Mathematical Models for Multi Container Loading Problems

M.T.Alonso [‡] ,**R.Alvarez-Valdes** [†] ,**M. Iori** [§],**F.Parreño** [‡] , **J.M.Tamarit** [†]

[‡] University of Castilla-La Mancha, Department of Mathematics, Albacete, Spain.
[†] University of Valencia, Department of Statistics and Operations Research, Burjassot, Valencia, Spain.
[§]University of Modena Reggio Emilia, Department of Sciences and Methods for Engineering, Reggio Emilia, Italy.
mariateresa.alonso@uclm.es; ramon.alvarez@uv.es; manuel.iori@unimore.it; francisco.parreno@uclm.es; jose.tamarit@uv.es

Abstract

This paper deals with the problem of a distribution company that has to serve its customers by putting first the products on pallets and then loading the pallets onto trucks. We approach the problem by developing and solving integer linear models. We start with basic models, that include the essential features of the problem, such as respecting the dimensions of the truck, and not exceeding the total weight capacity and the maximum weigh capacity on each axle. Then, we add progressively new conditions to consider the weight and volume of pallet bases and to include other desirable features for the solutions to be useful in practice, such as the position of the center of gravity and the minimization of the number of pallets.

The models have been tested on a large set of real instances involving up to 46 trucks and kindly provided to us by a distribution company. The results show that in most cases the optimal solution can be obtained in small running times. Moreover, when optimality cannot be proven, the gap is very small, so we obtain high quality solutions for all the instances that we tested.

Key words: containers; integer programming ; optimization ; cutting stock problem

1 Introduction

Everyday a distribution company has to decide how to put goods onto pallets to serve the customers' orders and then how to efficiently distribute these pallets among the trucks, in order to get the right goods to the right place, and in the desired conditions, while minimizing the number of trucks used. In this paper we will take as a reference a large company in Europe, but the problem is common to many other distribution companies around the world. Solving the problem in an optimal way produces a reduction in the transportation costs for the companies, thus increasing the profits and also decreasing the greenhouse emissions.

The loading problem consists of two, interrelated, phases. All items have to be placed on pallets; we call this phase *pallet building*, and all pallets have to be placed onto trucks; we call this phase *truck loading*. In the pallet building phase the items are grouped in layers and the layers are stacked on the pallet base. A *layer* is an arrangement of items of the same product, composing a rectangle whose dimensions and number of items are known. A layer completely covers the pallet base in horizontal directions and other layers can be placed on top of it to make up the pallet. The layer composition of each product has been previously decided, so we have to stack layers to build pallets. Once the pallets are built, they have to be placed onto the trucks. In the following, we suppose that there is an infinite supply of identical trucks.

The loading problem we deal with is a Multi Container Loading Problem (MCLP). The means of transport, in this case the use of trucks, introduces some constraints that have to be respected for safety reasons. One important feature is the weight limit. There is a maximum weight that can be loaded and this limit cannot be exceeded. There are also limits on the maximum weight each axle can bear. Indeed, there is not only a limit on the total weight of the cargo, but there are also limits on the weight that each axle can bear. Excesses over these weight limits represent a risk for traffic safety and can cause damage to the road. Therefore, they are strictly controlled and the violations severely punished. Some roads have Weight-In-Motion systems, that monitor axle weight violations while driving (see, e.g., Jacob and Feypell-de La Beaumelle [15]).Moreover, for safety reasons the load has to be well spread into the truck to avoid movements during the journey. This means that the center of gravity has to be placed in between the axles and as near as possible to the geometric center of the truck.

Unlike the Single Container Loading Problem (SCLP), that has been extensively studied, the MCLP has attracted less attention so far. In particular, the problem studied here of putting goods on pallets and pallets onto trucks is quite a new issue in the cutting and packing literature. As far as we know, just a few previous studies (discussed in the following section) dealt with the MCLP, and not all of them included real constraints such as total weight, axle weight, and position of the center of gravity. Even fewer exact algorithms have been proposed, and just for solving some particular versions of the problem.

For these reasons our proposal is to attempt to solve the problem exactly, in particular by using *Integer Linear Programming* (ILP) models. Apart from being rather easy to implement for a practitioner, ILP models are flexible tools for adding or removing constraints so as to meet the requirements of the specific MCLP at hand, and in recent years have acquired a very good computational behavior, as also witnessed by our results below. We model the characteristics of the problem, starting from a basic model with the main features and then gradually adding new features, one at a time, introducing in the model real constraints required by the company. The models have been tested with real instances provided by a distribution center and the results show that optimal solutions can be obtained in most cases with small computing times. In the cases in which optimality cannot be proven, the solutions usually have a very small gap, and therefore this approach is able to obtain high quality solutions for instances with up to 46 trucks. Our results refer to instances where pallets are loaded onto trucks, but also apply to the loading of containers in general. For that reason in the remaining of the paper we use the terms truck and container as synonyms.

The remainder of the paper is organized as follows. An overview of related existing research is presented in Section 2. In Section 3 the problem is formally described. In Section 4 the test instances are analyzed, and upper and lower bounds for each instance are calculated. In Section 5 we introduce the basic model which includes the main characteristics of the problem, such as not exceeding the truck dimensions and the maximum weight, and then we add axle weight constraints. In Section 6 we include the pallet bases, considering one and two pallets per position. We study additional conditions such as the center of gravity and the minimization of the number of pallets in Section 7. Section 8 contains the conclusions.

2 Previous work

There are not many papers that address the issues studied here, considering pallet and truck loading together. Following the typology for cutting and packing problems introduced by Wäscher et al. [28] the two problems, pallet and truck loading, can be classified as Single Stock Size Cutting Stock Problems. Morabito et al. [21] deal with the same problem but in two dimensions, because the products cannot be stacked. In a first phase, the problem consists in loading the maximum number of products on a pallet. They solve the problem by using the 5-block algorithm proposed by Morabito and Morales [20]. When the pallets are built, they use the same approach to load the pallets onto the trucks. Takahara [26] deals with the problem of loading a set of items on a set containers and pallets. A loading sequence for the items is chosen and it determines the order in which the items are inserted into the bins. The sequence is selected by a metaheuristic method based on a neighborhood search. A selector determines the sequence of the bins. When a bin is selected, the first item is loaded into the bin, placing it in the first space in which it fits. If the item does not fit into the bin, the next bin is selected. A strategic procedure determines when to exchange the sequence of the items with a neighbor sequence and when the choice of the bin is changed from following the sequence to being randomly chosen, depending on the quality of the solutions.

The SCLP is instead a well-studied problem (see, e.g., Bischoff and Ratcliff [3], Lim et al. [19], Jin et al. [16], Araujo and Armentano [1], Fanslau and Bortfeldt [10], and Junqueira et al. [17]), where the real constraints that we are also facing received an increased attention. According to the survey by Bortfeldt and Wäscher [7], at least 13.9% of the container loading literature deals with weight limit, while weight distribution is considered by 12.1% of the papers. Gehring and Bortfeldt [11], Bortfeldt et al. [6], Terno et al. [27], and Egeblad et al. [9] are some of the authors who include weight limit constraints in their studies. Indeed, when the cargo is heavy, the weight becomes a very restrictive constraint, more than the volume or the space occupied.

Weight distribution constraints require the weight of the cargo to be spread across the container floor, to avoid displacements during the journey or to balance the load between truck axles when the container is transported by truck. To achieve a good weight distribution, the center of gravity of the load should be in the geometrical mid-point of the container floor, as in Bischoff and Marriott [2], or should not exceed a certain distance from it, as in Bortfeldt and Gehring [5] and Gehring and Bortfeldt [11].

Axle weight is a constraint imposed by the means of transport and it has not been widely studied. Lim et al. [18] deal with a particular SCLP with axle weight constraints. They propose an integrated heuristic solution approach that combines a GRASP wall-building algorithm with ILP models. They first apply a customized wall-building heuristic based on the GRASP by Moura and Oliveira [22], including special considerations for box weight and density. Then they use an integrated approach to handle the weight requirements. If the container load limit is exceeded, they unload the necessary number of boxes by iteratively solving an ILP model to meet the requirement. If the axle weight limit is exceeded, they take two steps iteratively until the limit is satisfied: the first step consists in interchanging the positions of the walls created by the customized heuristic, whereas the second step consists in solving an ILP model to unload boxes and in applying one more time the first step to improve the container balance as well as to force a feasible weight distribution.

