A Column Generation Heuristic for the Three-Dimensional Truck Loading Problem

Rick Willemsen & Bart van Rossum

Extended abstract submitted for presentation at the 12th Triennial Symposium on Transportation Analysis conference (TRISTAN XII)
June 22-27, 2025, Okinawa, Japan

January 28, 2025

Keywords: truck loading, three-dimensional bin packing, practical loading constraints, column generation

1 INTRODUCTION

We consider a three-dimensional truck loading problem with practical loading constraints. We are given a set of heterogeneous items, as well as a fleet of heterogeneous trucks that operate on fixed routes with a specified deadline for delivering the items to a plant. The problem consists of two key decisions. First, all items must be assigned to trucks corresponding to a fixed route and arrival date. Additional trucks of existing types can be used at a fixed cost. Items that arrive too early at the plant incur inventory costs due to limited space. Second, all items assigned to a single truck must be organised and arranged into stacks within the truck that satisfy all practical loading constraints. The goal is to find an assignment of items to trucks that minimises total truck and inventory costs. This problem is motivated by the ROADEF2022 challenge, where our solution approach was awarded the prize for best junior team.

Numerous practical constraints must be taken into account per truck, which we categorise into four types: item, stack, placement and weight constraints. Among these are the so-called guillotine constraints, which specify that the bottom face of each item must be fully supported from below. Some items are allowed to be rotated around a vertical axis. Additionally, we consider several practical constraints that are less often addressed in the literature. As trucks visit multiple plants and docks, the order constraint enforces that items are ordered according to the supplier pickup order, supplier dock loading order and plant dock loading order. The most challenging constraint is a so-called weight balance requirement, stating that the weight be evenly distributed over the truck at all times while travelling between suppliers. Moreover, a placement constraint mandates that all items are positioned at the front of the truck to prevent movement in the event of a sudden stop by the driver. Note that satisfying all constraints simultaneously is highly non-trivial. For example, directly shifting all items forward to satisfy the placement constraint is likely to be in conflict with the balance constraint.

We develop a scalable column generation heuristic to solve this problem. In a master problem, we select a set of loaded trucks which minimise total costs and cover all items. In the pricing problems, one for each truck type, potentially beneficial truck loadings are constructed and added to the master problem. We solve these pricing problems heuristically using a labelling algorithm applied to a dynamically defined stack graph. Additionally, we apply pre- and postprocessing steps to improve the obtained solutions.

To our knowledge, this is the first application of column generation to a truck loading problem with such a wide variety of practical constraints. Our problem can be seen as a generalisation of the three-dimensional bin packing problem, where it is common to assume homogeneous bins.

Mahvash *et al.* (2018) and Elhedhli *et al.* (2019) have proposed column generation approaches for three-dimensional bin packing problems. Our problem is more related to the multiple container loading problem, for which column generation approaches exist, see for instance Zhu *et al.* (2012), Wei *et al.* (2015) and Rajaei *et al.* (2024). These authors also consider heterogeneous trucks and heterogeneous items, but impose a more limited set of practical constraints.

2 METHODOLOGY

We develop a three-stage algorithm for the truck loading problem, a complete overview of which is given in Figure 1. In the first stage, we quickly generate multiple feasible solutions using both a greedy approach and an assignment model. The assignment model is a relaxation of the truck loading problem and provides a valid lower bound on the optimal objective value. Subsequently, we apply a construction heuristic to quickly convert solutions from the assignment model into feasible ones. This provides us with a valid upper bound. Typically, these solutions are already of decent quality.

In the second stage, we use the feasible solutions to warm-start the column generation algorithm for the non-relaxed problem. We first solve a linear programming relaxation until a stopping criteria is satisfied. Next, we perform a dive-and-fix procedure to obtain a feasible solution. Typically, the second stage solution significantly improves upon the first-stage solution.

Finally, in the third stage we unfix part of the current best solution and reapply the column generation algorithm to explore the neighbourhood of this solution, aiming for further improvements. This third stage is repeated until a time limit is reached. To enhance our results, we apply multiple acceleration techniques. In addition, we incorporate local search heuristics in a postprocessing step to improve each new solution.

