View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Repositório Institucional da Universidade de Aveiro

FernUniversität in Hagen

A two-stage packing procedure for a Portuguese trading company

Ana Moura, Andreas Bortfeldt

Diskussionsbeitrag Nr. 457
November 2010

Diskussionsbeiträge der Fakultät für Wirtschaftswissenschaft
der FernUniversität in Hagen

Herausgegeben vom Dekan der Fakultät

Alle Rechte liegen bei den Autoren

A two-stage packing procedure for a Portuguese trading company

Ana Moura, Andreas Bortfeldt

Abstract:

This case study deals with a two-stage packing problem that has to be solved in the daily distribution process of a Portuguese trading company. At the first stage boxes including goods are to be packed on pallets while at the second stage these pallets are loaded into one or more trucks. The boxes have to be transported to different customers and the actual goal is to guarantee a sufficient utilization of the truck loading spaces. A two-stage packing procedure is proposed to cover both problem stages. First boxes are loaded onto pallets using a well-known container loading algorithm. Then trucks are filled with loaded pallets by means of a new tree search algorithm. The applicability and performance of the two-stage approach was evaluated with a set of instances that are based on actual company data.

Key words:

Packing, Container Loading, Bin Packing, heuristic, GRASP, tree search.

Fakultät für Wirtschaftswissenschaft, FernUniversität in Hagen

Profilstr. 8, D-58084 Hagen, BRD

Tel.: 02331/987–4433

Fax: 02331/987–4447

E-Mail: ana.moura@ua.pt, andreas.bortfeldt@fernuni-hagen.de

A two-stage packing procedure for a Portuguese trading company

Ana Moura[1,2], Andreas Bortfeldt[3]

[1]UA – Universidade de Aveiro, Portugal
[2]INESC-Coimbra – Instituto de Engenharia de Sistemas e Computadores, Coimbra, Portugal
[3]University of Hagen, Germany

1 Introduction

The Portuguese company Oliveira & Irmão (O&I) delivers sanitary articles and other goods to domestic and foreign customers. In the daily distribution process several decision problems regarding packing goods and routing trucks are to be solved. Up to now the daily packing and routing decisions are made by dispatchers throughout. The authors were asked by O&I to perform a case study and to examine whether an automated decision support tool could help improving the daily decisions in the distribution process.

Among the decision problems in the daily distribution there is a two-stage packing problem: in a first step boxes with goods are to be packed onto pallets while in the second step these pallets are to be loaded into trucks. Solving this problem should ensure a sufficient utilization of the truck loading spaces used for transportation. At the same time several side-constraints have to be observed, e.g., the so-called LIFO (last in, first out) constraint. It stipulates that no reloading of boxes must take place at the different customer locations of a given truck route. Little attention has been paid so far to this two-stage packing problem in the literature and so the paper at hand deals with the development of an effective solution procedure for it.

The remainder of the paper is structured as follows: in Section 2 the company and its distribution process are introduced. In Section 3 the two-stage packing problem mentioned above is formulated in detail. In Section 4 a general packing approach is devised for this task and in Sections 5 and 6 solution procedures for the two sub-problems of the two-stage packing problem are specified. Section 7 reports on computational results. Finally, in Section 8 some conclusions are drawn.

2 The company and its distribution process

Founded in 1954, the incorporated company Oliveira & Irmão is a long-established Portuguese company that is located in Aveiro, a city in the north of Portugal. Its core business is manufacturing plastic flushing cisterns and plumbing mechanisms, and marketing sanitary and central heating equipment. The company has experienced rapid growth in recent years and now half of the turnover comes from exports. O&I has turned itself from a small business into a medium-sized European company, with a strong customer base outside Portugal. While still most of the customers are located in Portugal, there are customers in Europe, Middle East and Asia as well.