Haessler and Talbot [12] describe a heuristic for loading customers' orders, and developing load patterns for trucks and rail shipments. The products have low density and for that reason the approach is based on loading by volume rather than by weight. To deal with axle weight constraints, stacks are sequenced by alternating the heaviest and lightest stacks.

Weight constraints also appear in recent studies combining loading and routing, such as, e.g., Iori et al. [14] and Bortfeldt [4]. Doerner et al. [8] deal with a particular vehicle routing problem in which the items are placed on pallets and stacked one above the other, producing piles. They propose two metaheuristic algorithms, a Tabu Search, and an ACO algorithm. A survey on loading and routing problems is presented by Iori and Martello [13]. A recent work not included in the survey is the one by Pollaris et al. [25], that combine a capacitated vehicle routing problem with the loading of homogeneous pallets inside the vehicle. They consider sequence-based pallet loading and axle weight constraints, and propose a mixed ILP formulation for the problem. Pallets may be placed in two rows inside the vehicle but cannot be stacked on top of each other because of their weight, fragility, or customer preferences. Sequence-based loading is assumed to ensure that, when arriving at a customer, no items belonging to customers served later on the route block the removal of the items of the current customer. Their model tries to minimize the transportation cost, respecting the axle weight constraint along the delivery route.

3 Problem description

The problem we deal with is to supply a costumer, whose demand can be defined as a list of products $j \in J$, which have to be served completely. Each product has a layer composition with dimensions (l_j, w_j, h_j), weight q_j, and a quantity of demanded layers n_j. Unless stated otherwise, layers are placed on flat bases, called pallet bases in the following, with dimensions (l^p, w^p, h^p) and weight q^p, forming pallets. A pallet is consequently composed of a flat base and a set of layers placed one above the other.

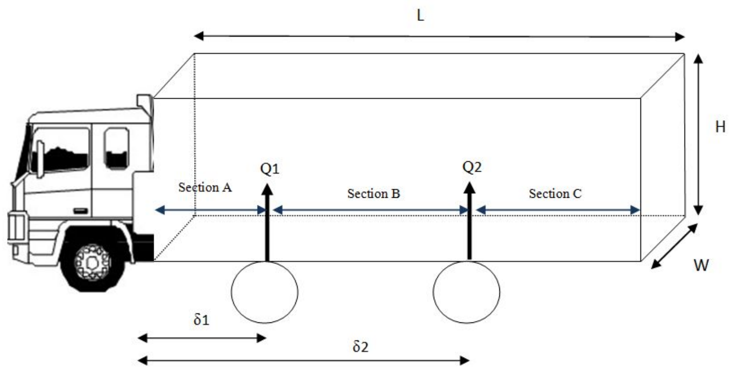


Figure 1: Dimensions and axles positions of the truck

The pallets are loaded onto trucks, and a set of trucks $k = \{1, \ldots, |K|\}$, large enough to accommodate all the products, is available. All trucks are identical and we know their dimensions (L, W, H), the maximum weight they can bear on the front axle, Q_1, and on the rear axle, Q_2, and the maximum total weight Q. The distances to the axles from the front part of the truck are δ_1 and δ_2. Figure 1 shows a truck with its dimensions and axle distances. The aim of the problem is to minimize the number of trucks used to load all products.

A consequence of the facts that we are not considering layers bigger than the pallet base, and that thus all pallets have the same surface dimensions, is that we can place pallets in a fixed number of positions, $i \in I$, onto the truck. In particular, the dimensions of the truck are (L, W, H), the dimensions of the pallet (l^p, w^p, h^p), and we consider that we can place $\lfloor \frac{L}{w^p} \rfloor$ pallets along the truck's length and $\lfloor \frac{W}{l^p} \rfloor$ across the truck's width. Therefore, the number of pallets that can be placed on the truck's surface is $|I| = \lfloor \frac{L}{w^p} \rfloor * \lfloor \frac{W}{l^p} \rfloor$. The resulting grid can be seen in Figure 2, that gives a top view of a truck with some of the positions in which a pallet can be placed. For the usual case of ISO pallets and containers, the resulting grid is formed by 2×16 positions. In the following we uniquely identify the positions on a truck by their middle points (p_i^x, p_i^y, p_i^z), $i \in I$, as can be observed again in Figure 2.

In the most common MCLP configuration, when the pallets are placed onto the truck, the load cannot exceed the maximum weight Q, and the distribution of the load cannot exceed the maximum

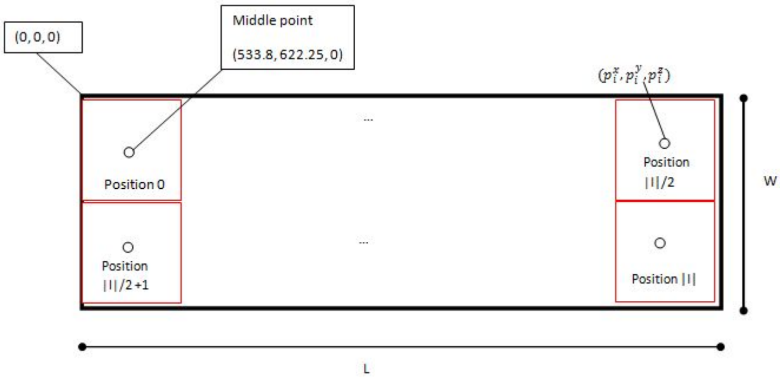


Figure 2: Pallet positions on the truck floor

axle weights Q_1 and Q_2. The way in which the pallet weight is supported by the axles follows the law of levers. The load supported varies in relation with the position of the load on the truck. In Figure 1, we have divided the truck into three sections. A pallet can be placed in Section A, between 0 and the first axle, or between the two axles, Section B, or beyond the second axle, Section C. Suppose a pallet i of weight q_i is placed in a position i on the truck, defined by its middle point p_i^x on the length dimension. Then the force applied on each axle depends on the weight and the position, in the way shown in Table 1.

	Position	Force on axle 1	Force on axle 2
Section A	$0 \le p_i^x \le \delta_1$	$q_i(\delta_2 - p_i^x)$	$-q_i(\delta_1 - p_i^x)$
Section B	$\delta_1 < p_i^x \le \delta_2$	$q_i(\delta_2 - p_i^x)$	$q_i(p_i^x - \delta_1)$
Section C	$\delta_2 < p_i^x \le L$	$-q_i(p_i^x - \delta_2)$	$q_i(p_i^x - \delta_1)$

Table 1: Axle forces per section

If the pallet is placed in front of axle 1, the force on axle 1 is positive and the force on axle 2 is negative. If the pallet is placed between the two axles, both forces are positive. If the pallet is placed behind axle 2 the force on axle 1 is negative and the force on axle 2 is positive. The forces have to be in equilibrium, not exceeding the maximum force allowed on each axle. For that reason the sum of all the forces exerted by the pallets from their positions in the truck should be lower than or equal to the maximum force each axle can withstand, thus satisfying Equations (1) and (2).

$$\sum_{i \in I} q_i(\delta_2 - p_i^x) \le Q_1(\delta_2 - \delta_1) \tag{1}$$

$$\sum_{i \in I} q_i(p_i^x - \delta_1) \le Q_2(\delta_2 - \delta_1) \tag{2}$$

The MCLP that we have just described may be characterized by additional practical loading requirements. For the sake of clarity, we postpone the description of these requirements to the respective sections below in which we also model them as ILP constraints.