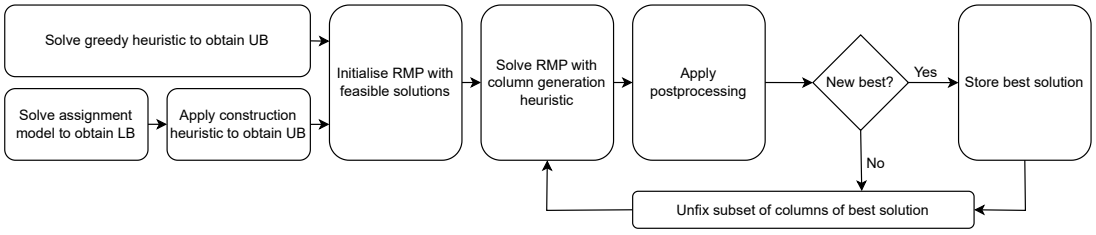


Figure 1 – *An overview of the different stages of the column generation heuristic. The two start heuristics provide lower bound (LB) and upper bound (UB) information on the optimal objective value. The solutions obtained from the start heuristics are used to initialise the restricted master problem (RMP), which is subsequently solved using column generation.*

The motivation behind our algorithm is as follows. First, we employ a construction heuristic to ensure that a feasible solution is always obtained, even for large-scale instances. Second, the second-stage master problem is able to make centralised decisions regarding the allocation of items to trucks, thereby hopefully avoiding local minima. Lastly, we implement tailored postprocessing heuristics to correct for the weaknesses of the column generation algorithm. In particular, while the column generation algorithm uses a strongly heuristic pricing algorithm that might not be able to generate the most efficient truck loading, we allow a local guided search over possible truck loadings in the postprocessing phase.

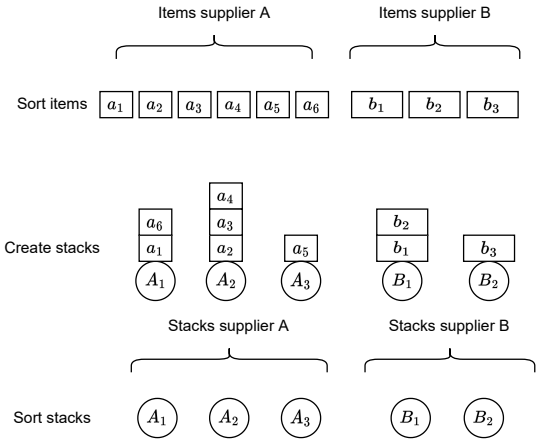
Pricing problem

The pricing problem consists of identifying a loaded truck with negative reduced cost. This process is intuitive and mirrors how one would manually load a truck, making it straightforward to integrate practical loading constraints and allows flexibility in selecting several strategies. We

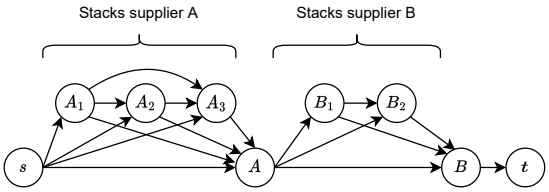
decompose the pricing problem over all truck types, modeling each as a resource constrained shortest path problem on a directed acyclic graph (DAG).

For each truck, we first retrieve the subset of items with negative reduced cost that potentially fit in this truck. These items are sorted by reduced cost and greedily partitioned into feasible stacks for each supplier, as shown in Figure 2a. This ensures that the order constraints are easily satisfied in subsequent steps. The stacks are again sorted by reduced cost. Each stack is represented by a node in a DAG as visualised in Figure 2b. We apply a labelling algorithm to find a path from the source to the sink, corresponding to a feasible truck loading with negative reduced cost. A label corresponds to a partial truck loading that satisfies all constraints except the weight balance constraint. Expanding a label to a stack node corresponds to adding a stack to a partial truck loading. Each label extension thus requires us to check feasibility of the resulting placed stack. There are several strategies to determine where to place stacks in the truck. Based on preliminary results we place new stacks on a set of corner points (Martello *et al.*, 2000).

We solve the labelling problem in a heuristic manner. We apply an aggressive dominance rule based on reduced cost only and limit the number of stored labels. Based on initial experiments it is not clear when and where to rotate a stack. Therefore, we propose multiple rotation strategies, such as random rotation, no rotation, or only rotating items of certain widths. We use a roulette wheel to determine which rotation strategy to use. The roulette wheel automatically updates the relative contribution of the rotation strategies to the incumbent solution. To handle the weight balance constraint, we explicitly verify this condition whenever moving to a new supplier node, indicated by A and B in Figure 2b.



