if the item is perishable and zero otherwise, and a binary indicator that is unitary if the item is radioactive and zero otherwise. For example, item 24 has a nominal length of 108, a nominal height of 57, can be rotated by $\frac{\pi}{2}$, is fragile, is perishable, but not radioactive.

To load pickle files, you need to import the pickle package into your code

```
import pickle
```

and, assuming you pickle file is **B.pickle**, use the following expression

```
with open('B.pickle', 'rb') as handle:
    B = pickle.load(handle)
```

which will return **B** (in your case, a dictionary).

3 Mathematical Model

As previously introduced, the 3DBPP described in reference https://onlinelibrary.wiley.com/doi/full/10.1111/itor.12111 should be modified so that only the x-z plane is considered. Most constraints remain unchanged, with a few of them needing minor tweaking.

You should first define the 2DBPP model needed, then code it (my suggestion is in **Python** using the **gurobipy** package that interfaces with the **Gurobi** commercial solver), and let Gurobi solve it via **branch and bound**. As the problem might be hard to solve to optimality, use a time limit of **two hours**. Make sure to save the **.log** file with the command

```
model.params.LogFile='name_model.log'
```

so that you can report **best incumbent**, **best bound**, **gap optimality**, and **computational time** in your analysis.

4 Extra simplifications with respect to the original model

On top of a vertical 2DBPP rather that a fully 3D case, the following additional simplifications are added to the model

- no weight restrictions are present. As no weight capacity is defined per bin type and no weight is defined per item, it is assumed that weight is not a limiting factor in the packing strategy
- two bin types are defined. One represents a frontal view of an LD7 pallet, and the other one represents a frontal view of an LD3 container. In the latter case, we simplify the problem by assuming no cut is present. Hence, both bin types are rectangles. This also means you do not have to define γ_i decision variables, as no cut is present to support items. Additionally, the stability constraint for every item is simplified as follows

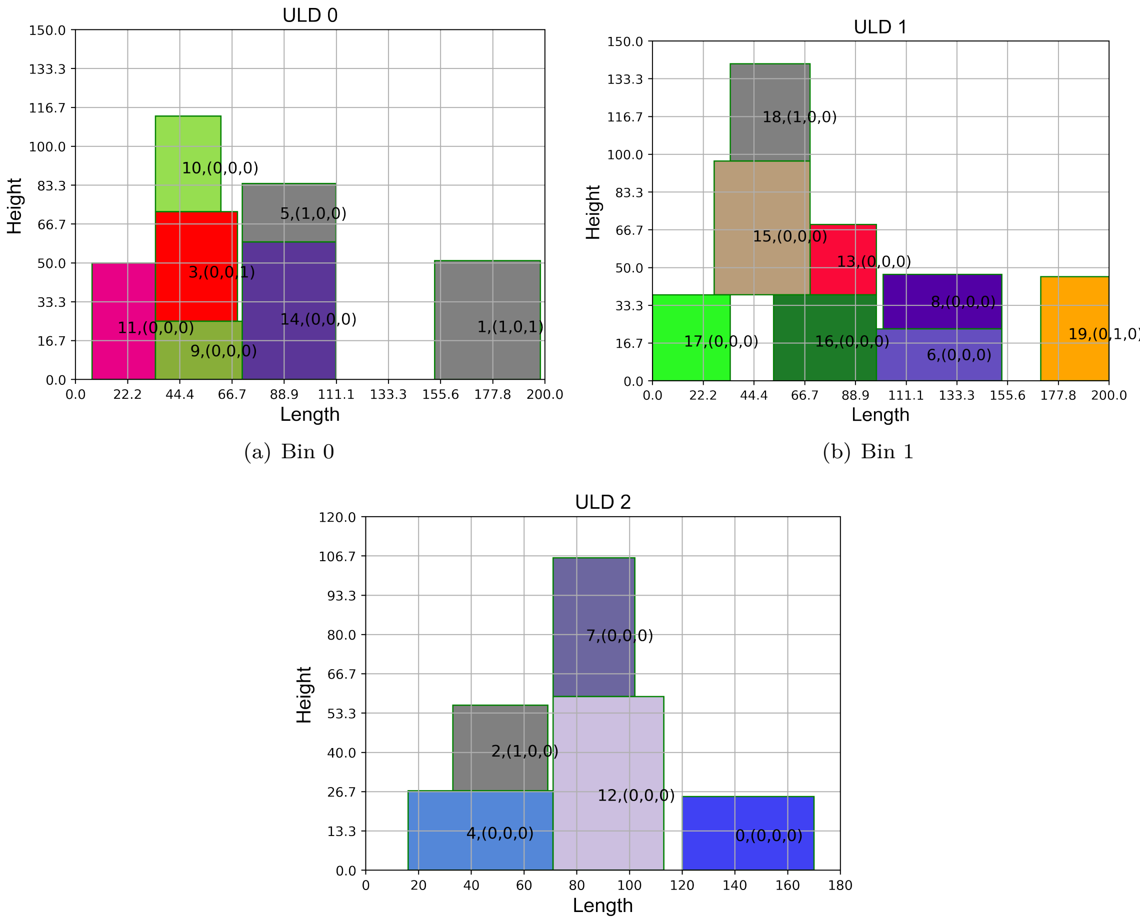
$$\sum_{k=1}^{2} \sum_{j \in \mathcal{I}} \beta_{i,j}^{k} + 2g_i = 2 \qquad \forall i \in \mathcal{I}$$

  because an item can be supported either by the ground (second term of the left hand side) or by a single or two distinct other boxes (first term of the left hand side)