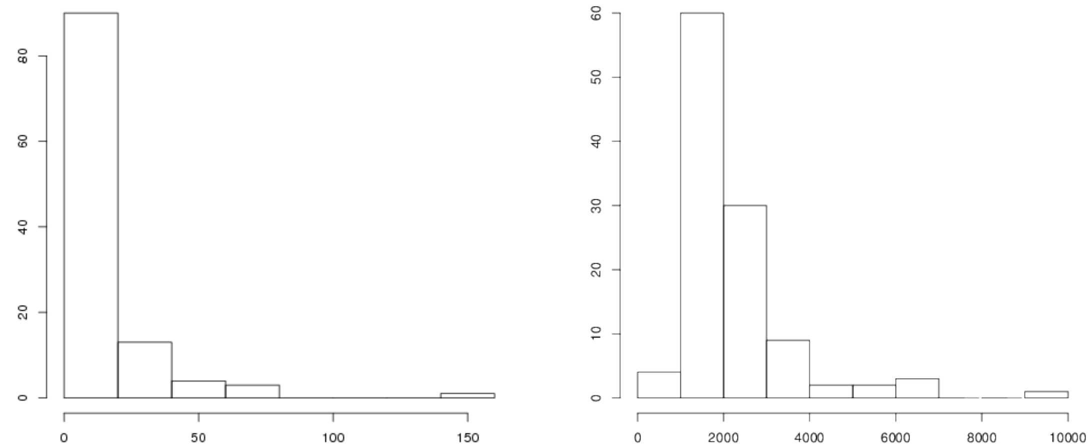
4 A real world benchmark dataset

Our benchmark is made by 111 real instances derived from the everyday distribution activity of a large company. The instances have been provided to us by ORTEC [23], a company developing planning and optimization solutions and services for manufacturing and logistics companies.

A first data analysis of the instances can be seen in Table 2. The distribution of the products ranges between 1 and 142 different types. The demand distribution varies from 241 to 9537 layers. The histograms of the data distribution by number of products and number of layers are shown in Figure 3.

	Products	Layers
Min	1	241
1st Quartile	3	1368
Median	8	1782
3rd Quartile	16.5	2510
Max	142	9537

Table 2: Statistical analysis of the number of products ($|J|$) and number of layers ($\sum_j n_j$)



(a) by products

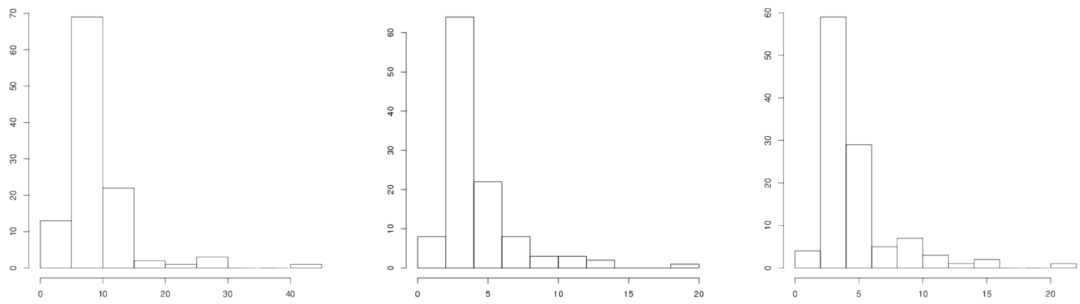
(b) by layers

Figure 3: Distribution of products and layers

Using these data we can calculate some first lower bounds on the number of trucks required for each instance, according to weight, volume, and number of positions. The bound based on the weight is the sum of the weights of all layers divided by the weight capacity of the truck. The bound based on the volume is calculated in the same way. The bound based on the number of positions is calculated dividing the sum of the heights of the layers by the the truck height and by the number of positions in a truck. Table 3 shows the distribution of the minimum number of trucks by weight in the first column, by volume in the second, and by positions in the third. The histograms of the bound distributions are then shown in Figure 4. It can be observed that the bound by positions dominates the bound by volume.

	By weight	By volume	By positions
Min	3	2	2
1st Quartile	6	3	3
Median	8	4	4
3rd Quartile	11	5	6
Max	41	19	21

Table 3: Statistical analysis on the lower bounds on the number of required trucks



(a) by weight (b) by volume (c) by positions

Figure 4: Distribution of the lower bounds on the number of required trucks

The maximum of these three bounds, L_{init}, given by Equation (3), is a valid lower bound on the number of trucks required for each instance for all the problem configurations that we address in the following sections.

$$L_{init} = max\left\{\left\lceil\frac{\sum_{j\in J} q_j * n_j}{Q}\right\rceil, \left\lceil\frac{\sum_{j\in J} l_j * w_j * h_j * n_j}{L * W * H}\right\rceil, \left\lceil\frac{\frac{\sum_{j\in J} h_j * n_j}{H}}{I}\right\rceil\right\} \quad (3)$$

We also need an upper bound on the number of trucks, in order to estimate the difficulty of the instances and limit the number of variables in our ILP models. This upper bound, called U_{init}, is calculated by using a constructive algorithm. This algorithm builds a solution by means of an iterative process made by two steps. The first step is to select a position in the truck to place the next pallet. Since the weight has to be balanced between the two axles, we divide the truck into two parts, front and back, and start the packing process from the center. By placing a pallet each time on a different side, we can control the balance of the weight. Every time a pallet is placed, the weight supported by each axle is calculated. If the weight on axle 1 is greater than that on axle 2, the next space is selected at the back, otherwise it is chosen at the front.

In the second step, a pallet is built taking into account the position in which it will be placed. The layers of products are ordered by density, with the most dense first. We take the first layer of the list and check if it is a candidate to compose a pallet. A layer is selected if the weight of the pallet, including this layer, does not exceed the maximum allowed weight for this space. The selected layer is added to the pallet, and the process continues until no more layers can be added and the pallet can be thus considered completed.

The process continues until all the layers have been placed or no more pallets can be placed on the truck because one of the basic constraints, limiting the volume, the weight, or the number of positions, has been reached. If there are still products to be shipped, a new truck is opened and the process goes on. The resulting solution is valid for the basic packing configuration of the MCLPs that we study, but the heuristic algorithm has also been adapted to take into consideration some more complex constraints, as described below in the following sections.

The instances have been classified into four classes according to the difference between these initial lower and upper bounds (see Table 11 in Annex): class A, when the difference between lower and upper bound is 0; class B when the difference is 1; class C when the difference is 2, and class D when the difference is 3 or more. The table shows the number of the instance, the number of products, the quantity of layers demanded, and the lower and upper bound values. In the next sections we also study how the difficulty of the instances might change and how the bounds might be affected by the addition of new loading constraints.

5 Basic MCLP models

The goal of this section is to introduce some first ILP models for the MCLPs. These models do not solve the complete optimization problem outlined in the previous sections, but focus on the basic packing issues. They are used to assess the difficulty of the real-world instances that we presented, and to provide valid lower bounds for the more complex MCLP configurations that we address in the next sections. In Section 5.1 we provide the basic MCLP model, that only deals with the packing of the layers intro the minimum number of trucks, whereas in Section 5.2 we include the constraint that limits the maximum weight on the truck axles (and that is common to almost all real-world MCLPs). Note that in this section we do not take into account pallet bases, that will be considered only starting from the next section.

5.1 Pure container packing

Let us consider the version of the MCLP in which we only deal with the packing of the layers into the minimum number of containers. By introducing the two decision variables

$$x_{kij} = \text{number of layers of product } j \text{ packed in position } i \text{ of container } k \qquad j \in J, i \in I, k \in K \qquad (4)$$

$$y_k = 1 \text{ if container } k \text{ is used, 0 otherwise} \qquad k \in K \qquad (5)$$

the problem can be modeled as the following ILP:

$$\text{(pure packing)} \quad \min \sum_{k \in K} y_k \qquad (6)$$

$$\sum_{k \in K} \sum_{i \in I} x_{kij} \geq n_j \qquad j \in J \qquad (7)$$

$$\sum_{j \in J} h_j x_{kij} \leq H y_k \qquad k \in K, i \in I \qquad (8)$$

$$\sum_{i \in I} \sum_{j \in J} q_j x_{kij} \leq Q y_k \qquad k \in K \qquad (9)$$

$$y_k \geq y_{k+1} \qquad k \in K : k < |K| \tag{10}$$

$$x_{kij} \geq 0, \text{integer} \qquad k \in K, i \in I, j \in J \tag{11}$$

$$y_k \in \{0,1\} \qquad k \in K \tag{12}$$

Constraints (7) impose that n_j layers are shipped for any product j. Constraints (8) require that the height of the layers packed in a position does not exceed the container height. Constraints (9) set the weight limit to Q for each container.

