

Cargo securing under multi-drop and axle weight constraints

Hatice Calik^{*1}, Marc Juwet², Hande Yaman³, and Greet Vanden Berghe¹

¹KU Leuven, Department of Computer Science, CODES, NUMA, Ghent, Belgium

²KU Leuven, Department of Mechanical Engineering, Ghent, Belgium

³KU Leuven, Faculty of Economics and Business, ORSTAT, Leuven, Belgium

August 24, 2022

Abstract

This study emerges from a real-world cargo securing application to ensure safer road transportation and falls into the category of container loading problems with practical constraints. The literature has lacked efficient methods for the secure loading of items with non-identical dimensions, weights and rotations into containers while ensuring that the securing efforts necessary are minimal when also taking into account multi-drop and axle weight constraints. This paper puts forward a new way of ensuring cargo stability that gives rise to a novel combinatorial optimization problem, which is a generalization of the two-dimensional rectangular strip packing problem with orthogonal rotations. We formally demonstrate the intractability of the problem and provide a mixed integer programming formulation. The formulation is based on a discretization of the packing polyhedron, enabling us to model a range of complicated and practical constraints. A group of practical constraints tends to be large in number and makes it difficult to solve the formulation in a reasonable amount of time. In order to overcome this difficulty, we develop an exact algorithmic framework. This framework initially solves certain relaxations of the problem to obtain strong lower bounds before subsequently embedding those lower bounds into a branch-and-cut algorithm. The experimental study serves three purposes: (i) evaluating the performance of the algorithmic framework and the mathematical formulation to assess the merits of the two methods, (ii) identifying the characteristics of hard problem instances and (iii) extracting insights regarding challenges in cargo securing to help managers and practitioners in decision making.

Keywords: Packing, container loading, cargo securing, multi-drop, axle weights.

1 Introduction

The *Mobility and Transport* department of the European Commission reports that up to 25% of truck accidents are related to poorly secured cargo (Commision, 2021). Improperly loaded cargo may fall out of the container or trailer. It may also affect a vehicle's balance and its braking ability, which may result in the vehicle tipping over and endangering the safety of not only the truck driver but also other road users. Therefore, the proper loading and securing of cargo is crucial in freight logistics. Minimum requirements for cargo loading in road transport for the European Union (EU) member states is provided by EU directive 2014/47. Nevertheless, following these guidelines

^{*}hatice.calik@kuleuven.be (Corresponding author)

with precision is not always possible by human practitioners unless they are also provided with an external recommendation computed by an automated tool. Human operators utilize only a subset of guidelines in combination with their experience and intuition (trial and error) to ensure cargo security. This often results in loading schemes that violate the limitations detailed in the aforementioned EU directive.

The focus of this study is on a specific application commonly encountered by logistics practitioners in Europe whose activities are regulated by the EU directive. Consider a vehicle container to transport a given set of pallets to a given set of customers. Each pallet is associated with a single customer, while multiple pallets might be destined for the same customer. Pallets must be loaded into the rectangular-shaped container at a depot location by taking into account the predetermined visit order to the customers. Pallets also have rectangular shapes which might be of different dimensions, and the load weight of pallets might also vary even among the pallets associated with the same customer. All pallets must fit onto the container surface without being stacked on top of each other. It is possible to rotate pallets 90 degrees around a vertical axis and hence place each pallet in two different orientations.

For operational reasons, pallets can only be unloaded at the dedicated customer location and cannot be moved at any other customer location. The major reason is to avoid confusion regarding responsibility for safety during the rest of the journey. Additional reasons involve one or more of the following:

- Extra movements would burden operators with additional unproductive workload.
- Customers do not provide space for unloading anything other than their own cargo.
- Customers authorize only a limited amount of parking time for the delivery vehicles.

Unloading is performed from the rear container door (no side unloading) with only a linear movement of each pallet. This is a conservative assumption, but it is essential since heavy pallets are unloaded by using large electric devices that are not flexible enough to reach behind other pallets. Thus, the position of pallets inside the container must be decided at the depot in such a way that no additional handling of any pallet will be necessary. In the remainder of the paper, we refer to these restrictions as the *multi-drop* constraints.

While taking into account the aforementioned loading constraints, one should also make sure that the loading scheme is safe for road transportation. More specifically, that the weight limits on the vehicle axles are not exceeded and that the pallets are properly secured so they do not move inside the container when the vehicle is in motion. For proper cargo securing, several methods can be used in combination. In the specific application that we consider, pallets are secured by utilizing air cushions and rigid blocking bars. The goal is to ensure a safe feasible loading scheme that requires a minimum amount of securing time and materials. We refer to this problem as the *Cargo Securing Problem with Multi-Drop and Axle Weight constraints (CSP-MD-AW)*.

The CSP-MD-AW consists of several components related to the rectangular two-dimensional strip packing problem (2D-SPP) and the two-dimensional container loading problem (2D-CLP). The most recent bibliometric overview on the 2D-SPP by Neuenfeldt et al. (2021) reveals that real-world practical constraints are not sufficiently addressed in the 2D-SPP academic literature. One of the main reasons identified by Neuenfeldt et al. (2021) for this trend of neglecting practical constraints is the difficulty of representing them in heuristics and mathematical models. A similar observation is made by Bortfeldt and Wäscher (2013) for the container loading problem (CLP) literature. This paper advances an attempt to overcome this challenge and provide useful methods and insights both for industrial managers and practitioners as well as academic researchers.

The CSP-MD-AW accommodates several practical constraints that are challenging to represent in a mathematical programming formulation. The efficient two-dimensional loading and packing methods in the literature cannot be utilized for solving the CSP-MD-AW as they are not compatible with axle weight and securing constraints. Due to axle weight limits, it is particularly difficult to make decisions concerning pallet positions and it may be inevitable to leave an unpredictable amount of space between pallets. The center of gravity and effective weights on the axles must be calculated to determine feasible pallet positions and secure pallets not only at the depot but also at each customer location. In order to overcome this challenge, we utilize a discretization of the container surface. This discretization facilitates the accommodation of these challenging practical constraints in a mixed integer programming (MIP) formulation. We anticipate that this modeling framework can easily accommodate some other potential real-world problem variants, which will be briefly mentioned throughout this paper. The main drawback of this discretization is that the number of variables and especially the multi-drop constraints can be too large for certain problem instances. Therefore, we develop an exact algorithmic framework which iteratively solves the relaxations of the CSP-MD-AW to obtain strong lower bounds. These bounds help reduce the size of the MIP to be solved by a branch-and-cut procedure which activates multi-drop constraints only if they are violated.

The key contributions of this paper are as follows:

- (i) A novel container loading problem that emerges from real-world practice and safety regulations in Europe for freight transportation is introduced. Its relation with other published work is indicated and NP-Hardness is proven.
- (ii) A mixed integer programming formulation based on container surface discretization is provided.
- (iii) An exact algorithmic framework to solve the MIP formulation effectively is proposed.
- (iv) Several managerial insights as well as future research directions are discussed.

Section 2 of this paper provides a detailed problem description and a proof of complexity. Section 3 highlights the differences between the CSP-MD-AW and the problems studied prior to this paper and evaluates the suitability of the existing methods. This is followed by a mathematical programming formulation for the CSP-MD-AW in Section 4. In order to solve this formulation in a reasonable time frame, Section 5 introduces an algorithmic framework. Section 6 presents an analysis through visualizations of solutions instead of numerical results and provides insights obtained from the computational study conducted. Section 7 concludes the paper and outlines several future research directions.

2 Notation and problem description

An ordered set $T = \{i_1, i_2, \dots, i_t\}$ of customers must be served by a vehicle departing from a depot node '0'. More specifically, a set P_i of rectangular pallets must be delivered to each customer $i \in T$. In doing so, one must obey certain loading and cargo safety restrictions. For each customer $i \in T$, the pallets in P_i are not necessarily identical either in size or weight (see Figure 1). Parameter w_p denotes the weight of the cargo placed on each pallet $p \in P = \bigcup_{i \in T} P_i$ (including the weight of the pallet itself), which is homogeneously distributed. The customer associated with pallet $p \in P$ is denoted by $c(p)$ and $c(p) = i$ for each $p \in P_i$.

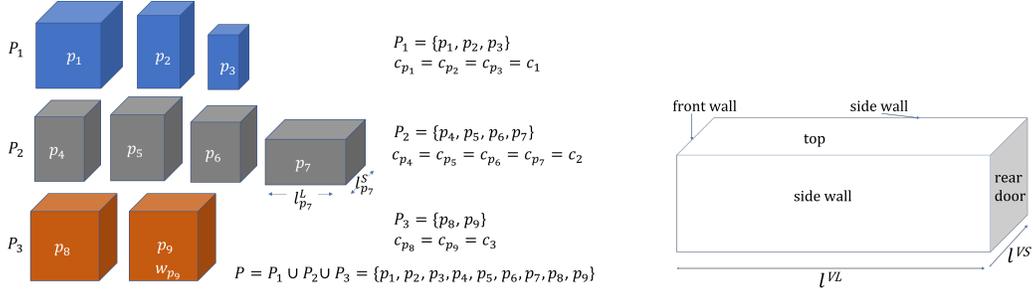


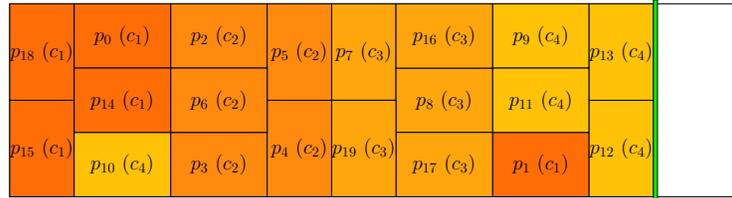
Figure 1: Pallet and container parameters.

Pallet positioning constraints: Pallets may not be stacked on top of each other, they cannot overlap and all the pallets must lie entirely inside the container, which is the part of the vehicle to load pallets. It is possible to position an arbitrary pallet in two different orientations:

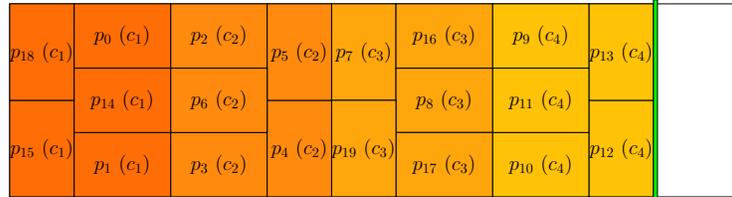
L-ways: The long edge of the pallet is parallel to the side walls of the container.

S-ways: The short edge of the pallet is parallel to the side walls of the container.

Multi-drop constraints: The placement of cargo inside a container should take into account the order in which customers are visited to ensure that the pallets of one customer are not blocked by those of another who will be visited later in the route. Figure 2a illustrates the top view of a sample loading scheme that violates multi-drop restrictions for a given customer route $T = \{c_4, c_3, c_2, c_1\}$. In this figure and in the remainder of the paper, we assume that the unloading of pallets is performed from the rear door (right-hand side in top view) of a container. In Figure 2a, pallet p_1 of customer c_1 blocks several pallets associated with other customers (p_3 and p_4 of c_2 , p_{19} and p_{17} of c_3 and p_{10} of c_4). Swapping the positions of p_1 and p_{10} leads to a loading scheme that respects the multi-drop restrictions, as depicted in Figure 2b.



(a) Multi-drop constraints are violated.



(b) Multi-drop constraints are respected.

Figure 2: Overhead views of sample loading schemes for $T = \{c_4, c_3, c_2, c_1\}$.

Securing: To avoid sliding and tilting, each empty space between pallets and between pallets and the front and side walls of the container must be filled using air cushions. These materials are

often not reusable and costly. Moreover, filling each empty space requires a non-negligible amount of time and should therefore be avoided as much as possible. The securing from the rear is done with a blocking bar. Blocking bars are mounted horizontally between the side walls of the container. They typically block the closest pallet(s) to the rear door of the container as illustrated in Figure 3. The goal is to pack the load behind the bar as densely as possible. This is considered advantageous for two reasons: (i) the densely loaded pallets better support each other and potentially little or no securing is needed and (ii) pallets are more evenly distributed along the two sides of the container and this potentially better balances the weight across the wheels which is safer.



Figure 3: Securing pallets with a blocking bar and inflatable cushions. After unloading pallets, the bar is moved closer towards the front wall and any empty spaces behind the bar are filled with cushions.

After unloading pallets at a customer location, the empty spaces created (if any) must be filled with cushions and the blocking bar must be moved closer to the front wall (if possible), as illustrated in Figure 3. In practice, securing pallets is undesirable not only at the depot, but also at customer locations. Therefore, it is not sufficient to simply minimize the empty space and securing costs only at the depot. The initial loading scheme should be determined while bearing in mind the necessary securing efforts at customer locations. However, quantifying the required securing efforts precisely with a mathematical equation is not straightforward. In an attempt to approximate this with manageable computational efforts, we minimize the total distance of the blocking bar from the front wall throughout the entire vehicle trip. This objective function can formally be expressed as follows. Suppose ϑ_i is the distance of the blocking bar to the front wall before departing from node $i \in T^0 = \{0\} \cup T = \{0, i_1, i_2, \dots, i_t\}$. The objective function is then $\min_{\vartheta \in X} \sum_{i \in T^0 \setminus \{i_t\}} \vartheta_i$ where X is the

feasible domain of vector ϑ .

Axle weight constraints: In addition to securing pallets, the distribution of the weight inside the container plays a crucial role in safe road transportation. We consider a semi-trailer supported at the coupling and center of the rear axles, as illustrated in Figure 4. Axle weight restrictions require weights f^1 and f^2 effective on the indicated axles to be no greater than the legal limits F^1 and F^2 , respectively.

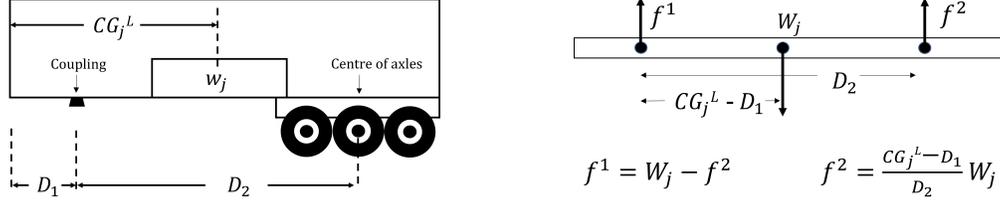


Figure 4: Weights f^1 and f^2 on the axles of the vehicle given a load with weight W_j is placed such that the center of load gravity is CG_j^L meters away from the front wall.

The following list of parameters fully describes the CSP-MD-AW.

D_1 is the distance from the front of the container to the coupling.

D_2 is the distance between the coupling and the center of the axles.

F^1 and F^2 are the maximum weights allowed on the coupling and the center of the axles, respectively.

T is the ordered set of customer nodes to visit. $T^0 = \{0\} \cup T = \{0, i_1, \dots\}$.

P_i is the set of pallets to be delivered to customer $i \in T$ and $P = \bigcup_{i \in T} P_i$.

$c(p) \in T$ is the customer of pallet $p \in P$, i.e., $p \in P_{c(p)}$.

w_p is the weight of pallet $p \in P$.

T_i^{fol} is the ordered set of successors of node $i \in T^0$ (nodes to be visited after i).