2.1 Basic facts concerning distribution

The daily distribution is based on a set of customer orders for a given day. The company provides their products by means of a product catalog. A customer order comprises a list of order positions and each order position consists of a product (selected from the catalog) and the demanded quantity (number of items). Moreover, an order has a delivery date and an appropriate time window. Typically the customers are served more or less periodically. Most of the customers get their goods delivered by trucks while some customers are served by means of standard containers. All goods are shipped in rectangular boxes and these boxes are the subject of packing. Each box may contain one or more items of a single product type.

Within the whole distribution process the company strives for two main goals. On the one hand the customers are to be served punctually, i.e. the orders placed for a given day should reach their purchasers at that day as far as possible. On the other hand the distribution should be carried out efficiently. This requires above all that the capacity of transportation means (trucks, containers) is efficiently used, i.e. unused loading space should be minimized.

There are three groups of daily orders that are dealt with strictly separately: (1) the orders of foreign customers who get their goods by trucks, (2) the orders of foreign customers who want to be served by containers and (3) the orders placed by domestic customers. These order groups are handled in a very different way and so they constitute three partial processes of the total daily distribution. The two-stage packing problem that is dealt with in this paper occurs in the distribution of goods for foreign customers by means of trucks. Hence, in the following the processing of orders of the first group is characterized more detailed and roughly compared to the processing of the orders of the other two groups.

2.2 Serving foreign customers by means of trucks

Only few foreign customers are to be served by trucks each day. These customers belong to groups of up to five customers whose members are far away from the members of all other groups. Hence defining suitable routes for a given day is a simple matter that can be well done by a (human) dispatcher. Moreover, routing and packing can be planned in two separate and consecutive steps without any negative impact on the quality of decision making. First some routes have to be stipulated and afterwards the stowage planning is performed.

Only one type of trucks (without trailer) is available and thus all trucks have the same rectangular loading space. Trucks are not loaded with boxes directly. Instead first identical pallets are packed with boxes and then these pallets are loaded into trucks (two-stage packing). Since the pallets are packed with boxes by means of robots, rather simple packing patterns are to be chosen. In particular only congruent boxes are to be packed on one pallet. A stacking of pallets within trucks is allowed. In general a truck contains pallets of several customers and the LIFO constraint (see Section 1) has to be satisfied.

All in all, in processing orders of the first group no automated decision support is needed for the routing of trucks while an effective procedure for solving the two-stage packing task could be a helpful tool for the O&I dispatchers.

In the subsequent comparison of the distribution processes for the three order groups only the main differences are mentioned. However it should become clear enough that very different tools are needed to enhance the decision making within these partial processes.

If foreign customers are served by containers (order group 2) each container is loaded by boxes of a single customer. Containers are packed with boxes directly while pallets are not used (one-stage packing). Figure 1 illustrates this difference in the distribution of orders of group 1 and 2, respectively.

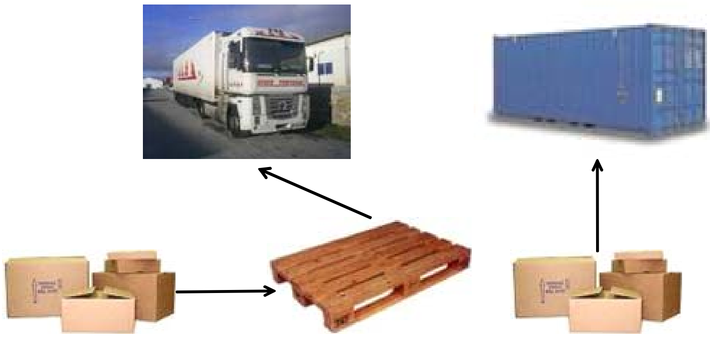

Figure 1 – 2-stage packing (order group 1) and 1-stage packing (order group 2)

O&I has to serve some dozens domestic customers each day by trucks. Hence in the distribution of orders of group 3 the routing of trucks is a non-trivial task and a computer-based solution procedure could save considerable costs. The operation efficiency of the distribution process for this order group depends on the quality of the routing as well as the packing decisions. Moreover, an artificial separation of routing and packing will have a negative influence on the operation efficiency. So the planning of routes and the stowage planning on the other hand should be done in a combined one-level solution process – differently to order group 1.