(a) *The creation of stacks per supplier, with both items and stacks sorted by reduced cost.*



(b) *The sorted stacks are placed in a directed acyclic graph. The goal is to find a path from the source s to the sink t with negative reduced cost, corresponding to a feasible truck loading. Each supplier has its own node to ensure that the ordering constraints and the weight balance constraint are satisfied.*

Figure 2 – *Example of the construction of a directed acyclic graph containing stacks.*

3 RESULTS

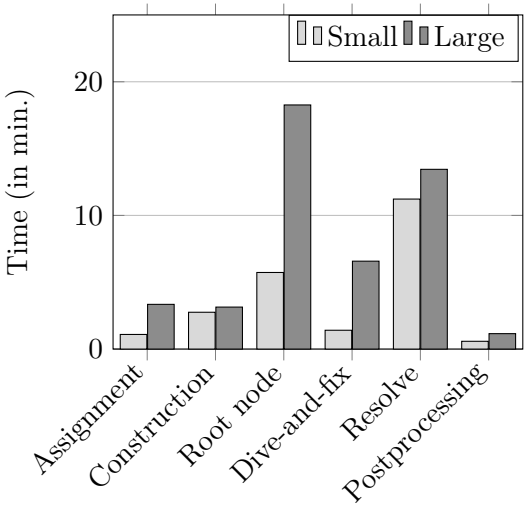
The instances used in our computational experiments are provided by the Renault Group during the ROADEF2022 challenge. There is a wide variety of instances, with a typical time horizon of around 10 to 15 weeks. To analyse the performance of our column generation heuristic we focus on a set of small and large instances. The small instances have on average 15,000 unique items and 1,900 trucks, while the large instances have on average 32,000 unique items and 3,700 trucks. There is a time limit of 30 and 60 minutes for small and large instances, respectively. We compare our method against the best-known solution across all ROADEF2022 submissions.

Figure 3 shows the computation time per stage of the column generation heuristic. The first stage of the column generation heuristic is often quite fast, the assignment model and construction heuristic steps take on average 3 minutes for the large instances. In the first stage,

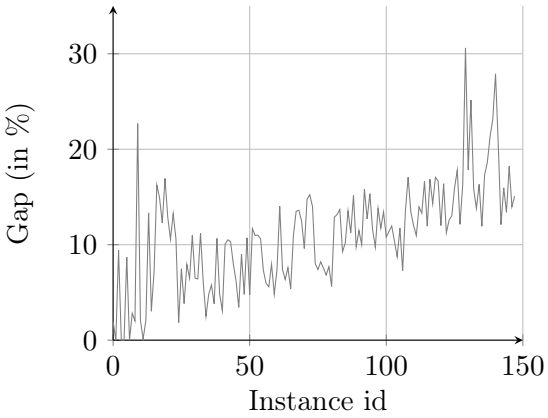
the obtained feasible solution has an average gap of 20% with respect to the best known solution. In the second stage, we solve the root node to find a fractional solution and after a dive-and-fix procedure we obtain a feasible solution. For small instances, it takes on average 7 minutes to find a feasible solution, whereas for large instances it takes on average 25 minutes. This is due to the tail-off effect in column generation; it takes time to converge, especially when solving the root node. The remainder of the time is spent on the third stage, the resolve procedure.

Figure 3b shows the percentage gap between the objective obtained from our column generation heuristic and the best known solution. We consider both small and large instances, where the instances are ordered based on the number of items per instance. The figure shows that for instances containing less items, the column generation heuristic is competitive and achieves a zero gap on multiple instances. Our column generation heuristic scales to larger instances, although the gap starts to increase.

The main idea of the column generation heuristic is to obtain good feasible solutions by making central decisions. A disadvantage of the current approach is that the pricing problem is solved using a strongly heuristic labelling algorithm which easily misses efficient loadings. We believe the labelling algorithm can be improved by using more ideas from the packing literature.



(a) *Running time (in minutes) per stage of the column generation heuristic for small and large instances.*



(b) *Gap (in %) between the objective obtained from the column generation heuristic and the best known solution for small and large instances. The instances are ordered based on the number of items.*

Figure 3 – *Computational results of the column generation heuristic on small and large instances.*

References

Elhedhli, Samir, Gzara, Fatma, & Yildiz, Burak. 2019. Three-dimensional bin packing and mixed-case palletization. *INFORMS Journal on Optimization*, **1**(4), 323–352.

Mahvash, Batoul, Awasthi, Anjali, & Chauhan, Satyaveer. 2018. A column generation-based heuristic for the three-dimensional bin packing problem with rotation. *Journal of the Operational Research Society*, **69**(1), 78–90.

Martello, Silvano, Pisinger, David, & Vigo, Daniele. 2000. The three-dimensional bin packing problem. *Operations research*, **48**(2), 256–267.

Rajaei, Maryam, Moslehi, Ghasem, & Reisi-Nafchi, Mohammad. 2024. The multiple container loading problem with loading docks. *International Transactions in Operational Research*, **31**(3), 1671–1698.

Wei, Lijun, Zhu, Wenbin, & Lim, Andrew. 2015. A goal-driven prototype column generation strategy for the multiple container loading cost minimization problem. *European Journal of Operational Research*, **241**(1), 39–49.

Zhu, Wenbin, Huang, Weili, & Lim, Andrew. 2012. A prototype column generation strategy for the multiple container loading problem. *European Journal of Operational Research*, **223**(1), 27–39.