$P_i^{fol} = \bigcup_{j \in T_i^{fol}} P_j$ is the set of pallets delivered to the successors of $i \in T^0$ (successor pallets).

$W_i = \sum_{p \in P_i^{fol}} w_p$ is the total weight in the container when departing from node $i \in T^0$.

l^{VS} and l^{VL} are the short and long edge lengths of the container, respectively.

l_p^S and l_p^L are the lengths of the short and long edges of pallet $p \in P$, respectively.

Before proceeding with the problem formulation, we prove in Proposition 2.1 that the CSP-MD-AW is an NP-Hard problem.

Proposition 2.1. *The CSP-MD-AW is NP-Hard.*

Proof. The CSP-MD-AW reduces to the *Two-Dimensional Strip Packing Problem with Rotations* when $|T| = 1$ and $l^{VL} = F^1 = F^2 = \infty$. The *Two-Dimensional Strip Packing Problem with Rotations* is already known to be NP-Hard (Iori et al., 2021). Hence the CSP-MD-AW is NP-Hard. \square

3 Related work

As mentioned in the introduction, some of the practical constraints present in the CSP-MD-AW relate to the 2D-SPP and 2D-CLP. This section positions the CSP-MD-AW with respect to these two problem categories. Only the most closely related papers considering the two-dimensional packing or loading problems are reviewed. We foresee that the methods focused on higher-dimension problems would be unnecessarily complicated for the CSP-MD-AW given that even the 2D problems with practical constraints lack efficient methods. Any loading scheme violating the multi-drop constraints is unacceptable for the CSP-MD-AW and methods that fail to enforce them do not guarantee a feasible CSP-MD-AW solution. Thus, we restrict the review to those strictly enforcing the multi-drop constraints. For readers interested in broader reviews on the 2D-SPP and CLP, we refer to Neuenfeldt et al. (2021) and Bortfeldt and Wäscher (2013), respectively.

Neuenfeldt et al. (2021) identify 27 papers that consider practical constraints in the 2D-SPP. Multi-drop constraints are addressed in five of these (Da Silveira et al., 2013, 2014; De Queiroz and Miyazawa, 2013, 2014; Wei et al., 2019). Among those, only De Queiroz and Miyazawa (2013, 2014) consider weight distribution (load balance) and load bearing constraints. The 2D-SPP variants studied by De Queiroz and Miyazawa (2013, 2014) consider a bin with length L and infinite height and a finite set of items with predetermined length, height and orientation. Thus, the bin is filled with rectangular items from the top. The objective is to minimize the height required to pack all the items. The load balance constraints require that the center of gravity is always within a predetermined rectangular area, referred to as *envelope*. This approach is also frequently adopted in early CLP studies. Ramos et al. (2018) indicate that this envelope no longer possesses a rectangular shape if vehicle specifications are taken into account.

Realistic load balance methods based on axle weight restrictions appear in several recent CLP studies (Alonso et al., 2019, 2017, 2020; Lim et al., 2013; Liu et al., 2017). However, these papers do not consider multi-drop constraints. Silva et al. (2018) address the multi-drop load balance recovery problem, which is not a CLP variant but closely related to the CLP. Given a cargo arrangement and a complete route, their algorithm minimizes the number of items to be rearranged to ensure load balance during the entire route.

Multi-drop constraints are more often addressed in vehicle routing problems (VRPs) with loading constraints, which can be considered extensions of the CLP. In combined routing-and-loading problems the packing component is usually a decision problem, which is equivalent to the *orthogonal packing problem* (Baker et al., 1980). Côté et al. (2014) provide an exact approach for handling the multi-drop constraints within the two-dimensional variant of this decision problem. Heuristic approaches are also available for tackling these *NP-Complete* decision problems. Yet the review by Pollaris et al. (2015) indicates the lack of integrated methods for solving the VRP with multi-drop and axle weight constraints. In a follow-up study, Pollaris et al. (2016) address a VRP variant with loading restrictions including multi-drop and axle weight constraints (VRP-MD-AW). The container surface is divided into two rows and several columns (depending on the container length) where each cell accommodates at most one pallet. Customer pallets have identical dimensions and they have to be placed in the same specified orientation. Pallets are placed alternately on the left and right rows without any gap between two adjacent pallets. Pallets of a customer have identical weights and cannot be aligned in a single row. The load weights on the two axles are limited by both a maximum and a minimum value while the multi-drop restrictions must be respected. The minimum axle weight restrictions are not present in the CSP-MD-AW because the real-world application inspiring the CSP-MD-AW does not consider such limits. Since the remaining loading restrictions in the

VRP-MD-AW reduce to a very restrictive special case of the CSP-MD-AW, the method provided by Pollaris et al. (2016) cannot be utilized for solving the CSP-MD-AW. A recent variant of the VRP-MD-AW which allows split deliveries has been studied by Alonso et al. (2022). As in the VRP-MD-AW, no empty spaces between pallets are allowed. However, empty space between the front wall and the pallets of the final customer is permitted.

Neither Pollaris et al. (2016) nor Alonso et al. (2022) quantify (and minimize) the additional securing efforts needed at intermediate customer points.

Although enforcing no empty space between pallets can be an effective method for ensuring cargo stability, it is not always possible to obtain such loading schemes for a given sequence. This is often the case when different pallet dimensions are considered. Therefore, within a VRP framework, such restrictions may lead to longer or additional vehicle routes which incur higher costs. In fact, the loading schemes created by the methods of Pollaris et al. (2016) and Alonso et al. (2022) may require securing of a pallet after every customer visit if a pallet is not supported by another pallet from one side. Pollaris et al. (2016) and Alonso et al. (2022) do not penalize or forbid such schemes although in reality such insufficiently supported pallets may slide and negatively affect the vehicle stability.

An attempt to model a generalization of the VRP-MD-AW where pallets may have general properties as in the CSP-MD-AW (nonidentical dimensions and weights with a possibility of 90 degrees rotation) was made by Çalik et al. (2019). However, securing restrictions were not addressed and the proposed model was not validated with computational experiments. In fact, the multi-drop constraints in the proposed model by Çalik et al. (2019) prevent blocking if the pallets must be positioned in one dimension, but may lead to blocking loading schemes if the pallets of non-immediate successors are allowed to be placed in adjacent rows. Although multi-drop restrictions are more commonly considered in recent VRP studies (Gandra et al., 2021; Ferreira et al., 2021), axle weight constraints remain rarely addressed (Krebs and Ehmke, 2021).

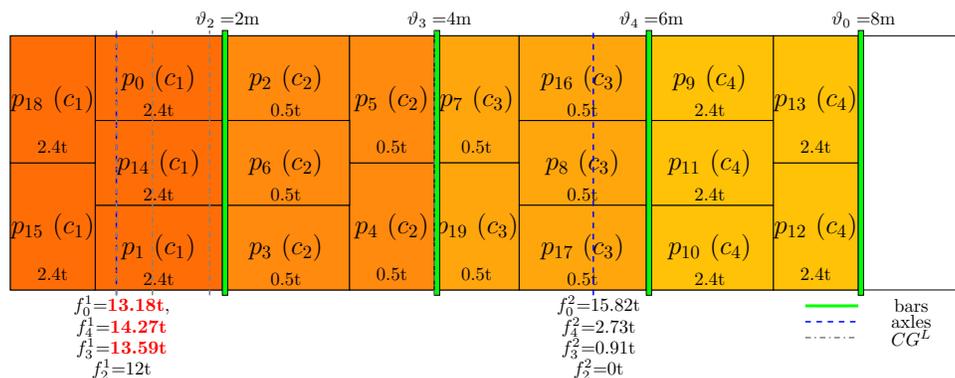


Figure 5: A loading scheme with $\sum \vartheta = 20m$ for $T = \{c_4, c_3, c_2, c_1\}$. Load weight limit on the coupling, which is 12.8t, is exceeded all the way from the depot until unloading the pallets of c_2 .

4 An integer programming formulation for the CSP-MD-AW

The objective of the CSP-MD-AW is to find a feasible and secure loading scheme. It is difficult to intuitively predict a finite set of potential points or a specific region of the container surface for placing each pallet or the pallets of a customer that guarantee an optimal solution. For example, consider the set of pallets and customers as in Figure 2a. Let the weight of the pallets associated

with customers c_1 and c_4 be all equal to $2.4t$ and the weight of those associated with customers c_2 and c_3 be all equal to $0.5t$. Intuitively, one might think that loading customer pallets as depicted in Figure 2b would be the best loading scheme. However, Figure 5 demonstrates how that loading scheme violates the axle weight restrictions. A feasible loading scheme is instead visualized in Figure 6.

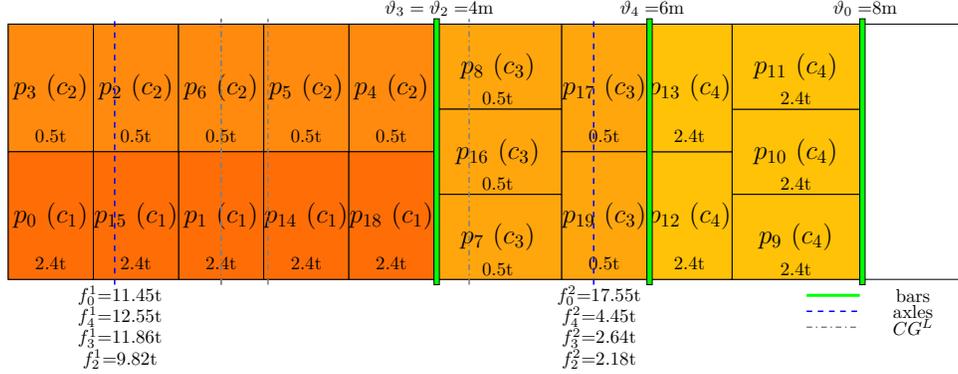


Figure 6: Best feasible solution provided by the solver with $\sum \vartheta = 22m$ for $T = \{c_4, c_3, c_2, c_1\}$.

In order to overcome this difficulty, we employ a grid discretization of the container surface, as is commonly done for solving irregular packing problems as well as some strip packing problems with practical constraints (Toledo et al., 2013; De Queiroz and Miyazawa, 2014). As depicted in Figure 7, we divide the container surface into identical grid squares with edge length l^G . We prefer this form

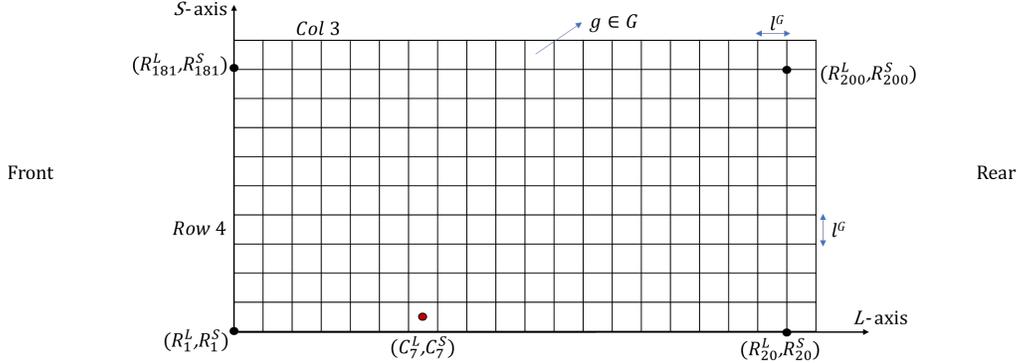


Figure 7: The container surface divided into identical grid squares visualized on the first quadrant of a two-dimensional Cartesian coordinate system.

of grid discretization over *normal patterns* and *useful numbers* (de Almeida Cunha et al., 2020) as we foresee that those alternative forms of discretization may be too restrictive in the presence of axle weight restrictions.

We choose two adjacent bottom corners of each pallet as potential reference points, one of which is to be located on the predefined reference point (the bottom left corner) of a grid square. All pallets must be positioned on distinct grid squares by respecting the aforementioned constraints. Note that smaller l^G values lead to a larger number of grid squares and a larger solution space for pallet positions. While larger l^G values shrink this solution space, they might lead to lower quality solutions or even infeasibilities if l^G is too large. Therefore, the selected l^G value plays a crucial

role in finding the right balance between solution quality and computational efficiency. We propose to set l^G equal to the greatest common divisor of the edge lengths of the pallets and the container. This value is sufficiently small to enable a loading scheme with no empty space between pallets.

What follows is a list of parameters resulting from the grid structure.

The container surface (see Figure 7):

L -axis and S -axis correspond to the long and short edges of the container surface, respectively.

$nCol = \lceil l^{VL}/l^G \rceil$ is the number of grid square columns (squares along the L -axis).

$nRow = \lceil l^{VS}/l^G \rceil$ is the number of grid square rows (squares along the S -axis).

Grid squares (see Figure 7):

G is the set of (identical) grid squares on the container surface.

l^G is the edge length of each grid square.

(R_g^L, R_g^S) is the reference point of grid square $g \in G$ where R_g^L and R_g^S are its coordinate values on the L -axis and S -axis, respectively.

(C_g^L, C_g^S) is the center point of grid square $g \in G$.

row_g and col_g are the row and column indices of $g \in G$, respectively.

Pallets (recall Figure 1):

n_p is the number of grid squares which pallet p has to occupy ($n_p = \lceil l_p^L/l^G \rceil * \lceil l_p^S/l^G \rceil$).

w_p^G is the weight of pallet $p \in P$ per grid square ($w_p^G = w_p/n_p$).

The remainder of this section presents the MIP formulation. The constraints and objective function are first grouped into separate subsections, before finally bringing everything together as a whole.

Decision variables and additional notation are introduced as needed. We begin with the pallet positioning decisions and constraints.

4.1 Pallet positioning constraints

The pallet positioning constraints we utilize can be considered a generalization of the packing constraints by Beasley (1985) so that the two orientations of the pallets can be taken into account in the model. We define two sets of binary variables to select the position of pallets on the container surface. These variables also determine pallet orientations. Figure 8 illustrates L -ways and S -ways pallet positioning with these variables.

$y_{pg}^L = 1$ if $p \in P$ is positioned L -ways on the reference point of grid square $g \in G$, 0 otherwise.

$y_{pg}^S = 1$ if $p \in P$ is positioned S -ways on the reference point of grid square $g \in G$, 0 otherwise.