5 Presentation of results

When presenting your results, use figures that are easy to interpret. A good (yet not perfect) example is shown in Fig. 2. Note that all constraints stemming from the problem are satisfied, such as stability (all items have both lower vertices supported), fragility (fragile items are either on top of stacks or on the ground with nothing on top), incompatibility between perishables and radioactive items.
On top of visualizing your results, you should be providing them as **pickle** files with the following three data structures

```
bins_used
Out[10]: [0, 1, 2]
```



(a) Bin 0 (b) Bin 1

(c) Bin 2

Figure 2: Example of output of packing strategy. Fragile items represented in gray. Numbers inside each item represent its original index, and then whether the item is fragile, perishable, and radioactive. For example, sequence 1,(1,0,1) implies that item 1 is fragile, not perishable, and radioactive.

List mapping the (sub)set of bins $\subseteq \mathcal{B}$ that are used in the solution

```
Items_in_Bin
Out[11]:
{0: [1, 3, 5, 9, 10, 11, 14],
 1: [6, 8, 13, 15, 16, 17, 18, 19],
 2: [0, 2, 4, 7, 12]}
```

Dictionary mapping, for each used bin, the items that were allocated to that bin by the 2DBPP

```
I_info_solution
Out[12]:
{0: [120.0, 0.0, 50, 25],
 1: [153.00000000000006, 0.0, 45, 51],
 2: [33.00000000000006, 27.0, 36, 29],
 3: [33.99999999999997, 25.0, 35, 47],
 4: [15.999999999999972, 0.0, 55, 27],
 5: [70.99999999999997, 59.0, 40, 25],
 6: [98.00000000000006, 0.0, 55, 23],
 7: [70.99999999999997, 59.0, 31, 47],
```

```
8: [101.00000000000006, 23.0, 52, 24],
9: [33.9999999999997, 0.0, 37, 25],
10: [33.99999999999997, 72.0, 28, 41],
11: [6.999999999999972, 0.0, 27, 50],
12: [70.99999999999997, 0.0, 42, 59],
13: [69.00000000000006, 38.0, 29, 31],
14: [70.99999999999997, 0.0, 40, 59],
15: [27.000000000000057, 38.0, 42, 59],
16: [53.00000000000006, 0.0, 45, 38],
17: [0.0, 0.0, 34, 38],
18: [34.00000000000003, 97.0, 35, 43],
19: [170.0, 0.0, 30, 46]}
```

Dictionary mapping, for each item $i \in \mathcal{I}$, the horizontal coordinate x_i and vertical coordinate z_i of the lower-left vertex, and the horizontal and vertical extension of the item (recall the horizontal extension could be the original height if the item was rotated and vice versa)

6 Structure of report

You should describe your model, results, and critical analysis in a report of maximum twelve pages (using fontsize and layout similar to this guideline). The structure of the report should be

1. Introduction: briefly introduce the problem and the motivation
2. Mathematical model: describe the 2DBPP using the standard notation. Introduce the sets, parameters, and decision variables. Then show the MILP and then describe (briefly) the objective and constraints
3. Packing results: describe the obtained packing strategy using visualizations and a brief critical analysis. Are results expected? Are all major constraints satisfied?
4. Algorithmic results: briefly describe the solution quality. Is the solution optimal? If not, what is the gap optimality? What are your insights in this regard? Can you prove with simple considerations based on geometry or other properties that the solution is optimal or quasi-optimal, regardless of the output of the branch and bound solver?
5. Managerial considerations: considering that every model is an approximation of reality, what is your opinion on the modeled 2DBPP? Apart from the obvious simplification that we are neglecting the third dimension, are there some other simplifications/assumptions that make the model not fully implementable in a real air cargo case study in your opinion? If so, can you define them?
6. Personal reflection: each member of the group should provide a brief self-reflection (two paragraphs maximum), describing the best and worst part of the assignment, together with the lessons learned in the process. In addition, what would you Drop, Add, Keep, and Improve (DAKI) if you were to re-design the assignment next year?

7 Grading

The grading of the assignment will be carried out using the following criterion

- 20%: general readability and content of the report
- 20%: correctness of 2DBPP model
- 30%: correctness of final packing solution
- 10%: quality of description of results (packing and algorithmic)
- 10%: quality of managerial insights description

- 10%: quality of personal reflection

Good luck with the assignment!