2.3 Proceeding of dispatchers and conflicting goals

From now on only the distribution process for foreign customers who get their goods by trucks (order group 1) is considered. The flowchart in Figure 2 shows the daily planning for these orders. In particular it is considered in the flowchart that a trade-off can occur between the two main goals of the distribution mentioned above. If all orders (in group 1) of a given day are planned without exception it may happen that some of the used truck loading spaces are not sufficiently utilized. It is assumed now that the daily planning is generally executed by a dispatcher while the stowage planning is supported by a computer-based solution procedure.

The flowchart is commented as follows:

- First the order stock is initialized by the set of all daily orders (of group 1) in order to satisfy the service goal of a punctual supply of all customers without exception. The dispatcher then defines a routing plan consisting of one or more routes of customers so that each of the orders is involved by exactly one route. The routing can result in, e.g., two routes where one route serves three Dutch customers while the second one visits two Italian customers.
- For each route of the current routing plan the corresponding two-stage packing problem is then solved by means of an automated packing procedure. The resulting total solution – comprising the routing plan and the packing plans for all routes – is stored (together with all input data).
- The dispatcher decides whether the total solution just generated can be accepted in terms of efficiency. A solution may be inefficient for two reasons: (1) The volume utilization of one or more truck loading spaces is too low. (2) A better clustering and sequencing of the customers could lead to a considerably smaller total travel distance. For example, suppose that two trucks are needed to serve two Italian customers located in Milan and Rome and each truck is loaded by the goods of only one of them. Let originally one route be given, namely Aveiro–Milan–Rome–Aveiro. Obviously the segmentation of this route, resulting in the routes Aveiro–Milan–Aveiro and Aveiro–Rome–Aveiro, yields a considerably saving of total travel distance.