Parameters A_{pg}^L and A_{pg}^S depicted in Figure 8 denote the rectangular areas on the container surface covered by pallet p when p is (hypothetically) positioned L -ways and S -ways on grid square g , respectively. Formally,

$$A_{pg}^L = \{h \in G : R_g^S \leq R_h^S \leq R_g^S + l_p^S, R_g^L \leq R_h^L \leq R_g^L + l_p^L\} \text{ and}$$

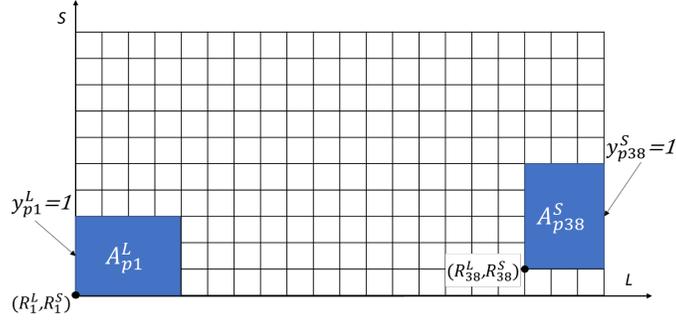


Figure 8: Illustration of L -ways and S -ways pallet positioning using y^L and y^S variables.

$$A_{pg}^S = \{h \in G : R_g^S \leq R_h^S \leq R_g^S + l_p^L, R_g^L \leq R_h^L \leq R_g^L + l_p^S\}.$$

If $l^G < l_p^S$ or $l^G < l_p^L$ as in Figure 9, not every grid square will be eligible to accommodate the reference points of p . For each pallet p , G_p^L and G_p^S denote the set of potential grid squares, on which p can be positioned L -ways and S -ways, respectively. Formally,

$$G_p^L = \{g \in G : R_g^S + l_p^S \leq l^{VS}, R_g^L + l_p^L \leq l^{VL}\} \text{ and}$$

$$G_p^S = \{g \in G : R_g^S + l_p^L \leq l^{VS}, R_g^L + l_p^S \leq l^{VL}\}.$$

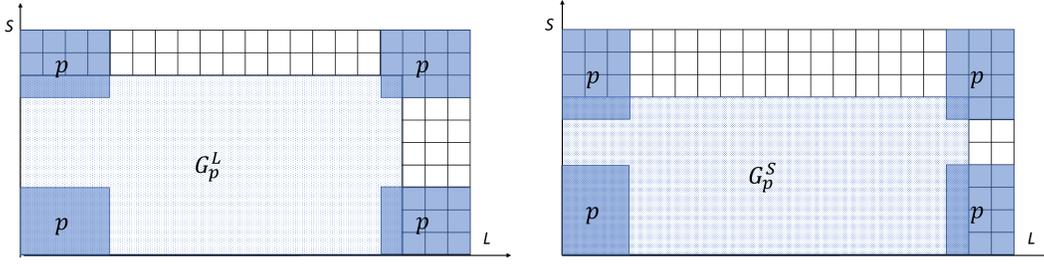


Figure 9: Feasible sets of grid squares to position a pallet L -ways (left) and S -ways (right).

Defining an auxiliary variable x_{pg} as in Constraints (2) makes it easier to follow the details of the MIP formulation. Note that such equality constraints and variables are typically eliminated during the *presolve* phase of IP solvers. Therefore, replacing all \mathbf{x} variables by the right-hand side of Constraints (2) results in a more efficient implementation. Now we are ready to present the pallet positioning constraints:

$$\sum_{g \in G_p^S} y_{pg}^S + \sum_{g \in G_p^L} y_{pg}^L = 1, \quad \forall p \in P \quad (1)$$

$$x_{pg} = \sum_{\substack{h \in G_p^S: \\ g \in A_{ph}^S}} y_{ph}^S + \sum_{\substack{h \in G_p^L: \\ g \in A_{ph}^L}} y_{ph}^L, \quad \forall p \in P, g \in G \quad (2)$$

$$\sum_{p \in P} x_{pg} \leq 1, \quad \forall g \in G \quad (3)$$

$$y_{pg}^L \in \{0, 1\}, \quad \forall p \in P, g \in G_p^L \quad (4)$$

$$y_{pg}^S \in \{0, 1\}, \quad \forall p \in P, g \in G_p^S \quad (5)$$

Constraints (1) assign each pallet to exactly one grid square to accommodate either the *L-ways* or *S-ways* reference point of that pallet (but not both). Therefore, the orientation of pallets is also determined by these constraints. Clearly, each pallet p with a chosen reference point occupies a rectangular area (RA) of grid squares. By Constraints (1) and (2), for any grid square g falling in RA , the corresponding x_{pg} variable takes a value equal to one whereas $x_{pg} = 0$ for any g that remains outside RA . In other words, x_{pg} indicates whether g is occupied by pallet p or not. Constraints (3) prevent any grid square from being occupied by multiple pallets. Constraints (4) and (5) are binary restrictions for the pallet positioning variables.

4.2 Objective function and securing constraints

In accordance with the notation introduced in Section 2, decision variable $\vartheta_i \geq 0$ denotes the distance of the blocking bar from the front wall when departing from node $i \in T^0 \setminus \{i_t\}$. The CSPMD-AW minimizes the sum of these distance values, which is formally expressed as $\sum_{i \in T^0 \setminus \{i_t\}} \vartheta_i$. By Constraints (6), ϑ_i is enforced to be greater than or equal to the L -axis value of the furthest point occupied by the successor pallets of i .

$$\vartheta_i \geq \sum_{g \in G_s^S} y_{pg}^S (R_g^L + l_p^S) + \sum_{g \in G_s^L} y_{pg}^L (R_g^L + l_p^L), \quad \forall i \in T^0 \setminus \{i_t\}, p \in P_i^{fol} \quad (6)$$

$$\vartheta_i \geq 0, \quad \forall i \in T^0 \setminus \{i_t\} \quad (7)$$

4.3 Multi-drop constraints

In order to formulate the multi-drop constraints, this section introduces the following *corridor* parameters for each $p \in P$ and $g \in G$ pair. Given that the *L-ways* reference point of pallet p is placed on g , COR_{pg}^L defines a rectangular corridor with a width of l_p^S starting from g and proceeding towards the rear of the container as depicted in Figure 10. A similar corridor COR_{pg}^S is defined for the placement of the *S-ways* reference point of p on g . Formally,

$$COR_{pg}^L = \{h \in G : R_g^S \leq R_h^S \leq R_g^S + l_p^S, R_g^L \leq R_h^L\} \text{ and}$$

$$COR_{pg}^S = \{h \in G : R_g^S \leq R_h^S \leq R_g^S + l_p^L, R_g^L \leq R_h^L\}.$$

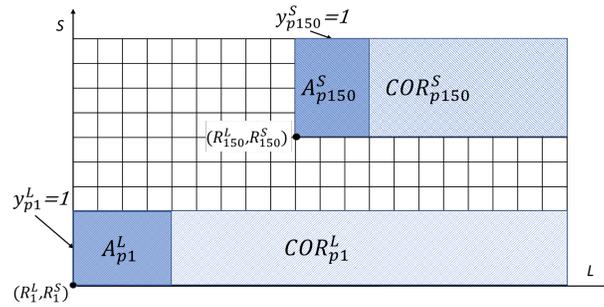


Figure 10: The corridors towards the rear door for given pallet positions.

Multi-drop constraints (8) and (9) employ the following idea: once a reference point is chosen for a pallet, the corridor towards the back of the container must not be occupied by any successor

pallet. Recall that P_i^{fol} denotes the set of pallets delivered to the successors of $i \in T^0$ (successor pallets) and $c(p) \in T \subset T^0$ denotes the customer associated with pallet p .

$$\sum_{\substack{h \geq g: \\ h \in COR_{pg}^L}} \sum_{p_1 \in P_{c(p)}^{fol}} x_{p_1 h} \leq \sum_{p_1 \in P_{c(p)}^{fol}} n_{p_1} (1 - y_{pg}^L), \quad \forall p \in P, g \in G_p^L \quad (8)$$

$$\sum_{\substack{h \geq g: \\ h \in COR_{pg}^S}} \sum_{p_1 \in P_{c(p)}^{fol}} x_{p_1 h} \leq \sum_{p_1 \in P_{c(p)}^{fol}} n_{p_1} (1 - y_{pg}^S), \quad \forall p \in P, g \in G_p^S \quad (9)$$

These constraints can be further strengthened by disaggregating them for each successor pallet or for each grid square in the corridors. We provide several disaggregated versions of the multi-drop constraints in A. Constraints (8) and (9) are fewer in number compared to the disaggregated constraints, but they are also weaker (the polyhedron of the linear programming relaxation contains more fractional solutions). On one hand, the disaggregated constraints exhaust the model as there can be an excessive number of them for certain instances. On the other hand, they may help closing the dual gap in fewer iterations during the branch-and-bound procedure.

4.4 Axle weight constraints

This section introduces a group of constraints that ensures that the total weight effective on each axle of the vehicle is within the corresponding safety limit at every segment of the delivery route. Suppose f_i^1 and f_i^2 are intermediary decision variables that denote the load weights effective on the coupling and the centre of axles when departing from node $i \in T^0$, respectively. Constraints (10) and (11) limit the load weights on the two axles with the predefined values.

$$f_i^1 \leq F^1 \quad \forall i \in T^0 \quad (10)$$

$$f_i^2 \leq F^2 \quad \forall i \in T^0 \quad (11)$$

In order to precisely compute the values of f_i^1 and f_i^2 , we define an additional intermediary variable CG_i^L that denotes the center of gravity on the L -axis when departing from node $i \in T^0$. Recall from Figure 4 in Section 2 that Equations (12) and (13) provide the values of f_i^1 and f_i^2 .

$$f_i^2 = \frac{(CG_i^L - D_1)}{D_2} W_i \quad \forall i \in T^0 \quad (12)$$

$$f_i^1 = W_i - f_i^2 \quad \forall i \in T^0 \quad (13)$$

Thanks to our discretization method, we can easily compute CG_i^L with linear equation (14).

$$CG_i^L = \frac{\sum_{g \in G} C_g^L \sum_{p \in P_i^{fol}} w_p^G x_{gp}}{W_i} \quad \forall i \in T^0 \quad (14)$$

We eliminate intermediary decision variables CG_i^L , f_i^1 and f_i^2 by means of several mathematical manipulations and finally express the axle weight constraints as in (15) and (16).

$$\sum_{g \in G} C_g^L \sum_{p \in P_i^{fol}} w_p^G x_{pg} - D_1 W_i \leq F^2 D_2, \quad \forall i \in T^0 \quad (15)$$

$$W_i D_2 - \sum_{g \in G} C_g^L \sum_{p \in P_i^{fol}} w_p^G x_{pg} - D_1 W_i \leq F^1 D_2, \quad \forall i \in T^0 \quad (16)$$

4.5 The MIP formulation

Given all constraints and the objective function detailed in Sections 4.1-4.4, the CSP-MD-AW can finally be formulated as a mixed integer program as follows:

$$\begin{array}{ll}
 FP & \min \quad \sum_{i \in T^0 \setminus \{i_t\}} \vartheta_i \\
 & \text{s.t.} \quad (1) - (9), (15), (16)
 \end{array} \tag{17}$$

5 Algorithmic framework

During our preliminary experiments, we observed that the formulation tends to have a large number of multi-drop constraints which makes it difficult to solve in one go using commercial solvers. Even building the model takes several minutes for instances with a relatively large number of customers and pallets with nonidentical dimensions. Therefore, we develop a branch-and-cut framework where the multi-drop constraints are only activated when they are violated. In order to further reduce the size of the mathematical model, we implement a few preprocessing procedures. These procedures aim to reduce the number of variables and obtain strong lower bounds to accelerate the branch-and-cut procedure. Section 5.1 explains the details of these preprocessing components.

5.1 Preprocessing procedures

The preprocessing stage starts by setting $l^G = GCD$, which is the greatest common divisor of pallet dimensions in the given instance. This is followed by the creation and computation of grid structure-related parameters introduced in Section 4, elimination of infeasible pallet-position combinations and computation of trivial lower bounds.

5.1.1 Eliminating infeasible pallet-position combinations

This procedure reduces the number of variables and constraints by eliminating pallet-position combinations that do not leave enough space for the successor pallets unless the multi-drop constraints are violated. Suppose pallet p is positioned L -ways on grid square g . Clearly, no pallets in $P_{c(p)}^{fol}$ can be positioned in COR_{pg}^L . Let $\overline{G}_{pg}^L = \{g_1 \in G \setminus COR_{pg}^L\}$. This procedure eliminates g from G_p^L if the total area of \overline{G}_{pg}^L is smaller than the total bottom area of all pallets in $P_{c(p)}^{fol}$. Similarly, g is removed from G_p^S if $\overline{G}_{pg}^S = \{g_1 \in G \setminus COR_{pg}^S\}$ cannot accommodate all the pallets in $P_{c(p)}^{fol}$.

5.1.2 Trivial lower bounds

Let us denote the optimal objective value of FP as z . One can obtain a lower bound, say $\underline{\vartheta}_i^0$, on the ϑ_i value by calculating the total bottom area of all pallets to be delivered to the successors of $i \in T^0 \setminus \{i_t\}$. Clearly, $\underline{z}^0 = \sum_{i \in T^0 \setminus \{i_t\}} \underline{\vartheta}_i^0$ is a lower bound for z . Let $\underline{\vartheta}^0$ denote the vector of $\underline{\vartheta}_i^0$ for $i \in T^0 \setminus \{i_t\}$, Algorithm 1 provides pseudocode of the procedure for obtaining \underline{z}^0 and $\underline{\vartheta}^0$.

Once these lower bounds are obtained, we utilize them within our formulation by including the following inequalities (18) and $(Eq(\underline{z}))$. We label Constraints $(Eq(\underline{z}))$ as a function of \underline{z} where

Algorithm 1 Trivial lower bounds

```

1:  $\underline{z}^0 = 0; \underline{\vartheta}^0 = \mathbf{0};$ 
2: for  $i \in T^0 \setminus \{i_t\}$  do
3:    $totalArea = 0;$ 
4:   for  $p \in P_i^{fol}$  do
5:      $totalArea = totalArea + l_p^S \times l_p^L;$ 
6:    $\underline{\vartheta}_i^0 = \left\lceil \frac{totalArea/l^{VS}}{l^G} \right\rceil \times l^G;$ 
7:    $\underline{z}^0 = \underline{z}^0 + \underline{\vartheta}_i^0;$ 
8: return  $\underline{z}^0, \underline{\vartheta}^0;$ 

```

\underline{z} is an input for the RHS. For instance, $(Eq(500))$ corresponds to $\sum_{i \in T^0 \setminus \{i_t\}} \vartheta_i \geq 500$. Note that Constraints (18) dominate Constraints (7) and hence can replace them.

$$\vartheta_i \geq \underline{\vartheta}_i^0, \quad \forall i \in T^0 \setminus \{i_t\} \quad (18)$$

$$\sum_{i \in T^0 \setminus \{i_t\}} \vartheta_i \geq \underline{z}. \quad (Eq(\underline{z}))$$

In order to ease the presentation of the algorithmic procedures throughout the remainder of the paper, we introduce the following notation. Let A be any mathematical programming formulation with X^A being its feasible region, then $A(\underline{z})$ corresponds to the same formulation with the restricted feasible region $X^A \cap (18) \cap (Eq(\underline{z}))$. With this notation clarified, Section 5.2 introduces two problem relaxations to obtain potentially improved lower bounds.

5.2 Problem relaxations for improved lower bounds

We first present a formulation (RP1) for Relaxation 1 where both axle weight and multi-drop constraints are excluded (CSP). This formulation utilizes the trivial lower bound \underline{z}^0 and variable bound vector $\underline{\vartheta}^0$. We denote the optimal value obtained from Relaxation 1 as \underline{z}^1 .

$$\begin{aligned}
 RP1(\underline{z}^0) \quad & \min & (17) \\
 & \text{s.t.} & (1) - (6), (18) \\
 & & \sum_{i \in T^0 \setminus \{i_t\}} \vartheta_i \geq \underline{z}^0
 \end{aligned}$$

Secondly, RP2 is a formulation for Relaxation 2 where only multi-drop constraints are excluded (CSP-AW). Relaxation 1 is also a relaxation of Relaxation 2. Thus, \underline{z}^1 is a lower bound for the optimal value of Relaxation 2. We denote the optimal value obtained from Relaxation 2 as \underline{z}^2 .

$$\begin{aligned}
 RP2(\underline{z}^1) \quad & \min & (17) \\
 & \text{s.t.} & (1) - (6), (15), (16), (18) \\
 & & \sum_{i \in T^0 \setminus \{i_t\}} \vartheta_i \geq \underline{z}^1
 \end{aligned}$$

Clearly, $\underline{z}^0 \leq \underline{z}^1 \leq \underline{z}^2 \leq z$.