Constraints (10) are not required for the completeness of the model, but reduce the size of the enumeration tree by forcing containers to be used by increasing order of index. To keep the number of variables as low as possible, the size of the fleet of trucks is set to $|K| = U_{init}$, i.e., to the upper bound value computed by the constructive heuristic of Section 4.

Apart from the simple improvements introduced to reduce the size of the enumeration tree and limit the number of variables, another improvement may be obtained by tightening the right-hand side of constraints (8) and (9). For what concerns (8), we attempt to reduce the maximum height allowed for a packing, by considering all combinations of layers heights not exceeding H. If the maximum attained value, say, H', is lower than H, then we can set $H = H'$ being sure that no feasible solution is missed. The value of H' may be obtained by solving a *subset sum problem* (SSP): given the set J of layers of height h_j, find the subset of J whose total height is a maximum but does exceed H. The SPP can be formally stated by introducing a binary variable ξ_j taking value 1 if and only if j is in the selected subset, thus obtaining:

$$\text{(SSP)} \quad \max \qquad H' = \sum_{j \in J} h_j \xi_j \tag{13}$$

$$\sum_{j \in J} h_j \xi_j \leq H \tag{14}$$

$$0 \leq \xi_j \leq n_j, \text{integer} \qquad j \in J \tag{15}$$

The minimum capacity value, say, Q', that can be used to tighten the right-hand side of (9) may be obtained by solving an equivalent SSP, in which q_j replaces h_j in (13) and (14) and Q replaces H in (14). The values of H' and Q' are then used to replace, respectively, H in (8) and Q in (9).

The ILP model obtained in this way has been solved with CPLEX 12.51, maintaining the default parameters but imposing the use of a single thread. The test were performed on a PC Inter Core i3-2100 (3.1Ghz, 4GB), allowing a maximum time limit of 5 CPU minutes per instance. In our implementation, instead of solving the SSP with the model (13)–(15), we found convenient to use the dedicated algorithm by Pisinger [24]. Before executing the ILP model we compute L_{init} and U_{init}, using the algorithms described in Section 4.

The computational results that we obtained are given in Table 4. The top part of the table gives aggregate information on the model behavior, as each line presents average or total values obtained for the corresponding class of instances. The bottom part of the table gives details on the unsolved instances. Apart from the name of the class and the number of instances in the class (*# inst.*), the columns in Table 4-(a) have the following meanings: L and U give the average lower and upper bound returned by the model; *missed* gives the total number of instances not solved to proven optimality; *gap* gives the total absolute gap, that is, the total difference between U and L on all the instances in the line; *nodes* gives the average number of nodes explored by the enumeration tree of the model , and *sec* the average computational effort in seconds; $nodes_{opt}$ and sec_{opt} provide, respectively, the same information given by *nodes* and *sec* but refer only to the instances solved to proven optimality by the model.

Table 4: Computational results for the pure packing model

(a) Aggregate results (total/average values)

class	# inst.	missed	gap	L	U	nodes	sec	nodes$_{opt}$	sec$_{opt}$
A	27	0	0	8.19	8.19	0	0.04	0	0.04
B	53	0	0	8.98	8.98	0	0.07	0	0.07
C	20	0	0	9.20	9.20	49	0.32	49	0.32
D	11	2	2	17.82	18.00	15425	54.65	116	0.52
tot/avg	111	2	2	9.70	9.72	1537	5.51	19	0.14

(b) Details for the unsolved instances

class	number	$\|J\|$	$\sum_j n_j$	L_{init}	U_{init}	L	U	nodes
D	17	25	6248	26	29	26	27	80322
D	106	25	4088	15	21	15	16	88303

Classes A, B, and C are very easy and our approach solves all of them in less than one second on average. Class D is more challenging. Nine instances are solved to proven optimality using 116 nodes and 0.5 seconds on average, but two instances remain unsolved in the given time limit. The use of the SPP-based improvement is quite positive, especially in reducing the number of explored nodes. Indeed nodes$_{opt}$ is about 49 on average when SPP is used, and about 209 on average when it is not. Details on the two unsolved instances are provided in Table 4-(b). The two instances are very different one from the other. In instance 17, the weight of the layers is a tight constraint, whereas their volume is not as important (indeed, for this instance we have $L_1 = 26$, $L_2 = 8$, and $L_3 = 9$). In instance 106 the volume is instead tighter than the weight ($L_1 = 11$, $L_2 = 14$, and $L_3 = 15$). The difference between the upper and lower bound is just one bin in both cases, so the gap is very small. Note that the model is very effective in reducing the initial upper bound. The number of nodes explored is quite high, about 80.000. Note that we could not find proven optimal solutions on these two instances even by running the ILP model for 3 CPU hours. This confirms the asymptotic behavior that is common to ILP applications for NP-hard problems.

5.2 Axle weight constraints

The majority of MCLPs encountered in practice contemplate a maximum weight limit on each axle of the truck. Following Equations (1) and (2), this restriction my be added to the previous pure packing ILP model as:

$$\sum_{i \in I} \sum_{j \in J} q_j x_{kij} (\delta_2 - p_i^x) \le Q_1 (\delta_2 - \delta_1) y_k \qquad k \in K \tag{16}$$

$$\sum_{i \in I} \sum_{j \in J} q_j x_{kij} (p_i^x - \delta_1) \le Q_2 (\delta_2 - \delta_1) y_k \qquad k \in K \tag{17}$$

Constraint (16) impose that the sum of the angular momenta of the layers on the front axle does not exceed the available capacity Q_1 multiplied by the distance between the axes $(\delta_2 - \delta_1)$ (refer to Figure 1). Constraint (17) impose the same restriction on the back axle of the truck.

The resulting ILP model is thus to minimize (6), subject to (7)–(17). Following what done for the pure packing model in the Section 5.1, also in this case we replace H with H' and Q with Q'. The computational results that we obtained with this model are presented in Table 5.

Table 5: Computational results with the inclusion of axle weight constraints

(a) Aggregate results (total/average values)

class	# inst.	missed	gap	L	U	nodes	sec	nodes$_{opt}$	sec$_{opt}$
A	27	0	0	8.19	8.19	0	0.07	0	0.07
B	53	0	0	9.00	9.00	19	0.19	19	0.19
C	20	1	1	9.20	9.25	13981	15.31	26	0.44
D	11	2	2	17.82	18.00	7931	55.07	128	0.93
tot/avg	111	3	3	9.71	9.74	3314	8.32	24	0.26

(b) Details for the unsolved instances

class	number	$\|J\|$	$\sum_j n_j$	L_{init}	U_{init}	L	U	nodes
C	34	22	1945	5	7	5	6	279117
D	17	25	6248	26	29	26	27	46477
D	106	25	4088	15	21	15	16	39607

The introduction of the axle weight constraints makes the MCLP slightly more difficult, as can be seen by comparing the results in Table 4 with those of Table 5. The average lower and upper bound value increase, respectively, from 9.70 to 9.71 and from 9.72 to 9.74. Three instances are now unsolved to proven optimality: instances 17 and 106 (that were unsolved even by the pure packing model), and instance 34. In particular, instance 34 had $L = U = 5$ in the pure packing case, but $L = 5$ and $U = 6$ with axle weight constraints. Notably the gap is just one bin for the three instances, and the model is effective in reducing the upper bound provided by the initialization phase.

6 MCLPs with pallets

The most common way of moving items is to first load them on pallet bases, whose dimensions have thus to be considered when providing packing solutions. In this section we show how to include pallet bases into the ILP models, by focusing on two cases. In the first case, presented in Section 6.1, we suppose that all layers occupying a position can be packed on a single pallet base, forming a stack as high as the container. In the second case, presented in Section 6.2, we deal instead with an additional maximum height constraint, quite common in practice, that forces stacks that are too high to be packed on two pallet bases, one above the other. As mentioned in Section 3, in our instances we deal with *iso-pallets*, whose base occupies exactly one position in the container. In the following let h^p be the pallet base height and q^p its weight.

