A partial study on container loading methods for a university's relocation

Pedro H. Centenaro[1], Luiz-Rafael Santos[2], Luiz F. Bossa[3]
UFSC, Blumenau, SC

The Federal University of Santa Catarina (UFSC) is one of the oldest higher-education institutions in the Brazilian south, with more than six decades of history. The university expanded to Blumenau in 2014, where it offers majors in chemistry, mathematics and three different engineering fields. Due to the relatively small size of the campus, and due to high interest in offering new majors, the university will move to a new campus with twice the area of the original, and lower rent, between the semesters of 2024. Given that this will involve moving classroom furniture, office resources and laboratory paraphernalia, cutting travel expenses is important. Hence, we are currently studying container loading problems (CLPs) to optimize the volume of cargo per container and minimize the number of travels (containers) needed for the university's relocation. Presently, we consider the following sets of boxes and containers.

Table 1: List of boxes for packing (measured in cm).

Type	Width	Height	Length	Stock
Small	32	27	43	124
Medium	50	40	40	122
Large	40	60	50	1383

Table 2: List of containers (measured in cm).

Type	Width	Height	Length
A	220	350	720
B	260	440	1000
C	260	440	1400

Since the sets of boxes and containers are small and of similar size, this CLP is categorized as a multiple stock-size cutting stock problem (MSSCSP) [2]. Although there are exact models that can optimally solve this problem [4], they cannot do so in an adequate amount of time. Therefore, we resort to heuristics in the hopes of obtaining box placements of comparable quality. At the moment, we are not taking into account constraints on weight, permissible box orientations, which boxes can be stacked on top of each other, and more. For this reason, the simple wall-building heuristic by George and Robinson [5] suffices. This heuristic constructively builds a solution by slicing the container into rectangular layers and applying a packing procedure to each layer. To determine a new layer's depth, we first check if there is any box type with stock left to pack that

[1] pedro.centenaro@grad.ufsc.br
[2] l.r.santos@ufsc.br
[3] l.f.bossa@ufsc.br

has already been placed in the container. If so, we choose the box type with the greatest remaining stock. Otherwise, we filter boxes through a series of rankings and then choose one of the types that remain. The rankings are, in order: (i) keep boxes with maximum smallest dimension; (ii) keep boxes with maximum remaining stock; (iii) keep boxes with maximum largest dimension. We then check which of the selected box's dimensions fits the remaining depth of the container and select the largest such dimension as the layer's depth. The initial layer space is then filled with boxes of the selected type, and the heuristic tries to fill any remaining spaces with other box types. When filling a space, we prioritize stacking columns of boxes side by side. When this is possible, we rotate boxes to fill as much vertical space as possible; otherwise, rotations that occupy more horizontal space are chosen. It is also possible to amalgamate the space of the current layer with the previous layer's adjacent unfilled spaces, so as to increase the odds of making space for large objects.

Though the heuristic by George and Robinson [5] was only considered for a single container, we made small adaptations so that our implementation (which can be found at `https://github.com/phcentenaro7/IC-CLP`) could fill multiple containers. The heuristic was coded in Julia [1], with the package `DataFrames.jl` [3] being especially useful to connect and keep track of every component of this problem. Table 3 shows our code's results for different container sequences.

Table 3: Wall-building heuristic results.

Container sequence	Layers	Unfilled container volume (%)			
		1	2	3	4
A,A,A,A	44	9.09	9.09	9.09	47.46
B,B	28	7.69	34.68	–	–
C,C	28	5.22	82.19	–	–
B,A,A	36	7.69	9.09	56.12	–
C,A	32	5.22	48.54	–	–

For this set of boxes, remarkably little space is left in each container, except for the last one, where there are fewer boxes to pack. We conclude that this heuristic satisfies our unconstrained problem, giving us great insight into possible container loading orders. This, combined with information on the cost of transportation for each container, allows us to choose the most economical option.

References

[1] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. "Julia: A fresh approach to numerical computing". In: **SIAM review** 59.1 (2017), pp. 65–98. URL: `https://doi.org/10.1137/141000671`.

[2] Andreas Bortfeldt and Gerhard Wäscher. "Constraints in container loading–A state-of-the-art review". In: **European Journal of Operational Research** 229.1 (2013), pp. 1–20.

[3] Milan Bouchet-Valat and Bogumi Kamiski. "DataFrames.jl: Flexible and Fast Tabular Data in Julia". In: **Journal of Statistical Software** 107.4 (2023), pp. 1–32. DOI: `10.18637/jss.v107.i04`. URL: `https://www.jstatsoft.org/index.php/jss/article/view/v107i04`.

[4] Chin-Sheng Chen, Shen-Ming Lee, and QS Shen. "An analytical model for the container loading problem". In: **European Journal of operational research** 80.1 (1995), pp. 68–76.

[5] John A George and David F Robinson. "A heuristic for packing boxes into a container". In: **Computers & Operations Research** 7.3 (1980), pp. 147–156.