5.3 Solving FP via a branch-and-cut procedure

As mentioned earlier, FP becomes too large for commercial solvers due to the multi-drop constraints. In this section, we describe a branch-and-cut procedure to solve FP . This procedure can be utilized for solving any model with multi-drop constraints by initially excluding them and adding these constraints on the fly only if violated. Therefore, in Algorithm 2, we present generic pseudocode for solving an arbitrary given model $A(\mathbf{y}, \vartheta)$ by employing such a branch-and-cut procedure $BC[A(\mathbf{y}, \vartheta)]$. The algorithm works using a traditional branch-and-bound. For each integer node of the branch-and-bound tree, the algorithm checks if the multi-drop constraints (8) are violated (line 4). If they are, Constraints (8) are added to the model (line 5). Afterwards, the algorithm checks if Constraints (9) are violated (line 6). If they are, Constraints (9) are added to the model (line 7).

Algorithm 2 Branch-and-cut procedure $BC[A(\mathbf{y}, \vartheta)]$

- 1: Start a traditional branch-and-bound (B&B) for $A(\mathbf{y}, \vartheta)$;
 - 2: **for** each integer feasible B&B node $(\bar{\mathbf{y}}, \bar{\vartheta})$ **do**
 - 3: **for** $p \in P, g \in G_p^L$ **do**
 - 4: **if** $\sum_{\substack{h \geq g: \\ h \in CO R_{pg}^L}} \sum_{p_1 \in P_{c(p)}^{fol}} \bar{x}_{p_1g} \geq \sum_{p_1 \in P_{c(p)}^{fol}} n_{p_1}(1 - \bar{y}_{pg}^L)$ **then**
 - 5: add $\sum_{\substack{h \geq g: \\ h \in CO R_{pg}^L}} \sum_{p_1 \in P_{c(p)}^{fol}} x_{p_1g} \leq \sum_{p_1 \in P_{c(p)}^{fol}} n_{p_1}(1 - y_{ph}^L)$;
 - 6: **if** $\sum_{\substack{h \geq g: \\ h \in CO R_{pg}^S}} \sum_{p_1 \in P_{c(p)}^{fol}} \bar{x}_{p_1g} \geq \sum_{p_1 \in P_{c(p)}^{fol}} n_{p_1}(1 - \bar{y}_{pg}^S)$ **then**
 - 7: add $\sum_{\substack{h \geq g: \\ h \in CO R_{pg}^S}} \sum_{p_1 \in P_{c(p)}^{fol}} x_{p_1g} \leq \sum_{p_1 \in P_{c(p)}^{fol}} n_{p_1}(1 - y_{ph}^S)$;
 - 8: **return** $z^*, (\mathbf{y}^*, \vartheta^*)$;
-

5.4 General algorithmic framework

The general algorithmic framework that we develop consists of three main steps following the preprocessing procedures which were detailed in Section 5.1. Step 1 solves Relaxation 1 model $RFP1(\underline{z}^0)$ that provides lower bound \underline{z}^1 and solution $(\mathbf{y}^1, \vartheta^1)$. If this solution violates any axle weight or multi-drop constraint, then the algorithm proceeds to Step 2; otherwise, the algorithm terminates since $(\mathbf{y}^1, \vartheta^1)$ is an optimal solution for the CSP-MD-AW. Step 2 solves Relaxation 2 model $RFP2(\underline{z}^1)$ that provides lower bound \underline{z}^2 and solution $(\mathbf{y}^2, \vartheta^2)$. If this solution violates any multi-drop constraints, then the algorithm proceeds to Step 3. In the case of no violation, the algorithm terminates and returns $(\mathbf{y}^2, \vartheta^2)$ as an optimal solution for the CSP-MD-AW. Step 3 solves $FP(\underline{z}^2)$ using the branch-and-cut procedure detailed in Section 5.3, which at termination provides an optimal solution for the CSP-MD-AW. Algorithm 3 presents pseudocode of the general algorithmic framework.

Algorithm 3 General algorithmic framework

```
1: Step 0: Preprocessing procedure;  
2:   Variable elimination;  
3:   Trivial lower bound  $\rightarrow \underline{z}^0$ ;  
4: Step 1: Solve relaxation  $RFP1(\underline{z}^0) \rightarrow \underline{z}^1, (\mathbf{y}^1, \vartheta^1)$ ;  
5: if  $(\mathbf{y}^1, \vartheta^1)$  violates (8),(9),(15), or (16) then  
6:   Go to Step 2;  
7: else  
8:    $(\mathbf{y}^*, \vartheta^*) \leftarrow (\mathbf{y}^1, \vartheta^1)$ ;  
9:    $\underline{z}^* \leftarrow \underline{z}^1$ ;  
10:  Go to Step 4;  
11: Step 2: Solve relaxation  $RFP2(\underline{z}^1) \rightarrow \underline{z}^2, (\mathbf{y}^2, \vartheta^2)$ ;  
12: if  $(\mathbf{y}^2, \vartheta^2)$  violates (8) or(9) then  
13:   Go to Step 3;  
14: else  
15:    $(\mathbf{y}^*, \vartheta^*) \leftarrow (\mathbf{y}^2, \vartheta^2)$ ;  
16:    $\underline{z}^* \leftarrow \underline{z}^2$ ;  
17:   Go to Step 4;  
18: Step 3: Solve  $FP(\underline{z}^2)$  via  $BC[FP(\underline{z}^2)] \rightarrow \underline{z}^*, (\mathbf{y}^*, \vartheta^*)$ ;  
19: Step 4: Return  $\underline{z}^*, (\mathbf{y}^*, \vartheta^*)$ ;
```

6 Computational study

We implemented the MIP and the algorithm in a Java environment where we call IBM ILOG CPLEX 12.8 to solve the mathematical models. Experiments were run on a server with $2 \times$ Intel® Xeon® E5-2660 v3 processor @2.6GHz, 20 cores and 160GB RAM. For each mathematical model, CPLEX is terminated if proven optimality is not reached within one hour.

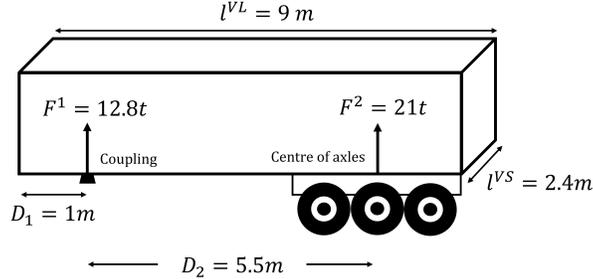


Figure 11: Container dimensions utilized in the experiments.

We conducted experiments for a single vehicle type with two axles and the container dimensions indicated in Figure 11. Instances contain three standard European pallet sizes: $0.8\text{ m} \times 1.2\text{ m}$, $1\text{ m} \times 1.2\text{ m}$ and $0.8\text{ m} \times 0.6\text{ m}$. Pallet weights are randomly generated and have values ranging from 50kg to 2.4t (max 1.2t for the smallest pallet type). We also generated a few artificial instances with very heavy pallets, which in reality may not be considered a safe pallet load, but can provide useful theoretical insights. The minimum grid square dimension l^G we utilized in the experiments is equal to 0.2m (the greatest common divisor of pallet dimensions). This enables us to round down the container surface dimensions to 2.4m and 9m which are originally 2.45m and 9.12m. The methods are valid for any container dimension of the same vehicle type.

The number of pallets $|P|$ per instance ranges from 7 to 40 while the number of customers $|T|$ ranges from 2 to 10. Depending on whether the considered pallets have identical sizes and weights

or not, instances can be classified into five categories as detailed in Table 1. We further categorize instances based on average weight per unit pallet area $\omega = \sum_{p \in P} w_p / (\sum_{p \in P} l_p^S \times l_p^L)$, as indicated in Table 2.

Table 1: Instance categories based on identical or nonidentical size and weight.

Category	Among all pallets	Among the same customer pallets	
	size	size	weight
III	Identical	Identical	Identical
IIN	Identical	Identical	Non-identical
NNN	Non-identical	Non-identical	Non-identical
NII	Non-identical	Identical	Identical
NIN	Non-identical	Identical	Non-identical

Table 2: Instance categories based on average weight per unit pallet area.

Weight category	Average weight per unit pallet area (ω)
L (low)	$\omega < 1000kg/m^2$
M (medium)	$1000kg/m^2 \leq \omega < 1300kg/m^2$
H (high)	$1300kg/m^2 \leq \omega$

The instance files are available online at doi:10.17632/4zhjc3sj8d.1. The first line of each instance file includes the following information: the number of pallets $|P|$ and the number of customers $|T|$. This line is followed by $|P|$ lines, each of which is dedicated to a pallet and contains the pallet index p , customer $c(p)$, dimensions l_p^L , l_p^S and weight w_p . Further details of instance categories are provided in B where each instance is analyzed in terms of average pallet weight, area and the number of each pallet type.

6.1 Methodological and computational insights for the theoretical problem

In order to provide a brief overview of the computational performance of our algorithm, Tables 3-5 summarize the average solving times and the number of instances for which Steps 2 and 3 of Algorithm 3 were activated. In these tables, the first column indicates the instance category and file ID that includes information concerning the number of pallets and customers as well as pallet dimension and weight categories. When there are multiple instances of the same category with different pallet-customer configurations, this is indicated by including an additional number at the end of the Category-ID (for example IIN-P20C4H and IIN-P20C4H2). Columns 2 and 3 contain the number of pallets and customers in the corresponding instance file. Column ‘# T ’ gives the number of distinct customer sequences for which the corresponding customer-pallet configuration has been investigated. Columns ‘ $t_{avg}^1(s)$ ’, ‘ $t_{avg}^2(s)$ ’ and ‘ $t_{avg}^3(s)$ ’ provide the average solving time (in seconds) for RFP1 (including preprocessing procedures), RFP2 and BC, respectively, while $t_{avg}(s)$ is the average total solving time over all considered customer sequences. As RFP2 and BC steps are only activated when needed, columns ‘# RFP2’ and ‘# BC’ report the number of sequences for which RFP2 and BC steps were activated, respectively. It is also important to note that $t_{avg}^2(s)$ and $t_{avg}^3(s)$ are calculated only over those instances where RFP2 and BC steps were employed, respectively. For example, if BC is activated only for one instance out of 120 and its BC time is 3600, then $t_{avg}^3(s) = 3600$ and not $3600/120 = 30$.

Table 3 summarizes the results obtained for 978 instances with up to 24 pallets and 5 customers. Relaxation RFP2 and BC were needed for 552 and 353 instances, respectively. The average solving

Table 3: Average computing time over all sequences for each step of Algorithm 3.

Category-ID	$ P $	$ T $	$\# T$	$t_{avg}^1(s)$	$\#$ RFP2	$t_{avg}^2(s)$	$\#$ BC	$t_{avg}^3(s)$	$t_{avg}(s)$
NIN-P7C3M	7	3	6	0.44	0		0		0.44
NII-P7C3M	7	3	6	0.50	0		0		0.50
NNN-P7C3M	7	3	6	0.67	0		0		0.67
NNN-P10C4L	10	4	24	1.95	1	23.99	0		2.95
NNN-P10C4M	10	4	24	1.86	1	23.99	0		2.85
NNN-P12C3M	12	3	6	32.73	0		0		32.73
NNN-P12C3H	12	3	6	1.37	6	21.97	3	3604.70	1825.69
NNN-P12C3H2	12	3	6	27.63	6	21.97	1	3.12	50.12
III-P14C2H	14	2	2	0.73	2	1.13	1	3602.78	1803.24
NNN-P14C2L	14	2	2	3.06	0		0		3.06
NNN-P14C3L	14	3	6	6.50	0		0		6.50
NNN-P14C4L	14	4	24	10.45	7	21.95	4		16.85
NNN-P14C5L	14	5	120	19.95	38	24.41	13	1226.82	160.58
NNN-P14C5M	14	5	120	9.95	65	24.95	34	1561.75	465.96
NNN-P16C2M	16	2	2	14.16	0		0		14.16
NNN-P16C3M	16	3	6	9.35	0		0		9.35
NNN-P16C4M	16	4	24	27.31	5	22.57	1	1356.61	88.53
NNN-P18C4M	18	4	24	47.48	0		0		47.48
NNN-P18C5M	18	5	120	39.51	105	114.14	77	2416.86	1690.21
III-P20C2H	20	2	2	0.75	2	14.50	2	3600.13	3615.37
III-P20C4H*	20	4	4	0.53	4	21.65	4	2701.67	2723.85
IIN-P20C4H	20	4	24	0.36	13	23.82	8	3562.91	1200.90
IIN-P20C4H2	20	4	24	0.38	20	23.78	15	3600.39	2270.44
IIN-P22C5H*	22	5	22	1.11	22	23.36	22	2951.27	2975.74
NNN-P23C3M	23	3	6	105.56	3	21.97	2	1930.41	760.02
NNN-P23C5M	23	5	120	115.83	104	24.80	81	2695.88	1957.05
NNN-P24C2H	24	2	2	24.53	2	14.50	1	3600.00	1839.03
NNN-P24C5M	24	5	120	16.46	52	24.95	22	1208.75	248.87
NNN-P24C5H	24	5	120	16.01	94	25.04	62	3439.06	1812.47
Total			978		552		353		
Average				18.52		25.97		2533.13	883.64

*The batch process of experiments was halted once a feasible solution was not found for a sequence.

times for RFP1 (including the preprocessing) and RFP2 are both under 26 seconds, which indicates that the models are solved very fast without the multi-drop constraints for these instances. Although the addition of multi-drop and axle weight constraints in combination makes the problem more difficult to solve, the overall average solving time (including all algorithmic steps) is 884 seconds (~ 15 minutes). Note that we ran experiments for all sequence permutations of each category in a separate batch. For a few instance categories the batch process was halted when no feasible solution was found for a sequence, with no experiments run for the remaining sequences. Although we did not visualize all of these solutions, our implementation creates a readily-executable file for the solution obtained at each step of Algorithm 3, which can be provided upon request. These files then can be compiled to visualize the solutions.

In order to observe the performance of Algorithm 3 for instances with a greater number of pallets, we created instances with 30 and 40 pallets and 5 customers with two configurations per category. Note that the container we consider cannot accommodate more than 44 of the 0.8×0.6 pallets and no more than 40 pallets if at least two are of a larger size category. Table 4 summarizes the results for these larger instances. We observe that the average solving times for RP1 and especially for RP2

are much much longer than those for the instances in Table 3 in Table 3, which were associated with fewer pallets. The BC solving times are also quite long, which was already observed for many of the instance categories in Table 3. Tables 3 and 4 are in fact good indicators of how the axle weight and multi-drop constraints increase the complexity of the problem. The last group of instance

Table 4: Instances with 30 and 40 pallets.