6.1 Single pallets

To model the case where a single pallet base is used for a stack of layers in a position, we consider the ILP model with axle weight constraints of Section 5.2 and introduce the additional variable

$$z_{ki} = 1 \text{ if a pallet is packed in position of } i \text{ of container } k, 0 \text{ otherwise} \qquad i \in I, k \in K \qquad (18)$$

The MCLP with axle weight constraints and single pallets can be modeled as the following ILP.

$$\text{(single pallets)} \quad \min \sum_{k \in K} y_k \qquad (19)$$

$$\sum_{k \in K} \sum_{i \in I} x_{kij} \geq n_j \qquad j \in J \qquad (20)$$

$$\sum_{j \in J} h_j x_{kij} + h^p z_{ki} \leq H'' y_k \qquad k \in K, i \in I \qquad (21)$$

$$\sum_{i \in I} (q^p z_{ki} + \sum_{j \in J} q_j x_{kij}) \leq Q'' y_k \qquad k \in K \qquad (22)$$

$$\sum_{i \in I} (q^p z_{ki} + \sum_{j \in J} q_j x_{kij})(\delta_2 - p_i^x) \leq Q_1(\delta_2 - \delta_1) y_k \qquad k \in K \qquad (23)$$

$$\sum_{i \in I} (q^p z_{ki} + \sum_{j \in J} q_j x_{kij})(p_i^x - \delta_1) \leq Q_2(\delta_2 - \delta_1) y_k \qquad k \in K \qquad (24)$$

$$z_{ki} \leq \sum_{j \in J} x_{kij} \qquad k \in K, i \in I \qquad (25)$$

$$y_k \geq y_{k+1} \qquad k \in K : k < |K| \qquad (26)$$

$$x_{kij} \geq 0, \text{integer} \qquad k \in K, i \in I, j \in J \qquad (27)$$

$$y_k \in \{0, 1\} \qquad k \in K \qquad (28)$$

$$z_{ki} \in \{0, 1\} \qquad k \in K, i \in I \qquad (29)$$

Constraints (21) generalize (8) by including the height of the pallet base in the maximum height allowed to each stack. The value of H'' has been computed by solving the following SSP: $H'' = h^p + \max\{\sum_{j \in J} h_j \xi_j$, subject to $\sum_{j \in J} h_j \xi_j \leq (H - h^p)$, $0 \leq \xi_j \leq n_j$ and integer for $j \in J\}$. Similarly, constraints (22) impose the maximum capacity on each container, and use $Q'' = q^p + \max\{\sum_{j \in J} q_j \xi_j$, subject to $\sum_{j \in J} q_j \xi_j \leq (Q - q^p)$, $0 \leq \xi_j \leq n_j$ and integer for $j \in J\}$. The original maximum weight constraints on the two axles, (16) and (17), have been extended respectively to (23) and (24) so as to include the weight of the pallet bases. Constraints (25) impose that a pallet is used in a given position of a container if at least a layer is packed there.

The results obtained by the ILP model (19)–(29) are given in Table 6. As can be noted by comparison with the results in Table 5, the impact of the pallets on the MCLP is quite relevant. Considering the instances solved to proven optimality, the average CPU effort increases from about 0.2 seconds to slightly more than a second, and the average number of explored nodes is about 20 times larger. Five instances are unsolved to proven optimality, one in class B and four in class D. The smallest unsolved instance has just 4 products, but, despite the exploration of more than 300,000 nodes, the difference between lower and upper bound is still of one bin. The gap is one bin also for the other four instances. For instance 17, the inclusion of the new constraints and variables has quite a large impact due to the high number of containers used, and indeed the average number of nodes has decreased to a third of those explored in the case without pallets.

Table 6: Computational results with the inclusion of single pallets in the packing

(a) Aggregate results (total/average values)

class	# inst.	missed	gap	L	U	nodes	sec	nodes$_{opt}$	sec$_{opt}$
A	27	0	0	8.19	8.19	0	0.10	0	0.10
B	53	1	1	9.00	9.02	5713	5.87	38	0.28
C	20	0	0	9.25	9.25	2403	3.42	2403	3.42
D	11	4	4	17.91	18.27	36624	111.31	231	3.99
tot/avg	111	5	5	9.73	9.77	6790	14.47	487	1.07

(b) Details for the unsolved instances

class	number	$\|J\|$	$\sum_j n_j$	L_{init}	U_{init}	L	U	nodes
B	60	4	3494	8	9	8	9	300837
D	93	15	2397	7	10	7	8	241866
D	17	25	6248	26	29	26	27	15194
D	95	15	4090	12	16	12	13	105861
D	106	25	4088	15	21	16	17	38327

6.2 Double pallets

An additional constraint imposing a maximum value on the height of the stacks is frequently encountered in practice. This derives from the fact that it can be difficult, or even impossible, to load/unload too high stacks in the containers or store them in the warehouses. A common way to deal with this constraint, while keeping at the same time dense loadings inside the containers, is to allow two pallets to be packed one over the other on the same position. In the following let H_{max} be the maximum height imposed on a stack ($0 \leq H_{max} \leq H$).

To model the packing of up to two stacks in the same position, we double the set of available positions in a container. To this aim let $I^2 = 1, 2, \ldots, 2|I|$ denote the new set of positions, and $i + |I|$ be the position right above i, for $i \in I$. We obtain the following ILP model.

$$\text{(double pallets)} \quad \min \sum_{k \in K} y_k \tag{30}$$

$$\sum_{k \in K} \sum_{i \in I^2} x_{kij} \geq n_j \qquad j \in J \tag{31}$$

$$\sum_{i \in I} (h^p z_{ki} + h^p z_{ki+|I|} + \sum_{j \in J} (h_j x_{kij} + h_j x_{ki+|I|j})) \leq \widetilde{H} y_k \qquad k \in K \tag{32}$$

$$\sum_{i \in I^2} (q^p z_{ki} + \sum_{j \in J} q_j x_{kij}) \leq Q'' y_k \qquad k \in K \tag{33}$$

$$\sum_{i \in I^2} (q^p z_{ki} + \sum_{j \in J} q_j x_{kij})(\delta_2 - p_i^x) \leq Q_1(\delta_2 - \delta_1) y_k \qquad k \in K \tag{34}$$

$$\sum_{i \in I^2} (q^p z_{ki} + \sum_{j \in J} q_j x_{kij})(p_i^x - \delta_1) \leq Q_2(\delta_2 - \delta_1) y_k \qquad k \in K \tag{35}$$

$$\sum_{i \in I^2} z_{ki} \le 2|I| y_k \qquad k \in K \tag{36}$$

$$\sum_{j \in J} h_j x_{kij} \le H^{max} z_{ki} \qquad k \in K, i \in I^2 \tag{37}$$

$$z_{ki} \le \sum_{j \in J} x_{kij} \qquad k \in K, i \in I^2 \tag{38}$$

$$z_{k,i+|I|} \le z_{ki} \qquad k \in K, i \in I \tag{39}$$

$$y_k \ge y_{k+1} \qquad k \in K : k < |K| \tag{40}$$

$$x_{kij} \ge 0, \text{integer} \qquad k \in K, i \in I^2, j \in J \tag{41}$$

$$y_k \in \{0, 1\} \qquad k \in K \tag{42}$$

$$z_{ki} \in \{0, 1\} \qquad k \in K, i \in I^2 \tag{43}$$

Constraints (31), (34), (35), and (38) generalize those of the model for the packing of single pallets by including the new set of positions. Constraints (32) and (33) impose the maximum height and weight, respectively, with $\widetilde{H} = 2h^p + \max\{\sum_{j \in J} h_j \xi_j$, subject to $\sum_{j \in J} h_j \xi_j \le (H - 2h^p)$, $0 \le \xi_j \le n_j$ and integer for $j \in J\}$, and Q'' computed as in Section 6.1. The maximum height of the stack is given by (37). Constraints (39) allow a stack to be packed in a top position only if a stack has been packed in the corresponding bottom position.