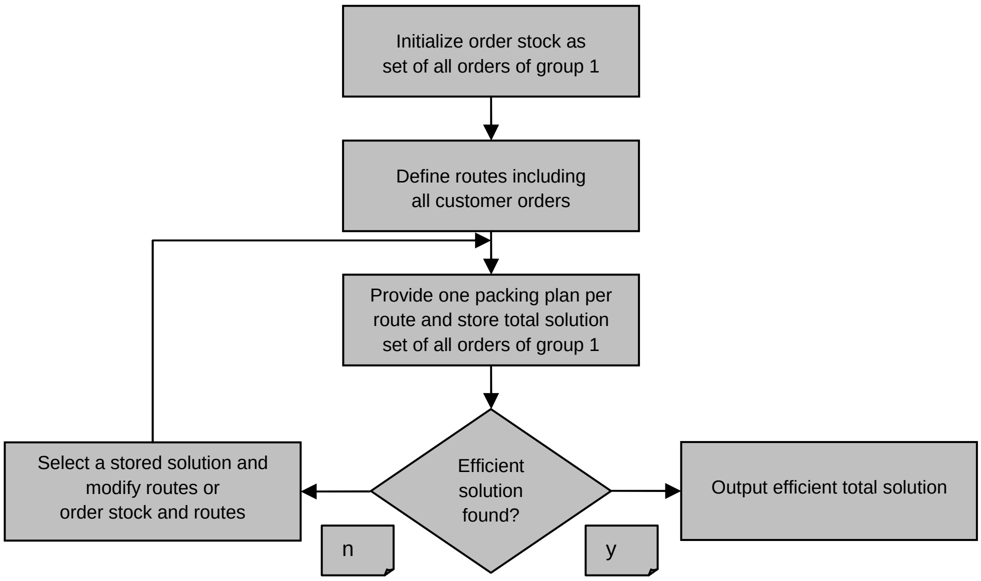

Figure 2 – Daily planning of the distribution process concerning order group 1

- If the total solution just generated is too inefficient then the dispatcher selects one of the stored total solutions including the corresponding input data. Now either the routes are changed while the order stock remains unchanged or the order stock is modified and the routes are also updated. Of course, the orders of a given day should only be changed after some negotiations with the affected customers. Different modifications of an order can be agreed: an order can be postponed to another day or the set of ordered goods may be extended etc. The modification of orders of a current day generally aims at a satisfying compromise between the two main goals of the distribution – punctual delivery of goods and efficiency of transportation.
- After the input data and routes of a solution were changed a new total solution is computed, i.e. new packing plans are provided that are based on the modified order stock (if so) and the modified routing plan. The planning is finished after the dispatcher has found the first efficient solution.

3 The two-stage packing problem

The two-stage packing problem occurring in the distribution of goods for foreign customers who are served by trucks is now defined with all details as follows.

Let be given the following entities:

- A single truck type with a rectangular loading space and a single pallet type; the numbers of available trucks and pallets are not limited.
- A route (i.e. sequence) of customers to be served and one order per customer. Each order comprises one or more box types and a corresponding number of demanded boxes per type.

It is required to pack all boxes on pallets and then to load the pallets in a minimal number of trucks. In doing so all side-constraints that are specified for both packing stages have to be observed.

When boxes are packed onto pallets the following constraints are to be met:

(C1) Each pallet can be packed only with boxes of a single customer. Furthermore each pallet can be packed only with boxes of the same type, i.e. congruent boxes.

(C2) Each box must lie completely on its pallet, i.e. it is not allowed that a box does protrude over the lateral borders of its pallet.

(C3) The loading height of a pallet is given by the distance between the highest top area of a box and the bottom of the pallet (the height of an empty pallet is viewed as part of the loading height). The loading height must not exceed a given maximum loading height which is stipulated individually per customer.

(C4) The bottom area of each box has to be fully supported (i.e. to 100%) from below by the bottom area of the pallet or by the top areas of other boxes.

(C5) For each box a single dimension is allowed to act as the height of the box.

The rectangular loading space of a truck is organized in two strips, called "left strip" and "right strip" (see Figure 3), with equal dimensions. Each strip consists of *nStPl* stack places where stack place 1 lies at the driver cabin while stack place *nStPl* of a strip lies at the rear (near to the door) of the loading space. At each stack place several pallets can be stacked on top of each other. The dimensions of the truck loading space and of the pallets are adjusted so that the larger horizontal dimension of a pallet (pallet length) is a bit smaller than the width of a strip. Hence a pallet is always to set on a stack place in such a way that the pallet length lies parallel to the width of the appropriate strip. It results that the position of a pallet can be described unambiguously by four indices: index of truck (1,2,…), index of strip (left or right), index of stack place (1,…, *nStPl*) and index of position in stack (1,2,…) where value 1 means that the pallet lies on the bottom of a stack.

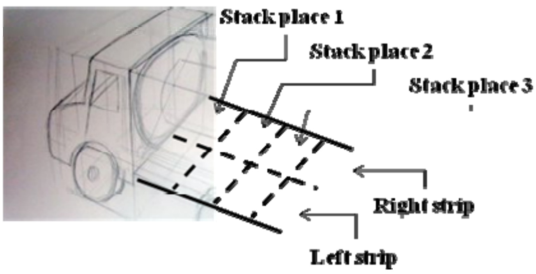

Figure 3 – Organization of a truck loading space

When pallets are loaded into one or more trucks the following constraints are to be satisfied:

(C6)The height of a stack of one or more pallets is defined as the sum of the loading heights of all pallets and it must not exceed the height of the truck loading space (truck height).