Category-ID	$ P $	$ T $	$\# T$	$t_{avg}^1(s)$	$\# RFP2$	$t_{avg}^2(s)$	$\# BC$	$t_{avg}^3(s)$	$t_{avg}(s)$
NNN-P30C5H	30	5	120	105.21	53	1061.07	36	3494.69	1622.26
NNN-P30C5H2	30	5	120	92.61	86	1275.92	55	2943.42	2356.08
NNN-P40C5H*	40	5	61	186.64	42	1232.37	26	3345.33	2461.04
NNN-P40C5H2*	40	5	3	274.07	3	1902.74	3	3600.14	5776.95
Total			304		184		120		
Average				164.63		1368.03		3345.89	3054.08

*The batch process of experiments for all sequence permutations is halted once a feasible solution is not found for a sequence.

categories we considered contain 10 customers and 16 to 40 pallets. We test these instance categories for a single sequence $\{c_1, c_2, \dots, c_{10}\}$ since testing the complete permutation would involve millions of sequences. Although it is difficult to draw general conclusions, we observe that Algorithm 3 could not find a feasible solution for two of the nine instances within the time limit. The computation times for RP2 are again longer than the average value reported in Table 3.

Table 5: Instances with 10 customers.

Category-ID	$ P $	$ T $	$\# T$	$t_{avg}^1(s)$	$\# RFP2$	$t_{avg}^2(s)$	$\# BC$	$t_{avg}^3(s)$	$t_{avg}(s)$
NNN-P16C10M	16	10	1	9.89	0		0		9.89
NNN-P18C10M	18	10	1	31.61	0		0		31.61
IIN-P20C10H	20	10	1	0.3	0		0		0.30
IIN-P22C10H	22	10	1	0.92	1	40.69	1	3600.00**	3641.61
NNN-P24C10H	24	10	1	51.08	1	89.76	1	612.41	753.25
NNN-P30C10H	30	10	1	40.75	1	117.33	0		158.08
NNN-P30C10H2	30	10	1	316.07	1	408.60	0		724.67
NNN-P40C10H	40	10	1	127.06	1	775.10	0		902.16
NNN-P40C10H2	40	10	1	145.99	1	3600.28**		3600.00**	7346.27
Total			9		6		3		
Average				80.41		838.63		2604.14	1507.54

**No feasible solution is found.

While Algorithm 3 was unable to converge to proven optimality or find a feasible solution for some instances, one may question the performance of the mixed integer programming formulation FP as a one-shot method. We therefore solved a large subset of the instance categories using FP and compared the solving times in Table 6. While Algorithm 3 outperforms FP for a large majority of the categories, FP is competitive for a few. For category III-P20C4H, none of the methods could find a feasible solution for sequence $\{c_3, c_2, c_1, c_4\}$ and FP was unable to find a feasible solution for one more instance. FP is not able to find any feasible solution for instances NNN-P30C10H, NNN-P40C10H and NNN-P40C10H2 and it cannot find an optimal solution for NNN-P30C10H2 within the time limit.

Table 6: Computational comparison of *FP* and Algorithm 3.

Category-ID	$ P $	$ T $	$\# T$	$t_{avg}^1(s)$	$\# \text{RFP2}$	$t_{avg}^2(s)$	$\# \text{BC}$	$t_{avg}^3(s)$	$t_{avg}(s)$	$t_{avg}^{MIP}(s)$
NIN-P7C3M	7	3	6	0.44	0		0		0.44	38.98
NII-P7C3M	7	3	6	0.50	0		0		0.50	31.89
NNN-P7C3M	7	3	6	0.67	0		0		0.67	31.97
NNN-P10C4L	10	4	24	1.95	1	23.99	0		2.95	133.42
NNN-P10C4M	10	4	24	1.86	1	23.99	0		2.85	128.43
NNN-P12C3M	12	3	6	32.73	0		0		32.73	311.46
NNN-P12C3H	12	3	6	1.37	6	21.97	3	3604.70	1825.69	2544.01
NNN-P12C3H2	12	3	6	27.63	6	21.97	1	3.12	50.12	660.47
III-P14C2H	14	2	2	0.73	2	1.13	1	3602.78	1803.24	1802.39
NNN-P14C2L	14	2	2	3.06	0		0		3.06	251.51
NNN-P14C3L	14	3	6	6.50	0		0		6.50	639.60
NNN-P14C4L	14	4	24	10.45	7	21.95	4		16.85	923.00
NNN-P14C5L	14	5	120	19.95	38	24.41	13	1226.82	160.58	1776.05
NNN-P14C5M	14	5	120	9.95	65	24.95	34	1561.75	465.96	1342.35
NNN-P16C2M	16	2	2	14.16	0		0		14.16	730.16
NNN-P16C3M	16	3	6	9.35	0		0		9.35	942.28
NNN-P16C4M	16	4	24	27.31	5	22.57	1	1356.61	88.53	2137.29
NNN-P18C4M	18	4	24	47.48	0		0		47.48	3456.81
NNN-P18C5M	18	5	120	39.51	105	114.14	77	2416.86	1690.21	3415.49**
III-P20C2H	20	2	2	0.75	2	14.50	2	3600.13	3615.37	3601.46
III-P20C4H	20	4	4	0.53	4	21.65	4	2701.67	2723.85	3600.00**
IIN-P20C4H	20	4	24	0.36	13	23.82	8	3562.91	1200.90	1083.80
NNN-P23C3M	23	3	6	105.56	3	21.97	2	1930.41	760.02	3320.76
NNN-P24C2H	24	2	2	24.53	2	14.50	1	3600.00	1839.03	2207.26
Total			572		260		151			
Average				16.14		26.50			681.71	1462.95

**No feasible solution found within the time limit for some sequences.

We compare the performance of *FP* and algorithm on four specific instances in more detail in Table 7, for which both methods terminate prematurely (before proven optimality) due to the limit. *FP* provides better feasible solutions, whereas Algorithm 3 provides better lower bounds thanks to the RP2 solution.

Table 7: Lower and upper bound comparison of *FP* and Algorithm 3 for a selection of instances.

Category-ID	$ P $	$ T $	T	BC LB	BC UB	MIP LB	MIP UB
III-P20C2H	20	2	$\{c_2, c_1\}$	1440.00	1680	1356.71	1560
III-P20C2H	20	2	$\{c_1, c_2\}$	1280.00	1640	1252.14	1360
IIN-P20C4H2	20	4	$\{c_3, c_2, c_1, c_4\}$	2160.00	2440	2073.54	2200
IIN-P20C4H2	20	4	$\{c_3, c_2, c_4, c_1\}$	2160.00	2640	2083.01	2280

The following list summarizes our observations regarding the computational performance of Algorithm 3 and *FP*:

- Instances with low weights where the axle weight constraints are not binding can be solved in a few seconds using our algorithm. More specifically, among the instances considered, those from the low- or medium-weight category with up to 16 pallets and 3 customers could be solved in less than 8 seconds on average. Instances from the H category can prove computationally challenging, even with 12 pallets and 3 customers or 14 pallets and 2 customers.
- Another factor that increases the computational difficulty in solving the problems is the number of customers. We observe this most clearly for the NNN-P14-L and NNN-P16-M instances,

but also for the NNN-P18-M and NNN-P23-M instances.

- When all the pallets have identical dimensions, Algorithm 3 struggles in converging to prove optimality. This is due to very high symmetry regarding the positioning of pallets, which leads to a high number of solutions that violate multi-drop constraints while trying to satisfy the axle weight constraints. Note that without the axle weight constraints, a solution that satisfies the multi-drop constraints is often obtained relatively quickly.
- Instances with a larger number of pallets are not always more challenging than those with fewer pallets. However, on average they do indeed require more computational effort.
- Solving times may differ significantly for different customer sequences, even those sharing identical pallet-customer configurations. Algorithm 3 may not find a feasible solution for some sequences whereas for others an optimal solution can be obtained rather quickly. If the pallets of each customer can be perfectly packed independently without any empty spaces between pallets and walls (front and side-walls) or between pallets themselves, the problem is likely to be solved very quickly if the given sequence permits such perfect packing schemes without violating the axle weight restrictions. If not, the algorithm may spend a large amount of time eliminating all the solutions violating multi-drop constraints while satisfying axle weight constraints with a packing scheme without any empty space between pallets. Therefore, the customer sequence plays a very important role.
- The trivial lower bounds on ϑ_i variables are often helpful in obtaining solutions which respect multi-drop constraints without actually activating these constraints.
- Many solutions obtained in Step 1 of Algorithm 3 respect the multi-drop constraints even if they violate the axle weight constraints.
- *FP* may be a good alternative to obtain higher quality solutions when the BC terminates prematurely. In such cases, one could also consider switching to *FP* as an additional framework step by utilizing the bounds obtained from the BC.
- We also implemented Algorithm 3 by using the strengthened multi-drop constraints introduced in A. We did not observe a variant whose computational performance dominates the one with Constraints (8) and (9).

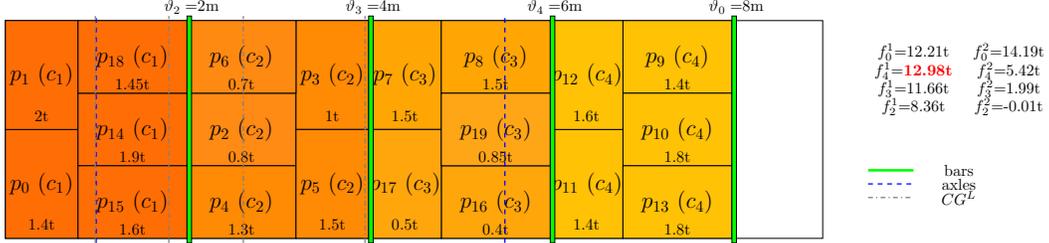
6.2 Managerial insights for the practical problem

We provide a meticulous analysis of problem features based on a careful selection of solution characteristics. The visualizations are restricted to a reasonable number to avoid a lengthy paper, yet sufficient to demonstrate all important findings.

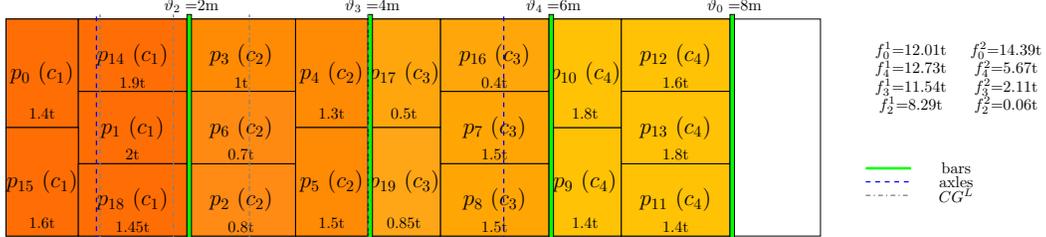
6.2.1 Impact of the axle weight constraints

The purpose of this section is to analyze through examples how axle weight constraints may significantly impact the solutions. Section 4 provided an example where axle weight restrictions would lead to an optimal solution (Figure 6) whose objective value is strictly worse than the optimal value when axle weight restrictions are ignored (Figure 5). The solution depicted in Figure 6 required filling the empty space left behind the bar after unloading the pallets of c_2 , whereas this was not the case in Figure 5.

Figure 12 compares two loading schemes with the same objective value for another problem instance with 4 customers. Each customer is associated with 5 pallets of the same size but different



(a) A loading scheme violating the axle weight constraints.



(b) An alternative and optimal loading scheme respecting the axle weight constraints.

Figure 12: IIN-P20C4H2, $T = \{c_4, c_3, c_2, c_1\}$.

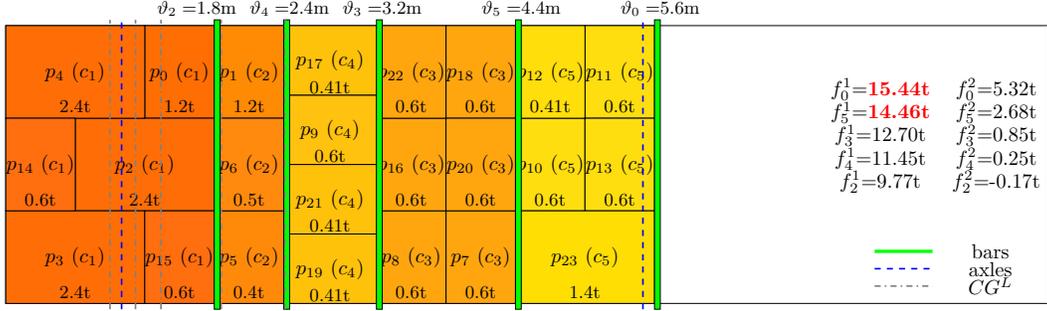
weights. Figure 12a illustrates an optimal loading scheme which ignores the axle weight restrictions. The customers must be served in the order $T = \{c_4, c_3, c_2, c_1\}$. In this loading scheme, the axle weight limits are exceeded when travelling from customer c_4 to customer c_2 and hence this is not a feasible solution for the CSP-MD-AW. For the same problem instance, an alternative loading scheme given in Figure 12b respects the axle weight limits throughout the entire trip. Given that it also respects the multi-drop constraints, this loading scheme is optimal for the CSP-MD-AW. Note that for a particular $i \in T$, the ϑ_i value is identical in Figures 12a and 12b. Although the intuitive placement of blocking bars is optimal for this instance, the arrangement of pallets behind the bar plays a crucial role. A complete enumeration of such different arrangements would lead to a large number of distinct loading schemes, hence would be impractical.

Another example where the axle weight constraints lead to an optimal solution which is unintuitive for human practitioners can be observed in Figure 13. If one would ignore the axle weight limits, an optimal solution would look like Figure 13a. Such a loading scheme would not require any cushions and securing could be performed using blocking bars only. This type of loading scheme can consequently be considered an ‘*ideal*’ one in practice. However, it is not always possible to find an ‘*ideal*’ loading scheme, especially due to the multi-drop and axle weight constraints. In fact, the solution in Figure 13a violates the axle weight limits and must be prevented in reality. The optimal solution given by CPLEX respecting the axle weight constraints is depicted in Figure 13b. However, since this solution requires inflating several cushions, it is unlikely to be employed by the practitioners. Therefore, it may be necessary to further look for alternative optimal or feasible solutions.

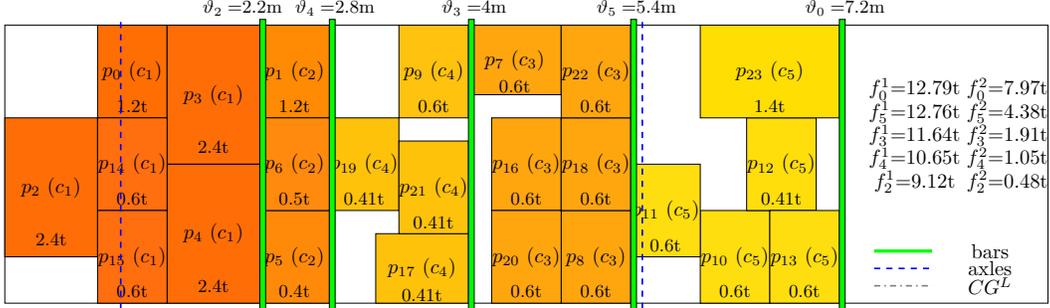
6.2.2 Impact of the multi-drop constraints

The multi-drop constraints help minimize the additional movements of pallets at customer nodes. We now investigate the impact of ignoring the multi-drop constraints.