We tested model (30)–(43) with $H_{max} = H/2$. The results are given in Table 7. A proven optimal solution is not achieved for just four instances out of 111. The average lower bound value increases from 9.73 for the single pallet packing to 9.81, and similarly the upper bound value varies from 9.77 to 9.88. The gaps are small on average but quite large for the instances 17 and 106. In this case the double number of positions, and the consequent increase in the number of constraints and variables, reduce the number of explored nodes by 3/5 times with respect to the case with single pallets, and thus the model is not so effective in providing improved lower and upper bound values. Note that for instance 95 we could find a proven optimal solution using 13 bins with a slightly changed configuration of the model.

7 Additional problem features

As discussed in the introduction, the MCLPs may be characterized by a great variety of technical features. In this section we describe two features that are of practical interest and may enrich the ILP models presented in the previous sections.

7.1 Center of gravity

A good distribution of the load weight inside the container is important to avoid load movements or unbalanced situations. This is crucial when, for example, the container is lifted to be loaded on trains or ships, or even more when it is moved on road by a truck that may brake or bend sharply. Let us define by (G_x, G_y) the desired position of the *center of gravity*. When the container is loaded on a truck, the load can be the geometrical center of the truck, or, in general, any position not exceeding the truck rear axle. We consider a load as feasible if its center of gravity lies in the x-interval $[G_x - \tau_1^x, G_x + \tau_2^x]$ and in the y-interval $[G_y - \tau_1^y, G_y + \tau_2^y]$, where the τ are input tolerance parameters whose values are imposed by the MCLP at hand.

The computation of the overall center of gravity has to take into account the contribution of the weight of the empty container, defined Q^{empty} in the following, and of the weights of the loaded

Table 7: Computational results with the inclusion of up to two pallets per position

(a) Aggregate results (total/average values)

class	# inst.	missed	gap	L	U	nodes	sec	nodes$_{opt}$	sec$_{opt}$
A	27	0	0	8.19	8.19	0	0.43	0	0.43
B	53	0	0	9.04	9.04	31	1.72	31	1.72
C	20	1	1	9.35	9.40	754	16.77	1	1.96
D	11	3	7	18.36	19.00	3813	85.02	63	5.70
avg/sum	111	4	8	9.81	9.88	529	12.37	20	1.73

(b) Details for the unsolved instances

class	number	$\|J\|$	$\sum_j n_j$	L_{init}	U_{init}	L	U	nodes
C	37	19	2064	6	8	6	7	15060
D	17	25	6248	26	29	26	29	4900
D	95	15	4090	12	16	13	14	28855
D	106	25	4088	15	21	17	20	7691

items. The sum of these contributions should not exceed the minimum and maximum distances from the desired position on both axis. This can be modeled in terms of ILP by using the following constraints.

$$Q^{empty} G_x + \sum_{i \in I} \sum_{j \in J} p_i^x q_j x_{kij} \leq (\sum_{i \in I} \sum_{j \in J} q_j x_{kij} + Q^{empty})(G_x + \tau_1^x) \qquad k \in K \tag{44}$$

$$Q^{empty} G_x + \sum_{i \in I} \sum_{j \in J} p_i^x q_j x_{kij} \geq (\sum_{i \in I} \sum_{j \in J} q_j x_{kij} + Q^{empty})(G_x - \tau_2^x) \qquad k \in K \tag{45}$$

$$Q^{empty} G_y + \sum_{i \in I} \sum_{j \in J} p_i^y q_j x_{kij} \leq (\sum_{i \in I} \sum_{j \in J} q_j x_{kij} + Q^{empty})(G_y + \tau_1^y) \qquad k \in K \tag{46}$$

$$Q^{empty} G_y + \sum_{i \in I} \sum_{j \in J} p_i^y q_j x_{kij} \geq (\sum_{i \in I} \sum_{j \in J} q_j x_{kij} + Q^{empty})(G_y - \tau_2^y) \qquad k \in K \tag{47}$$

Constraints (44)–(47) may be added to the ILP models of the previous sections when needed. We decided to computationally test their impact by adding them to model (19)–(29) of Section 6.1 (single pallets), and to model (30)–(43) of Section 6.2 (double pallets). In our tests we set (G_x, G_y) as the middle point of the truck, τ_1^x and τ_2^x equal to the width of one position, τ_1^y and τ_2^y equal to the length of one position, and $Q^{empty} = 3500$ Kg. We also disregarded the use of the initial heuristic, because it does not handle the center of gravity restriction.

Table 8 shows the results that we obtained by adding the center of gravity constraints to the case of single pallets. With respect to the original model, see Table 6, we can notice that two more instances are now optimally unsolved in class B and one less in class D. The overall gap has increased consistently, from 5 to 11 containers in total. This is mostly due to instance 106 of class D, whose final gap is of 6 containers, whereas for the other instances the gap is of just one container. For the 6 unsolved instances, the mathematical model improves the lower bound found in the initialization phase just once. We can conclude that the inclusion of the center of gravity has a large impact on the behavior of the single pallet model.

Table 8: Computational results with single pallet model plus center of gravity

(a) Aggregate results (total/average values)

class	# inst.	missed	gap	L	U	nodes	sec	nodes$_{opt}$	sec$_{opt}$
A	27	0	0	8.67	8.67	45	0.33	45	0.33
B	53	3	3	9.53	9.58	19568	17.36	12	0.51
C	20	0	0	9.60	9.60	1637	3.91	1637	3.91
D	11	3	8	18.45	19.18	33594	86.16	252	7.12
avg/sum	111	6	11	10.22	10.32	12978	17.61	348	1.62

(b) Details for the unsolved instances

class	number	$\|J\|$	$\sum_j n_j$	L_{init}	U_{init}	L	U	nodes
B	84	8	3278	14	19	14	15	195900
B	60	4	3494	8	10	8	9	449157
B	103	4	3740	11	15	11	12	391446
D	93	15	2397	7	10	7	8	230822
D	95	15	4090	12	16	12	13	93642
D	106	25	4088	15	22	16	22	43054

Table 9 gives instead the results that we obtained for the case of double pallets. The inclusion of the additional constraints has in this case a lower impact, because the problem was already quite difficult to solve. We can notice a decrease in both the number of unsolved instances, from 4 to 3, and in the total gap, from 8 to 7. Some instances remain very difficult, as instance 106 whose gap is of 5 bins. We finally notice that the behavior of the models depend on the value assumed by the parameters τ, and allowing larger tolerance may decrease the number of containers required to find feasible solutions.

7.2 Minimizing the loading/unloading effort

The models presented in the previous sections aim at minimizing the number of containers used for the loading. However, solutions using the minimum number of containers may have a large difference in terms of the number of pallets used to load the items. A smaller number of pallets is obviously preferable, because it can lead to a reduction in the time required for the loading/unloading operations. The number of pallets can thus be seen as a secondary objective function with respect to the number of containers. Preliminary tests showed, however, that embedding these two functions in the same ILP model can lead to very poor computational results. We thus adopted a two-level approach: we first compute the minimum number of containers, say U, by using one of the above models, and then we compute the minimum number of pallets by solving a new ILP model in which the number of containers cannot exceed U.

For the single pallet model of Section 6.1, the second ILP model to be solved consists of

$$\min \quad \sum_{i \in I} \sum_{k \in K} z_{ki}, \tag{48}$$

Table 9: Computational results with double pallet model plus center of gravity

(a) Aggregate results (total/average values)

class	# inst.	missed	gap	L	U	nodes	sec	nodes$_{opt}$	sec$_{opt}$
A	27	0	0	8.67	8.67	54	1.36	54	1.36
B	53	1	1	9.55	9.57	3770	10.27	355	4.71
C	20	1	1	9.70	9.75	445	18.08	30	3.48
D	11	1	5	18.91	19.36	2064	73.11	1691	50.49
avg/sum	111	3	7	10.29	10.35	2098	15.74	346	7.90

(b) Details for the unsolved instances

class	number	$\|J\|$	$\sum_j n_j$	L_{init}	U_{init}	L	U	nodes
B	103	4	3740	11	15	11	12	181315
C	37	19	2064	6	8	6	7	8332
D	106	25	4088	15	22	17	22	5791

subject to (20)–(29) and

$$\sum_{k \in K} y_k \leq U \tag{49}$$

Similarly, the minimization of the loading/unloading effort for the double pallets model of Section 6.2 requires to minimize $\sum_{i \in I^2} \sum_{k \in K} z_{ki}$, subject to (31)–(43) and (49).