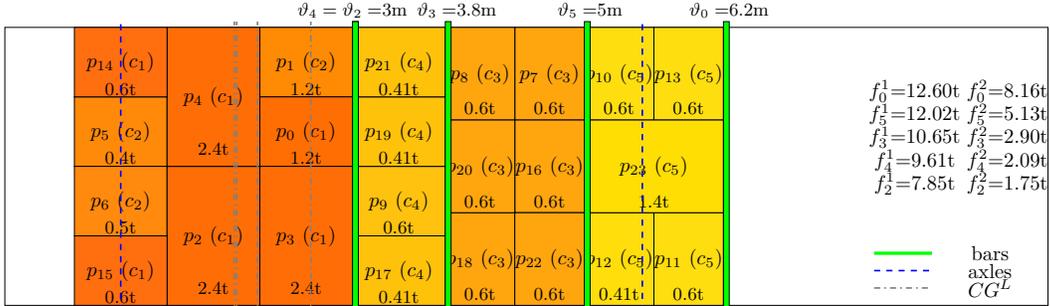
Recall the optimal loading schemes which respect the multi-drop constraints while also taking into account axle weight restrictions (Figure 13a) and when ignoring them (Figure 13b). When



(a) An optimal loading scheme with $\sum \vartheta = 17.4m$ when the axle weight restrictions are ignored.



(b) An optimal loading scheme with $\sum \vartheta = 21.6m$ when axle weight restrictions are enforced.



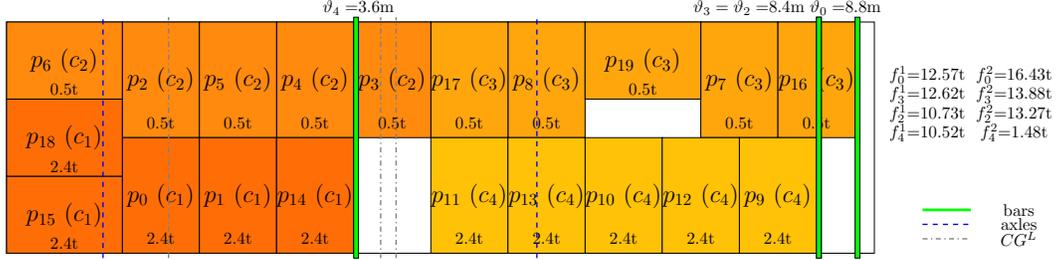
(c) The best solution provided by CPLEX when multi-drop constraints are ignored.

Figure 13: NNN-P24C5H, $T = \{c_5, c_3, c_4, c_2, c_1\}$.

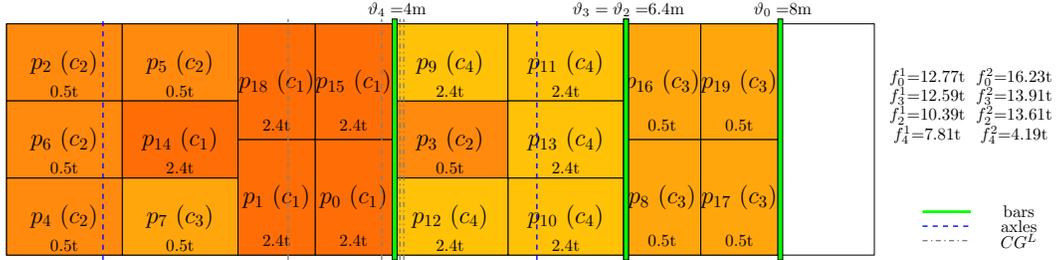
the axle weight restrictions are enforced without the multi-drop constraints, we obtain the loading scheme depicted in Figure 13c. Although this is a loading scheme with fewer empty spaces to be filled, it is infeasible since pallets p_0 , p_2 , p_3 and p_4 must be unloaded and reloaded at customer c_2 in order to be able to unload pallets p_5 and p_6 .

Figures 14a and 14b depict the optimal loading schemes for another instance when the multi-drop constraints are respected and ignored, respectively. In Figure 14a, the pallets of each customer are aligned on one side (either right or left) and empty spaces are needed between pallets in order to respect the axle weight limit in combination with the multi-drop constraints. Given the need for cushions, this would not be considered an *ideal* loading scheme in practice. The solution in Figure 14b packs pallets at the depot without the need for cushions, while also ensuring axle weight limits are not violated. However, it will be necessary to unload and reload six pallets (p_0 , p_1 , p_3 , p_{10} , p_{12} , p_{13}) at customer c_3 in order to unload pallet p_7 . It will then also be necessary to fill the empty

space of p_7 with a cushion. All these operations require undesirable amounts of time and effort at customer locations.



(a) An optimal loading scheme.



(b) An optimal loading scheme when the multi-drop constraints are ignored.

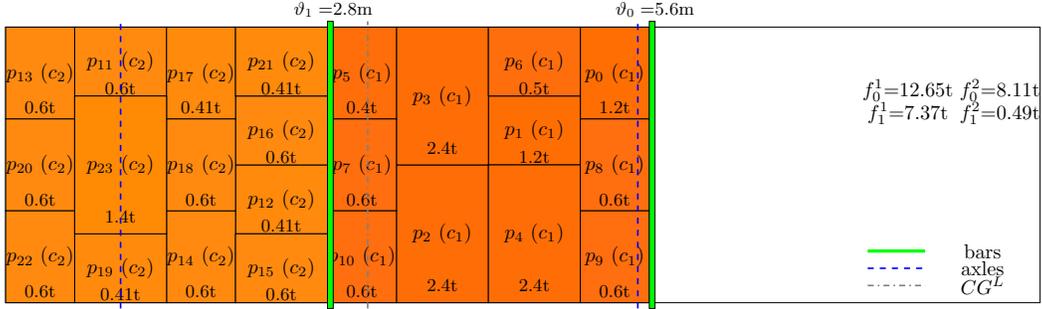
Figure 14: III-P20C4H, $T = \{c_3, c_2, c_4, c_1\}$.

6.2.3 Impact of customer sequence (the order of visits)

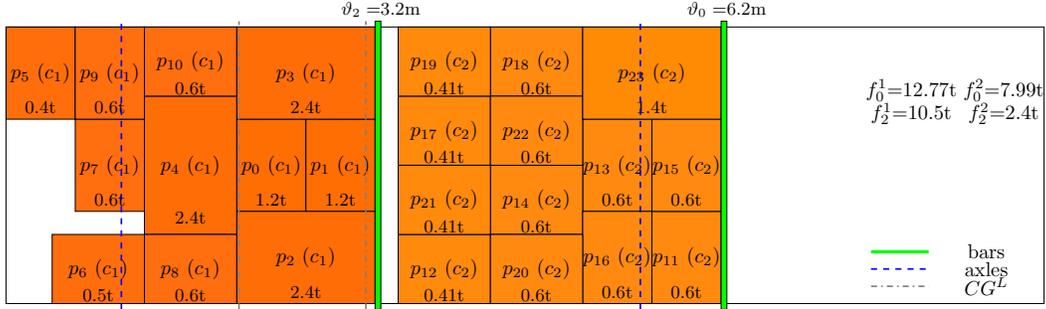
In this section, we provide examples where different customer sequences lead to significantly different solutions for the same set of customer pallets. We begin with Figures 15a and 15b, which illustrate optimal loading schemes for two problem instances whose parameters are identical except for the customer sequence. For the first instance with $T = \{c_1, c_2\}$, Figure 15a is an optimal ‘ideal’ loading scheme. However, for the instance with $T = \{c_2, c_1\}$, it is not possible to come up with such an ‘ideal’ solution due to the axle weight restrictions. It is necessary to leave certain empty spaces to be filled with cushions behind the blocking bar. These two figures should motivate managers to take into account the importance of customer visit orders, which are often given as input for the loading problems (this is also the case for the CSP-AW-MD). The order of customer visits is usually based on the total traveling distance (or time). For this particular set of customers, the total traveling distances of $T^1 = \{0, c_1, c_2, 0\}$ and $T^2 = \{0, c_2, c_1, 0\}$ would be identical if the underlying distance network was undirected. In the absence of any other criterion to prioritize the two orders, a wise managerial decision would be to employ T^1 and not T^2 .

We observe a similar situation with the instances regarding Figures 13a and 13b. Recall the optimal solution for $T = \{c_5, c_3, c_4, c_2, c_1\}$ in Figure 13b. If the visit order to these customers would be reversed as $T = \{c_1, c_2, c_4, c_3, c_5\}$, an optimal loading scheme would look like Figure 16, which would again be an ‘ideal’ solution that does not require any cushions. Note that no ideal loading scheme exists for $T = \{c_5, c_3, c_4, c_2, c_1\}$ because the axle weight limits are crucial for safety.

In Figure 17, we observe how the solutions change when the visit order of only two consecutive customers (c_5 and c_1) are swapped. Although both orders would enable geometrically ‘ideal’ loading



(a) An optimal loading scheme for $T = \{c_1, c_2\}$.



(b) An optimal loading scheme for $T = \{c_2, c_1\}$.

Figure 15: NNN-P24C2H.

schemes, as depicted in Figures 17a and 17c, the scheme in Figure 17a violates the axle weight restrictions. The best feasible solution we obtain from CPLEX is depicted in Figure 17b, which is clearly inferior to the scheme in Figure 17c.

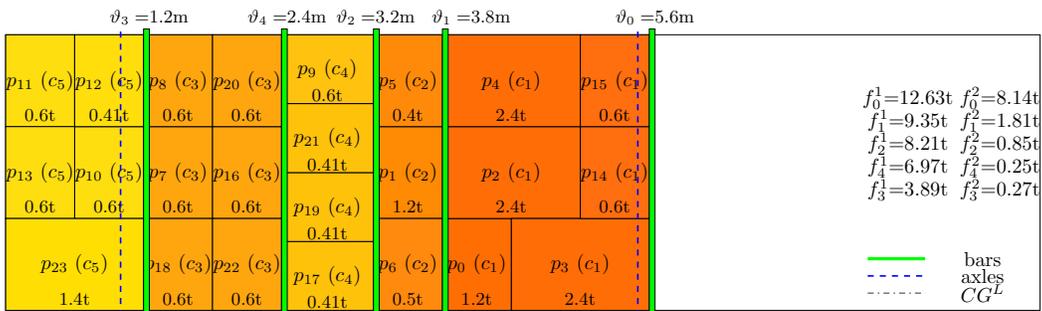
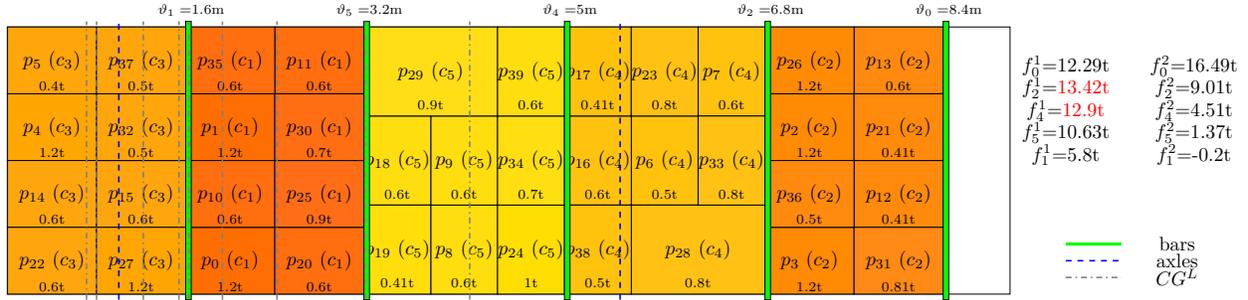
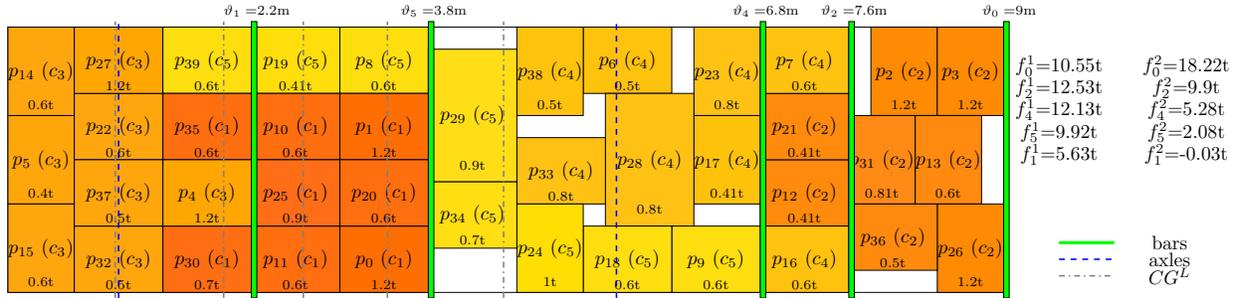


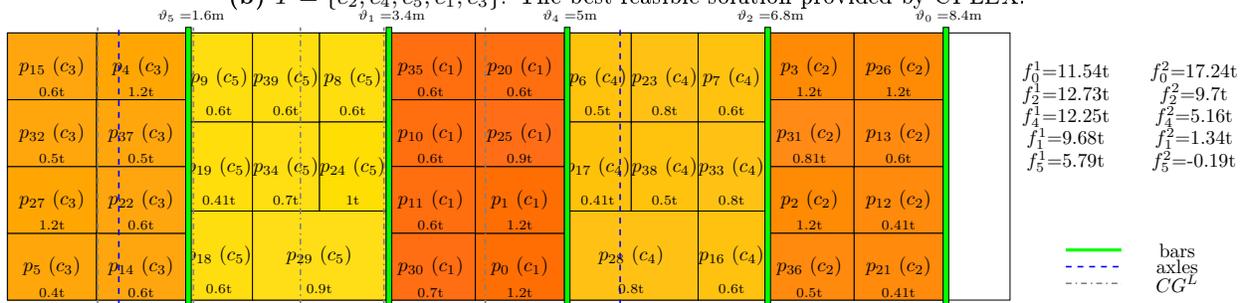
Figure 16: NNN-P24C5H, $T = \{c_1, c_2, c_4, c_3, c_5\}$: An optimal loading scheme.



(a) $T = \{c_2, c_4, c_5, c_1, c_3\}$: An optimal loading scheme when the axle weight restrictions are ignored.



(b) $T = \{c_2, c_4, c_5, c_1, c_3\}$: The best feasible solution provided by CPLEX.



(c) $T = \{c_2, c_4, c_1, c_5, c_3\}$: An optimal solution.

Figure 17: NNN-P40C5H, $T = \{c_2, c_4, c_5, c_1, c_3\}$ and $T = \{c_2, c_4, c_1, c_5, c_3\}$.

6.2.4 Impact of rotation

Another managerial decision concerns the rotation of pallets. In many cases, pallets can only be loaded in a fixed orientation. One practical and methodological advantage of this restriction is that the number of feasible loading schemes can be considerably smaller and that even complete enumeration could be possible, especially for instances with identical pallet size $1.2m \times 0.8m$ and low pallet weights.