The results obtained by running the additional ILP models are presented in Table 10. The original models for the single and double pallets cases have been run for 300 seconds as before. Then the additional ILPs have been run for 300 seconds more. Apart from the name of the class and the number of instances, the leftmost columns of the table give some details of the original solutions found when minimizing the number of containers. Namely, L and U give, respectively, the average lower and upper bound values on the number of containers (also refer to Table 6), and $U^0_{pallets}$ the average number of pallets used. The rightmost columns give the results of the additional ILP model. Namely, $L_{pallets}$ and $U_{pallets}$ give the lower and upper bound values on the minimum number of pallets, %diff the percentage difference with respect to the original solutions, computed as $100(U^0_{pallets} - U_{pallets})/U^0_{pallets}$, # opt is the number of instances solved to proven optimality, and nodes and sec give, respectively, the average number of nodes and seconds elapsed. The top part of the table refers to the single pallet model, and the bottom part to the double pallets one.

For the single pallet case, it can be noticed that the original solutions use about 17.5 pallets per container (between 15 and 17 pallets on average for the easy instances of Classes A and B, and between 19 and and 21.5 on average for the instances of Classes C and D). The second ILP model imposes a relevant burden to the solver, because only 60 out of 111 instances are optimally solved. However, it is very effective in reducing the number of used pallets, which is decreased by 10% on average, leading to only 16 pallets on average per container. Considering the overall set of instances, this amounts to a reduction of from 18929 to 17367 pallets, i.e., more than 1500 pallets. An example of this optimization process is depicted in Figure 5, where the same container load is shown before and after the pallets minimization. The load on the left contains some pallets of short height, formed by just a few layers. The load on the right is characterized instead by a smaller number of very high

Table 10: Minimization of the unloading effort

(a) The case of single pallets

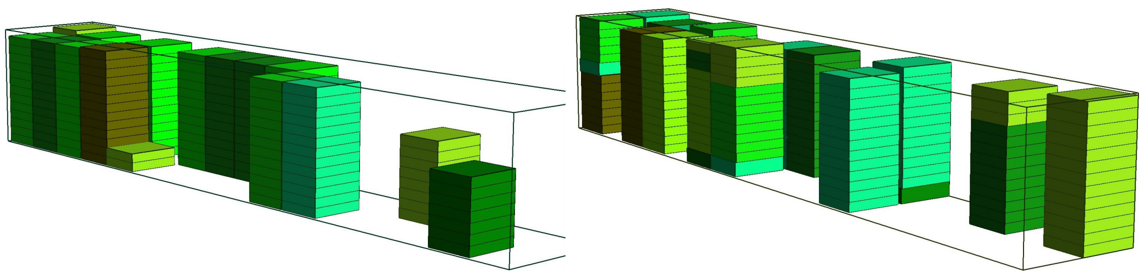
class	# inst.	Original model L	U	$U^{init}_{pallets}$	Minimize unloading effort $L_{pallets}$	$U_{pallets}$	%diff	# opt	nodes	sec
A	27	8.19	8.19	138.15	112.19	112.96	20%	17	197980	223.78
B	53	9.00	9.02	139.11	126.17	127.32	9%	28	337159	287.46
C	20	9.25	9.25	197.95	174.35	192.75	3%	11	79579	237.07
D	11	17.91	18.27	351.55	245.73	337.64	4%	4	48299	270.50
avg/sum	111	9.73	9.77	170.53	143.30	156.46	10%	60	228268	261.21

(b) The case of double pallets

class	# inst.	Original model L	U	$U^{init}_{pallets}$	Minimize unloading effort $L_{pallets}$	$U_{pallets}$	%diff	opt	nodes	sec
A	27	8.19	8.19	273.15	209.63	211.74	24%	15	112735	270.64
B	53	9.04	9.04	287.66	229.89	248.60	16%	35	52031	204.38
C	20	9.35	9.40	429.00	367.55	388.15	10%	13	50498	226.87
D	11	18.36	19.00	741.73	657.64	673.18	8%	5	19458	291.02
avg/sum	111	9.81	9.88	354.59	292.15	306.86	16%	68	63293	233.14

pallets. Note that both loads have empty positions and pallets spread evenly on the surface. This is a typical configuration that arises to satisfy maximum weight capacity and axle weight restrictions when the loaded products are heavy.

For the double pallets case the reduction is even more evident. The original solutions use about 36 pallets per containers, whereas the solutions optimized for the loading/unloading effort only 31. This amounts to a percentage reduction of about 16%, and the total number of used pallets passes from 39360 to 34061 pallets, i.e., more than 5000 pallets are saved. Further improved solutions are likely to be found, as can be noticed by the remaining gap between the values of $U_{pallets}$ and $L_{pallets}$, but still we can conclude that the proposed ILP models are effective in reducing the loading/unloading effort.



(a) Solution of the single pallet model (b) Solution after minimization of the number of pallets

Figure 5: Same container load before and after the minimization of the number of pallets

8 Conclusions

Container loading problems are an important class of optimization problems that model several real-world situations. The literature has largely studied the case of a single container loading, whereas in this paper we focus on the less treated case of multi container loading. The aim is to minimize the number of containers used to transport from a single origin to a single destination a set of items that have been previously packed in layers. We decided to pursue optimization by the use of simple, yet effective, Integer Linear Programming models.

The problem definition depends from the real-world loading situation at hand. Here we studied the real-world optimization problem that derives from the everyday activity of ORTEC [23], one of the largest European providers of logistics planning and optimization services. The problem is very complex, so we decided to proceed in an analytical way, starting from the basic loading issues and then adding new constraints, and/or secondary objective functions, one at a time. In this way we gained several insights in the difficulty of this class of problems.

The original test set of 111 instances was first solved as a basic packing model, then with the addition of axle-weight constraints, of pallet bases in two loading configurations, and of center of gravity constraints on top of the pallet bases configurations. Out of the 666 instances addressed in these tests, 643 were solved to proven optimality in less than 5 minutes on a standard PC. For the 23 unsolved instances, the gap between the obtained upper and lower bound values was higher than one container in just four cases (one for instance 17 and three for instance 106). Additional tests were also performed to minimize the number of pallets used, showing that high gains can be achieved with limited computational efforts. We can conclude that the proposed Integer Linear Programming models are practical, effective, and flexible tools for handling multi container loading problems.

Acknowledgements

Manuel Iori gratefully acknowledges partial financial support by CAPES/Brazil under Grant PVE no. A007/2013. Maria Teresa Alonso, Francisco Parreño, Jose Manuel Tamarit and Ramon Alvarez-Valdes gratefully acknowledge partial financial support by the Spanish Ministry of Science and Technology, DPI2011-24977, by Project PPII-2014-011-A, Consejeria de Educacion y Ciencia, Junta de Comunidades de Castilla-La Mancha, and by Generalitat Valenciana, PROMETEO/2013/049.

We kindly thank ORTEC [23] for providing us with the test instances.

References

References

[1] O.C.B. Araujo and V.A. Armentano. A multi-start random constructive heuristic for the container loading problem. *Pesquisa Operacional*, 27(2):311 – 331, 08 2007.

[2] E.E. Bischoff and M.D. Marriott. A comparative evaluation of heuristics for container loading. *European Journal of Operational Research*, 44(2):267 – 276, 1990.

[3] E.E. Bischoff and M.S.W. Ratcliff. Issues in the development of approaches to container loading. *Omega*, 23(4):377 – 390, 1995.

[4] A. Bortfeldt. A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. *Computers and Operations Research*, 39(9):2248 – 2257, 2012.