Consider the same instance as the one associated with Figures 5 and 6. This time we allow pallets to be positioned only S -ways. If we deactivate the axle weight restrictions, an optimal solution is depicted in Figure 18a. When the axle weight constraints are enforced, the best solution we obtain from the solver is depicted in Figure 18b. The loading schemes in Figures 18a and 18b are inferior to the ones in Figures 5 and 6, respectively, not only in terms of theoretical objective value, but also

from a practical handling perspective since they will feature additional empty spaces which must be filled at customer locations.

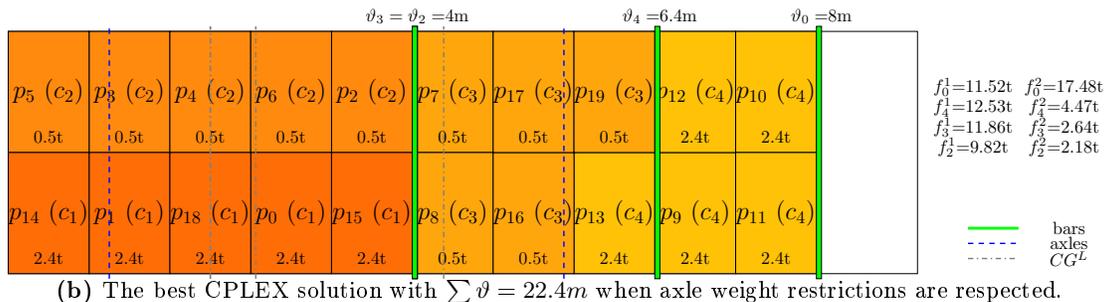
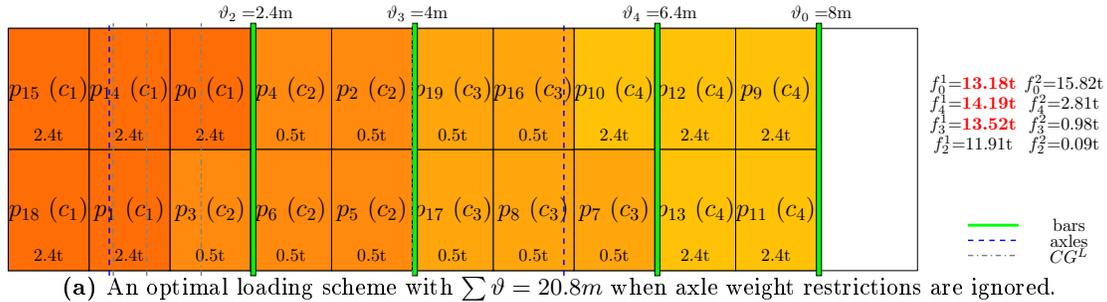


Figure 18: III-P20C4H, $T = \{c_4, c_3, c_2, c_1\}$ with S -ways placement of pallets only.

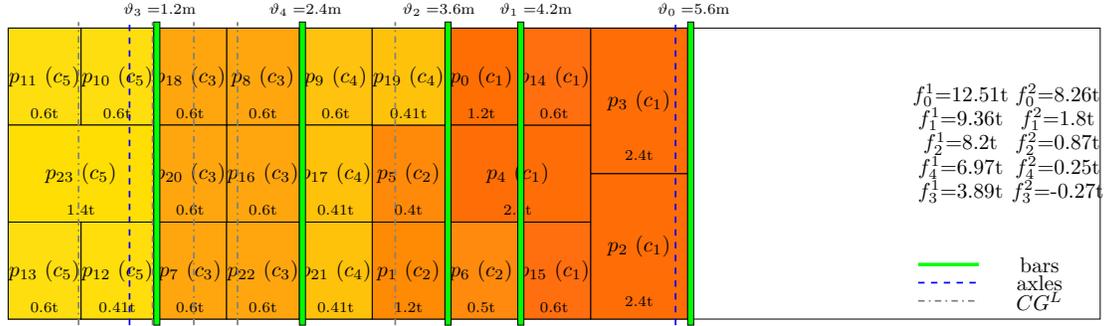
Recall the optimal loading scheme in Figure 16, which could be achieved thanks to the rotation of small pallets. Such an ‘ideal’ loading scheme would not be possible if the rotation of small pallets was restricted. Figures 19a and 19b are the optimal schemes when the small pallets can only be placed S -ways and L -ways, respectively. Due to the resulting empty spaces, both of these schemes are less desirable from a practical perspective.

6.2.5 Critical analysis on the mathematical objective function

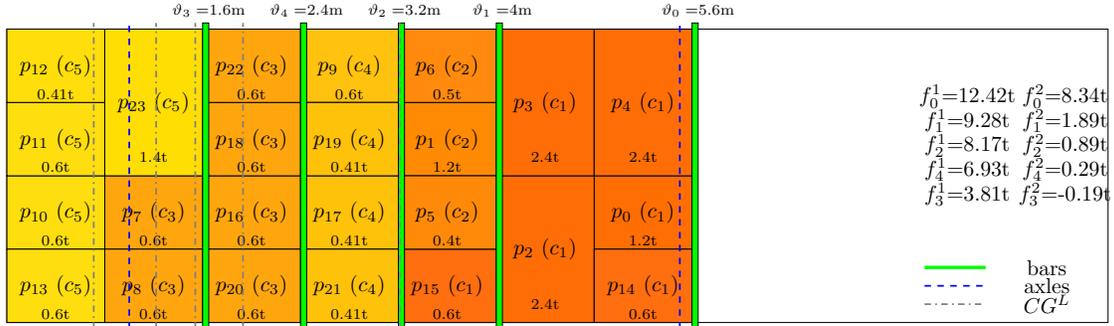
The objective function of the cargo securing problem introduced in this paper can be formulated with different mathematical equations. The challenge is to come up with ones that are computationally light and as realistic as possible. The mathematical objective function we chose in this paper satisfies these requirements to a large extent. We need only $|T|$ continuous decision variables and $O(|T| \times |P|)$ constraints to express the physical securing objective with the positioning, multi-drop, axle weight constraints and variables at hand. For certain instances, there exist several alternative optimal solutions that provide equal values for Equation (17). The solver provides us with a single optimal solution. It is likely that there are other alternative solutions that would be preferable in practice compared to the one provided by the solver.

One potential approach to overcome this drawback could be to develop a postprocessing procedure. Given an optimal solution by the solver, this procedure could arrange the pallets in accordance with the practical preferences. This could also be supported with an interactive visualization with which the human planners can rearrange the pallets and see whether the axle weight limits are respected.

Determining the best objective function for minimizing the securing cost still remains a big chal-



(a) An optimal loading scheme with only S -ways placement of small pallets.



(b) An optimal loading scheme with only L -ways placement of small pallets.

Figure 19: NNN-P24C5H, $T = \{c_1, c_2, c_4, c_3, c_5\}$.

lenge, also due to the lack of sufficient physical testing to see how and under what conditions pallets support each other. For example, in Figure 19a, the empty space between pallets p_{11}, p_{12}, p_{13} and p_{23} will most likely not require a cushion. This type of specifications should be clearly documented in manuals. However, we should note that including every specification prior to optimization is likely to lead to an unsolvable mathematical model. Hence, some of them may be kept for the postprocessing procedure.

The following is a list of important practical and managerial findings of our experimental study:

- Making wise decisions regarding customer sequences is critical in reducing securing efforts and costs.
- Axle weight and multi-drop restrictions in combination or individually may lead to undesirable empty spaces in containers which in turn bring additional costs to the system.
- Theoretically, the case with fixed orientations cannot lead to strictly better solutions. However, Algorithm 3 may terminate prematurely at a sub-optimal solution when solving the general problem within a short time limit. In such cases, it is sometimes possible to obtain better solutions from the problem with fixed orientations. Thus, solving the case with fixed orientations may be worth the computational effort if the general problem solution does not have the required quality.
- For some instances, the optimal (or the best) solution provided by the solver can further be processed with very simple intuitive moves to obtain alternative solutions that are more preferable in practice. Therefore, a postprocessing algorithm may prove useful.
- We observe that utilizing a different mathematical objective function sometimes leads to solutions that are more preferable in practice (for example Figure 14a vs minimizing ϑ_0 instead

of $\sum \vartheta$). Therefore, there is room for further investigation on modeling suitable mathematical objective functions.

7 Conclusions and future research directions

Secure cargo loading is a crucial concern in freight transportation. The dynamic and static stability of the cargo has been indicated as an essential research matter that must be tackled. Nevertheless, the literature previously lacked efficient methods to minimize securing efforts in the presence of multi-drop and axle weight constraints when loading pallets of non-identical size and weight with rotations. This paper introduced an efficient way of ensuring secure loading schemes using blocking bars and inflating cushions. The additional challenge of respecting the axle weight limitations and multi-drop restrictions encouraged us to develop a mixed integer programming formulation based on a grid-square discretization of the container surface. The large number of multi-drop constraints resulting from this discretization led us to develop a decomposition framework.

The algorithmic framework decomposes the original problem (CSP-MD-AW) into smaller problems (relaxations) and obtains valuable information (lower bounds) by solving these smaller problems in order to eventually obtain an optimal solution for the CSP-MD-AW. Note that it is also possible to obtain quick upper bounds by increasing the grid square dimensions and/or partially fixing certain decisions such as rotation of the pallets. This information can then be used to further reduce the solution space for the CSP-MD-AW by eliminating unnecessary grid squares towards the rear of the container. However, due to difficulty of mathematically defining the best securing strategy, a theoretically better solution with the selected objective function may not always lead to loading scheme deemed preferable by human practitioners. This indicates the need for additional physical testing to be conducted in order to determine the best securing practices.

Another functionality of the algorithmic framework presented in this paper is the number of relaxations solved prior to the branch-and-cut procedure. The framework consists of three main steps. It is possible to reduce the number of steps to two or one and employ a single branch-and-cut procedure that activates both axle weight and multi-drop constraints as they are violated. Moreover, considering the fact that the MIP formulation provided better upper bounds for some challenging instances, an additional step to solve the MIP when the branch-and-cut procedure terminates prematurely may be incorporated. In our preliminary experiments, we observed a slight performance improvement with the current framework. However, it is worthwhile investigating the potential of different algorithm variants with comprehensive experiments in the future.

We showed that the CSP-MD-AW is a generalization of the two-dimensional strip packing problem with rotations (2D-SPP-R). Even the special case where no axle weight restrictions are considered remains a generalization of the 2D-SPP-R. We foresee that this special case may be of high relevance to the cutting and packing literature. The framework we provide can be readily utilized for solving this special case, but it is worth investigating other potential methods that can work more efficiently in the absence of the axle weight restrictions.

The managerial insights obtained in this study should direct the future research towards integrated problem variants where the customer sequence also represents a decision that must be made. An intermediate variant could require solving the problem both with the given and the reverse sequence and then select the more convenient solution. A more general but more challenging variant could involve quantifying the securing costs in comparison to the fuel cost of the sequence, before solving an integrated traveling salesperson problem where the objective is to minimize the total

traveling and securing cost. The CSP-MD-AW is defined over a setting where customer pallets are already allocated to individual containers. It is also relevant to consider pallet-container allocation as part of the integrated decisions that must be made. Together with the sequence decisions, this extension would require solving a vehicle routing problem where the objective is to minimize the total fuel and securing cost as well as vehicle purchasing/leasing and personnel hiring cost when relevant, which differs from common practice where the loading sub-problem is treated as a feasibility problem rather than an optimization problem. The methods we provide in this paper can be readily incorporated into an iterative algorithm that solves the CSP-MD-AW for the selected vehicle routes at each iteration. It would also be possible to solve the CSP-MD-AW in parallel for different vehicle routes with identical or non-identical containers.

Acknowledgements

This research was supported by Internal Funds KU Leuven and by the Research Foundation Flanders (FWO) Strategic Basic Research project Data-driven logistics (S007318N). The authors would like to thank Tony Wauters (KU Leuven) for his valuable recommendations. Editorial consultation has been provided by Luke Connolly (KU Leuven).

A Strengthened multi-drop constraints

We construct strengthened versions of Constraints (8) and (9) by disaggregating them for each successor pallet as in Constraints 19 and 20, respectively. Note that n_{p_1} can be replaced by $\min\{n_{p_1}, |COR_{pg}^L|\}$ and $\min\{n_{p_1}, |COR_{pg}^S|\}$ in Constraints 19 and 20, respectively. These constraints can be further strengthened to Constraints 21 and 22.

$$\sum_{h \geq g: h \in COR_{pg}^L} x_{p_1 h} \leq n_{p_1} (1 - y_{pg}^L), \quad \forall p \in P, p_1 \in P_{c(p)}^{fol}, g \in G_p^L \quad (19)$$

$$\sum_{h \geq g: h \in COR_{pg}^S} x_{p_1 h} \leq n_{p_1} (1 - y_{pg}^S), \quad \forall p \in P, p_1 \in P_{c(p)}^{fol}, g \in G_p^S \quad (20)$$

$$\sum_{h \in G: A_{p_1 h}^L \cap COR_{pg}^L \neq \emptyset} y_{p_1 h}^L + \sum_{h \in G: A_{p_1 h}^S \cap COR_{pg}^L \neq \emptyset} y_{p_1 h}^S \leq 1 - y_{pg}^L, \quad \forall p \in P, p_1 \in P_{c(p)}^{fol}, g \in G_p^L \quad (21)$$

$$\sum_{h \in G: A_{p_1 h}^L \cap COR_{pg}^S \neq \emptyset} y_{p_1 h}^L + \sum_{h \in G: A_{p_1 h}^S \cap COR_{pg}^S \neq \emptyset} y_{p_1 h}^S \leq 1 - y_{pg}^S, \quad \forall p \in P, p_1 \in P_{c(p)}^{fol}, g \in G_p^S \quad (22)$$

Another disaggregation of Constraints (8) and (9) can be done for each grid square in the corridor resulting in Constraints 19 and 20, respectively.

$$\sum_{p_1 \in P_{c(p)}^{fol}} x_{p_1 h} \leq 1 - y_{pg}^L, \quad \forall p \in P, g \in G_p^L, h \geq g : h \in COR_{pg}^L \quad (23)$$

$$\sum_{p_1 \in P_{c(p)}^{fol}} x_{p_1 h} \leq 1 - y_{pg}^S, \quad \forall p \in P, g \in G_p^S, h \geq g : h \in COR_{pg}^S \quad (24)$$

In order to further strengthen the introduced multi-drop constraints, the term $(1 - y_{pg}^L)$ on the right-hand side (RHS) of Constraints (8), (19), (21) and (23) can be replaced by $(1 - \sum_{h \leq g: row_g = row_h} y_{ph}^L +$

y_{ph}^S). The term $(1 - y_{pg}^S)$ on the RHS of Constraints (9), (20), (22) and (24) can be replaced by $(1 - \sum_{h \leq g: \text{row}_g = \text{row}_h} y_{ph}^S)$. The intuition behind these replacements is illustrated in Figure 20. For a particular pallet p , COR_{p13}^S is a subset of COR_{p1}^S and COR_{p5}^S . Therefore, no successor pallet would be allowed in COR_{p13}^S if $y_{p13}^S = 1$, $y_{p1}^S = 1$ or $y_{p5}^S = 1$. Given that at most one of these variables can be equal to one, Constraints (26) are valid and stronger than Constraints (9) and (20). Similarly, COR_{p13}^L is a subset of COR_{p1}^L and COR_{p5}^S . Therefore, no successor pallet would be allowed in COR_{p13}^L if $y_{p13}^L = 1$, $y_{p1}^L = 1$ or $y_{p5}^S = 1$. Hence, Constraints (25) are valid and stronger than Constraints (8) and (19).

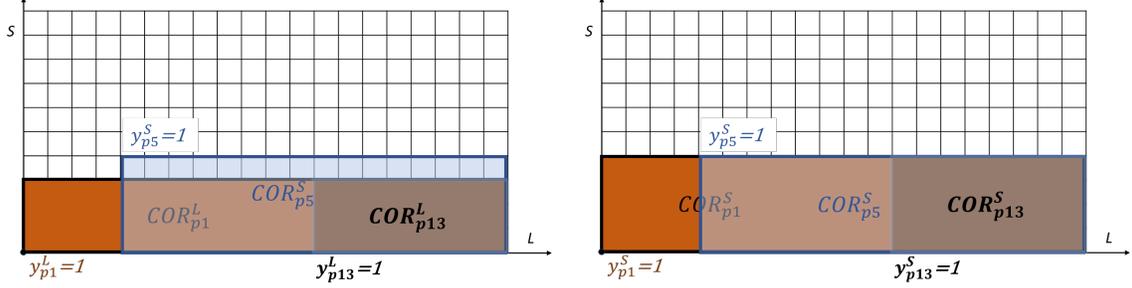


Figure 20: Illustration of the intuition behind Constraints (26) and (25).

$$\sum_{h \geq g: h \in COR_{pg}^L} x_{p1h} \leq n_{p1} \left(1 - \sum_{h \leq g: \text{row}_g = \text{row}_h} (y_{ph}^L + y_{ph}^S)\right), \quad \forall p \in P, p1 \in P_{c(p)}^{fol}, g \in G_p^L \quad (25)$$

$$\sum_{h \geq g: h \in COR_{pg}^S} x_{p1h} \leq n_{p1} \left(1 - \sum_{h \leq g: \text{row}_g = \text{row}_h} y_{ph}^S\right), \quad \forall p \in P, p1 \in P_{c(p)}^{fol}, g \in G_p^S \quad (26)$$

As a comparison, the following aggregations of (21) and (22) are stronger than the multi-drop constraints (7) introduced by De Queiroz and Miyazawa (2013):

$$\sum_{p1 \in P_{c(p)}^{fol}} \left(\sum_{h \in G: A_{p1h}^L \cap COR_{pg}^L \neq \emptyset} y_{p1h}^L + \sum_{h \in G: A_{p1h}^S \cap COR_{pg}^L \neq \emptyset} y_{p1h}^S \right) \leq |P_{c(p)}^{fol}| \left(1 - \sum_{h \leq g: \text{row}_g = \text{row}_h} (y_{ph}^L + y_{ph}^S)\right), \quad \forall p \in P, g \in G_p^L \quad (27)$$

$$\sum_{p1 \in P_{c(p)}^{fol}} \left(\sum_{h \in G: A_{p1h}^L \cap COR_{pg}^S \neq \emptyset} y_{p1h}^L + \sum_{h \in G: A_{p1h}^S \cap COR_{pg}^S \neq \emptyset} y_{p1h}^S \right) \leq |P_{c(p)}^{fol}| \left(1 - \sum_{h \leq g: \text{row}_g = \text{row}_h} y_{ph}^S\right), \quad \forall p \in P, g \in G_p^S \quad (28)$$

B Detailed instance properties

The quantity of $0.8m \times 1.2m$, $1m \times 1.2m$ and $0.8m \times 0.6m$ pallets are denoted by #1, #2 and #3, respectively. Moreover, $|P| = \min_{i \in T} |P_i|$, $|\bar{P}| = \max_{i \in T} |P_i|$, $A_{tot} = \sum_{p \in P} l_p^S \times l_p^L$, $A_{av} = \sum_{p \in P} l_p^S \times l_p^L / |P|$, $W_{tot} = \sum_{p \in P} w_p$, $W_{av} = \sum_{p \in P} w_p / |P|$, $\underline{w} = \min_{p \in P} w_p$, $\bar{w} = \max_{p \in P} w_p$.

Table 8: Detailed instance properties

Category-ID	$ P $	$ \underline{P} $	$ \overline{P} $	#1	#2	#3	A_{tot}	A_{av}	$ T $	W_{tot}	W_{av}	\underline{w}	\overline{w}	ω
NIN-P7C3M	7	2	3	5	2	0	7.20	1.03	3	8935	1276.43	1175	1401	1240.97
NII-P7C3M	7	2	3	5	2	0	7.20	1.03	3	10300	1471.43	500	2400	1430.56
NNN-P7C3M	7	2	3	5	2	0	7.20	1.03	3	8935	1276.43	1175	1401	1240.97
NNN-P10C4L	10	1	5	6	4	0	10.56	1.06	4	6827	682.70	50	1175	646.50
NNN-P10C4M	10	1	5	6	4	0	10.56	1.06	4	12871	1287.10	1102	1401	1218.84
NNN-P12C3M	12	1	7	4	7	1	12.72	1.06	3	14535	1211.25	415	1401	1142.69
NNN-P12C3H	12	2	7	3	0	9	7.20	0.60	3	18700	1558.33	400	5000	2597.22
NNN-P12C3H2	12	1	7	4	7	1	12.72	1.06	3	18820	1568.33	1000	1700	1479.56
III-P14C2H	14	7	7	14	0	0	13.44	0.96	2	20300	1450	500	2400	1510.42
NNN-P14C2L	14	7	7	6	5	3	13.20	0.94	2	5448	389.14	60	1360	412.73
NNN-P14C3L	14	4	5	6	5	3	13.20	0.94	3	5448	389.14	60	1360	412.73
NNN-P14C4L	14	2	5	6	5	3	13.20	0.94	4	5448	389.14	60	1360	412.73
NNN-P14C5L	14	1	4	6	5	3	13.20	0.94	5	5448	389.14	60	1360	412.73
NNN-P14C5M	14	1	5	6	5	3	13.20	0.94	5	14448	1032.00	60	1401	1094.55
NNN-P16C2M	16	8	8	8	7	1	16.56	1.04	2	19447	1215.44	415	1401	1174.34
NNN-P16C3M	16	4	7	8	7	1	16.56	1.04	3	19447	1215.44	415	1401	1174.34
NNN-P16C4M	16	1	7	8	7	1	16.56	1.04	4	19447	1215.44	415	1401	1174.34
NNN-P18C4M	18	1	7	9	8	1	18.72	1.04	4	20628	1213.41	415	1401	1101.92
NNN-P18C5M	18	2	5	8	5	5	16.08	0.89	5	17827	1048.65	60	1401	1108.64
III-P20C2H	20	10	10	20	0	0	19.20	0.96	2	29000	1450.00	500	2400	1510.42
III-P20C4H	20	5	5	20	0	0	19.20	0.96	4	29000	1450.00	500	2400	1510.42
IIN-P20C4H2	20	5	5	20	0	0	19.20	0.96	4	26400	1320.00	400	2000	1375.00
IIN-P20C4H3	20	5	5	20	0	0	19.20	0.96	4	29300	1465.00	400	2000	1526.04
IIN-P22C5H	22	2	5	22	0	0	21.12	0.96	5	31200	1418.18	1200	1500	1477.27
NNN-P23C3M	23	5	9	8	5	10	18.48	0.80	3	20549	893.43	60	1401	1111.96
NNN-P23C5M	23	4	5	8	5	10	18.48	0.80	5	20549	893.43	60	1401	1111.96
NNN-P24C2H	24	11	13	4	0	20	13.44	0.56	2	20762	865.08	400	2400	1544.79
NNN-P24C5M	24	4	6	4	0	20	13.44	0.56	5	14970	623.75	60	1402	1113.84
NNN-P24C5H	24	3	6	4	0	20	13.44	0.56	5	20762	865.08	400	2400	1544.79
NNN-P16C10M	16	1	2	8	7	1	16.56	1.04	10	19447	1215.44	415	1401	1174.34
NNN-P18C10M	18	1	2	9	8	1	18.72	1.04	10	20628	1213.41	415	1401	1101.92
IIN-P20C10H	20	2	2	20	0	0	19.20	0.96	10	26400	1320.00	400	2000	1375.00
IIN-P22C10H	22	2	4	22	0	0	21.12	0.96	10	31200	1418.18	1200	1500	1477.27
NNN-P24C10H	24	2	3	4	0	20	13.44	0.56	10	20762	865.08	400	2400	1544.79
NNN-P30C5H	30	6	6	10	0	20	19.20	0.64	5	26160	872	400	2400	1362.50
NNN-P30C5H2	30	3	10	10	0	20	19.20	0.64	5	26160	872	400	2400	1362.50
NNN-P30C10H	30	3	3	10	0	20	19.20	0.64	10	26160	872	400	2400	1362.50
NNN-P30C10H2	30	2	5	10	0	20	19.20	0.64	10	26160	872	400	2400	1362.50
NNN-P40C5H	40	4	4	2	0	38	20.16	0.50	10	28775	719.375	400	1200	1427.33
NNN-P40C5H2	40	4	4	2	2	36	21.60	0.54	10	28775	719.375	400	1200	1332.18
NNN-P40C10H	40	4	4	2	0	38	20.16	0.50	10	28775	719.375	400	1200	1427.33
NNN-P40C10H2	40	4	4	2	2	36	21.60	0.54	10	28775	719.375	400	1200	1332.18

References

Alonso, M., Alvarez-Valdes, R., Iori, M., and Parreño, F. (2019). Mathematical models for multi container loading problems with practical constraints. *Computers & Industrial Engineering*, 127:722–733.

- Alonso, M., Alvarez-Valdes, R., Iori, M., Parreño, F., and Tamarit, J. (2017). Mathematical models for multicontainer loading problems. *Omega*, 66:106–117.
- Alonso, M., Martinez-Sykora, A., Alvarez-Valdes, R., and Parreño, F. (2022). The pallet-loading vehicle routing problem with stability constraints. *European Journal of Operational Research*, 302(3):860–873.
- Alonso, M. T., Alvarez-Valdés, R., and Parreño, F. (2020). A grasp algorithm for multi container loading problems with practical constraints. *4OR*, 18(1):49–72.
- Baker, B. S., Coffman, Jr, E. G., and Rivest, R. L. (1980). Orthogonal packings in two dimensions. *SIAM Journal on computing*, 9(4):846–855.
- Beasley, J. E. (1985). An exact two-dimensional non-guillotine cutting tree search procedure. *Operations Research*, 33(1):49–64.
- Bortfeldt, A. and Wäscher, G. (2013). Constraints in container loading – A state-of-the-art review. *European Journal of Operational Research*, 229(1):1–20.
- Çalk, H., D’hondt, J., Juwet, M., and Vanden Berghe, G. (2019). An integer programming formulation for the vehicle routing problem with cargo stability and multi-drop constraints. *European Association of Research in Transportation (hEART)*, pages 1–6.
- Commision, E. (2021, accessed 03.09.2021). Cargo securing and abnormal loads. https://ec.europa.eu/transport/road_safety/topics/vehicles/cargo_securing_loads_en.
- Côté, J.-F., Gendreau, M., and Potvin, J.-Y. (2014). An exact algorithm for the two-dimensional orthogonal packing problem with unloading constraints. *Operations Research*, 62(5):1126–1141.
- Da Silveira, J. L., Miyazawa, F. K., and Xavier, E. C. (2013). Heuristics for the strip packing problem with unloading constraints. *Computers & Operations Research*, 40(4):991–1003.
- Da Silveira, J. L., Xavier, E. C., and Miyazawa, F. K. (2014). Two-dimensional strip packing with unloading constraints. *Discrete Applied Mathematics*, 164:512–521.
- de Almeida Cunha, J. G., De Lima, V. L., and De Queiroz, T. A. (2020). Grids for cutting and packing problems: a study in the 2d knapsack problem. *4OR*, 18(3):293–339.
- De Queiroz, T. A. and Miyazawa, F. K. (2013). Two-dimensional strip packing problem with load balancing, load bearing and multi-drop constraints. *International Journal of Production Economics*, 145(2):511–530.
- De Queiroz, T. A. and Miyazawa, F. K. (2014). Order and static stability into the strip packing problem. *Annals of Operations Research*, 223(1):137–154.
- Ferreira, K. M., de Queiroz, T. A., and Toledo, F. M. B. (2021). An exact approach for the green vehicle routing problem with two-dimensional loading constraints and split delivery. *Computers & Operations Research*, 136:105452.
- Gandra, V. M. S., Çalk, H., Wauters, T., Toffolo, T. A., Carvalho, M. A. M., and Vanden Berghe, G. (2021). The impact of loading restrictions on the two-echelon location routing problem. *Computers & Industrial Engineering*, 160:107609.

- Iori, M., de Lima, V. L., Martello, S., Miyazawa, F. K., and Monaci, M. (2021). Exact solution techniques for two-dimensional cutting and packing. *European Journal of Operational Research*, 289(2):399–415.
- Krebs, C. and Ehmke, J. F. (2021). Axle weights in combined vehicle routing and container loading problems. *EURO Journal on Transportation and Logistics*, page 100043.
- Lim, A., Ma, H., Qiu, C., and Zhu, W. (2013). The single container loading problem with axle weight constraints. *International Journal of Production Economics*, 144(1):358–369.
- Liu, M., Man, X., Zheng, F., and Chu, C. (2017). An integer programming model for the single container loading problem with axle weight constraints. In *2017 International Conference on Service Systems and Service Management*, pages 1–5. IEEE.
- Neuenfeldt, A., Silva, E., Francescatto, M., Rosa, C. B., and Siluk, J. (2021). The rectangular two-dimensional strip packing problem real-life practical constraints: A bibliometric overview. *Computers & Operations Research*, page 105521.
- Pollaris, H., Braekers, K., Caris, A., Janssens, G. K., and Limbourg, S. (2015). Vehicle routing problems with loading constraints: state-of-the-art and future directions. *OR Spectrum*, 37(2):297–330.
- Pollaris, H., Braekers, K., Caris, A., Janssens, G. K., and Limbourg, S. (2016). Capacitated vehicle routing problem with sequence-based pallet loading and axle weight constraints. *EURO Journal on Transportation and Logistics*, 5(2):231–255.
- Ramos, A. G., Silva, E., and Oliveira, J. F. (2018). A new load balance methodology for container loading problem in road transportation. *European Journal of Operational Research*, 266(3):1140–1152.
- Silva, E., Ramos, A. G., and Oliveira, J. F. (2018). Load balance recovery for multi-drop distribution problems: A mixed integer linear programming approach. *Transportation Research Part B: Methodological*, 116:62–75.
- Toledo, F. M., Carravilla, M. A., Ribeiro, C., Oliveira, J. F., and Gomes, A. M. (2013). The dotted-board model: A new mip model for nesting irregular shapes. *International Journal of Production Economics*, 145(2):478–487.
- Wei, L., Wang, Y., Cheng, H., and Huang, J. (2019). An open space based heuristic for the 2d strip packing problem with unloading constraints. *Applied Mathematical Modelling*, 70:67–81.