Table 11: Details of the real-world benchmark set

class	instance	Products	Layers	L_{init}	U_{init}	class	instance	Products	Layers	L_{init}	U_{init}
A	28	15	1164	7	7	B	58	16	2285	10	11
A	44	10	1170	7	7	B	89	11	2580	11	12
A	30	12	1248	7	7	B	43	1	2600	6	7
A	39	33	1288	7	7	B	36	18	2748	12	13
A	73	40	1314	7	7	B	75	11	2761	11	12
A	24	11	1362	7	7	B	38	1	2772	6	7
A	62	11	1374	8	8	B	55	9	2890	12	13
A	79	4	1380	7	7	B	13	1	2968	7	8
A	6	15	1386	7	7	B	87	6	3036	12	13
A	88	5	1452	7	7	B	35	18	3135	14	15
A	7	18	1482	8	8	B	84	8	3278	14	15
A	23	21	1536	8	8	B	60	4	3494	8	9
A	22	42	1536	8	8	B	80	1	3696	8	9
A	29	15	1596	9	9	B	103	4	3740	11	12
A	25	12	1884	9	9	B	111	9	3951	17	18
A	90	5	1920	8	8	B	67	1	4158	9	10
A	53	12	2016	9	9	B	74	1	4320	11	12
A	18	16	2078	9	9	B	41	1	4620	10	11
A	107	5	2251	9	9	B	65	1	4680	11	12
A	109	8	2332	9	9	B	66	1	4680	12	13
A	91	44	2737	7	7	B	86	1	5040	13	14
A	83	1	2748	7	7	B	81	1	5082	11	12
A	56	4	2808	13	13	B	64	1	5544	12	13
A	100	1	2880	8	8	B	50	1	11144	21	22
A	105	1	3564	9	9	C	98	142	1905	6	8
A	96	1	5200	8	8	C	52	3	1920	6	8
A	51	1	6104	12	12	C	34	22	1945	5	7
B	92	4	480	6	7	C	37	19	2064	6	8
B	57	3	705	6	7	C	15	5	2154	9	11
B	97	1	728	7	8	C	99	2	2262	6	8
B	1	62	1088	6	7	C	85	3	2304	7	9
B	68	29	1199	6	7	C	76	4	2340	5	7
B	101	42	1290	6	7	C	77	4	2350	5	7
B	16	15	1296	6	7	C	78	4	2360	5	7
B	82	4	1378	6	7	C	21	24	3035	11	13
B	69	32	1406	6	7	C	19	21	3102	13	15
B	47	7	1410	6	7	C	9	6	3456	9	11
B	61	12	1422	8	9	C	8	7	3584	9	11
B	4	69	1473	8	9	C	110	1	3840	8	10
B	108	4	1500	6	7	C	10	16	4092	18	20
B	42	4	1536	6	7	C	27	1	4224	9	11
B	104	45	1554	7	8	C	102	1	4380	10	12
B	71	4	1566	6	7	C	3	23	5258	23	25
B	12	13	1639	7	8	C	54	1	6006	13	15
B	5	2	1728	6	7	D	31	1	482	6	9
B	33	13	1792	8	9	D	93	15	2397	7	10
B	94	15	1860	6	7	D	70	10	3808	15	18
B	63	17	1862	8	9	D	26	1	4992	10	13
B	45	7	1929	6	7	D	20	27	6237	26	29
B	40	5	1968	7	8	D	17	25	6248	26	29
B	59	14	2002	8	9	D	49	16	6897	27	30
B	32	22	2024	9	10	D	95	15	4090	12	16
B	72	8	2112	8	9	D	14	5	4096	11	15
B	11	20	2136	9	10	D	48	18	10197	41	46
B	2	71	2237	11	12	D	106	25	4088	15	21
B	46	4	2256	10	11						

[5] A. Bortfeldt and H. Gehring. A hybrid genetic algorithm for the container loading problem. *European Journal of Operational Research*, 131(1):143 – 161, 2001.

[6] A. Bortfeldt, H. Gehring, and D. Mack. A parallel tabu search algorithm for solving the container loading problem. *Parallel Computing*, 29(5):641 – 662, 2003.

[7] A. Bortfeldt and G. Wäscher. Constraints in container loading. A state of the art review. *European Journal of Operational Research*, 229(1):1 – 20, 2013.

[8] K.F. Doerner, G. Fuellerer, M. Gronalt, R.F. Hartl, and M. Iori. Metaheuristics for the vehicle routing problem with loading constraints. *Networks*, 49(4):294 – 307, 2007.

[9] J. Egeblad, C. Garavelli, S. Lisi, and D. Pisinger. Heuristics for container loading of furniture. *European Journal of Operational Research*, 200(3):881 – 892, 2010.

[10] T. Fanslau and A. Bortfeldt. A tree search algorithm for solving the container loading problem. *INFORMS Journal on Computing*, 22(2):222–235, 2010.

[11] H. Gehring and A. Bortfeldt. A genetic algorithm for solving the container loading problem. *International Transactions in Operational Research*, 4(5-6):401 – 418, 1997.

[12] R.W. Haessler and F.B. Talbot. Load planning for shipments of low density products. *European Journal of Operational Research*, 44(2):289 – 299, 1990.

[13] M. Iori and S. Martello. Routing problems with loading constraints. *TOP*, 18(1):4 – 27, 2010.

[14] M. Iori, J.J. Salazar González, and D. Vigo. An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science*, 41:253–264, 2007.

[15] B. Jacob and V. Feypell-de La Beaumelle. Improving truck safety: Potential of weigh-in-motion technology. *IATSS Research*, 34(1):9 – 15, 2010.

[16] Z. Jin, K. Ohno, and J. Du. An efficient approach for the three-dimensional container packing problem with practical constraints. *Asia-Pacific Journal of Operational Research*, 21(3):279–295, 2004.

[17] L. Junqueira, R. Morabito, and D.S. Yamashita. Three-dimensional container loading models with cargo stability and load bearing constraints. *Computers and Operations Research*, 39(1):74 – 85, 2012.

[18] A. Lim, H. Ma, C. Qiu, and W. Zhu. The single container loading problem with axle weight constraints. *International Journal of Production Economics*, 144(1):358 – 369, 2013.

[19] A. Lim, B. Rodrigues, and Y. Wang. A multi-faced buildup algorithm for three-dimensional packing problems. *Omega*, 31(6):471 – 481, 2003.

[20] R. Morabito and S. Morales. A simple and effective recursive procedure for the manufacturer's pallet loading problem. *Journal of the Operational Research Society*, 49(8):819 – 828, 1998.

[21] R. Morabito, S.R. Morales, and J.A. Widmer. Loading optimization of palletized products on trucks. *Transportation Research Part E: Logistics and Transportation Review*, 36(4):285 – 296, 2000.

[22] A. Moura and J.F. Oliveira. A GRASP approach to the container-loading problem. *IEEE Intelligent Systems*, 20(4):50–57, July 2005.

[23] ORTEC. company. `www.ortec.com`, 2015. [Online; accessed 22-April-2015].

[24] D. Pisinger. A minimal algorithm for the bounded knapsack problem. *INFORMS Journal on Computing*, 12(1):75 – 82, 2000.

[25] H. Pollaris, K. Braekers, A. Caris, G. Janssens, and S. Limbourg. Capacitated vehicle routing problem with sequence-based pallet loading and axle weight constraints. *EURO Journal on Transportation and Logistics*, pages 1–25, 2015.

[26] S. Takahara. Loading problem in multiple containers and pallets using strategic search method. In VicenÃ§ Torra, Yasuo Narukawa, and Sadaaki Miyamoto, editors, *Modeling Decisions for Artificial Intelligence*, volume 3558 of *Lecture Notes in Computer Science*, pages 448 – 456. Springer Berlin Heidelberg, 2005.

[27] J. Terno, G. Scheithauer, U. Sommerweiss, and J. Riehme. An efficient approach for the multi-pallet loading problem. *European Journal of Operational Research*, 123(2):372 – 381, 2000.

[28] G. Wäscher, H. Haußner, and H. Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3):1109 – 1130, 2007.