(C7)The total top area of a loaded pallet is defined as the sum of the top areas of all boxes whose top area does reach the loading height of the pallet (see (C3)). The support rate *supTopArea* of a loaded pallet A is given as the quotient (total top area of pallet A) / (bottom area of pallet type) (in %). A further pallet may only be stacked upon pallet A if *supTopArea*(A) is not smaller than a given support rate *minSupTopArea* (in %). Otherwise pallet A is called a cap pallet and has to be packed on the top of a stack.

(C8)The boxes and pallets have to be stowed in such a way that no reloading of boxes is necessary at the different customer locations of a given truck route (LIFO constraint). This constraint shows two aspects. Let customers A and B be served in the same route and let customer A be visited first. Then a pallet with boxes for customer B must not lie upon a pallet for customer A in the same stack of a truck. Moreover, if a pallet for customer A is packed then a pallet for customer B must not be packed on a stacking place with higher index in any strip of the same truck (cf. Figure 3).

All given boxes of all orders have to be packed, i.e. the output of the packing process is fixed in a sense. On the other hand the number of trucks, acting as input of the process, is to be minimized. Hence the two-stage packing problem can be viewed as a three-dimensional Bin Packing problem (3D-BPP). Note that if two solutions differ with regard to the needed number of stacks they need not to differ regarding the needed number of trucks. Therefore, a second optimization criterion is introduced for the problem of loading pallets into trucks: if two solutions need the same number of trucks the solution with the lower maximum stack height is considered the better one. Obviously, this criterion will strengthen the load stability of generated packing plans.

4 A general approach to the two-stage packing problem

A straightforward solution approach to the two-stage packing problem was drafted:

- For both packing stages a separate solution procedure is devised and each of the procedures is called once only per problem instance. By means of the first procedure, called LBOP (as loading boxes onto pallets) all boxes are packed on pallets and by means of the second procedure, called LPIT (as loading pallets into trucks) all pallets are loaded in a minimum number of trucks. The procedures are called consecutively and there is no feedback, i.e. after the boxes were packed the pallets are no longer repacked.
- The procedure LBOP is based on a container loading procedure that serves to pack a single container with rectangular boxes. This allows for reverting to existing packing procedures (see below). The pallets are loaded separately and the container loading procedure is applied once per pallet.
- After the pallets were loaded a one-dimensional packing problem remains to be solved. Each truck offers the same number of stack places (cf. Section 3) and the maximal height of each stack is the same. Each stack can be filled by one or more pallets with known (and generally different) pallet heights. In order to minimize the number of needed trucks the pallets are to be packed in such a way that the number of needed stacks becomes as small as possible. Hence the packing problem of the second stage turns out to be a (variant of) one-dimensional Bin Packing Problem (1D-BPP). Since the 1D-BPP has been successfully tackled in the literature by means of problem specific tree search procedures (see below) the procedure LPIT for loading pallets into trucks is devised as a tree search procedure too.

In the sequel the procedures LBOP and LPIT are specified in detail.

5 Loading boxes onto pallets

To specify the procedure LBOP for loading boxes onto pallets first a container loading method is selected. The Container Loading Problem (CLP) requires packing a subset of a given set of boxes in a single rectangular container with given dimensions so that the volume utilization is as large as possible. The 3D-CLP is NP-hard in the strict sense and is regarded in practice as very difficult (cf. Pisinger 2002). Most of the solution methods that were proposed in recent years are heuristics and meta-heuristics. In Table 1 multiple successful CLP solution methods are listed.

Table 1 – Recent solution methods for the three-dimensional Container Loading Problem

Authors, source	Type of method
Bortfeldt and Gehring (2001)	Genetic algorithm
Pisinger (2002)	Tree Search
Eley (2002)	Tree Search
Bischoff (2004)	Local Search
Mack et al. (2004)	Hybrid (Tabu Search, Simulated Annealing)
Moura and Oliveira (2005)	GRASP
Parreño et al. (2007)	GRASP
Fanslau and Bortfeldt (2010)	Tree Search

The CLP solution methods quoted in Table 1 behave differently in at least three regards. First they achieve different levels of container utilization, secondly they have different abilities regarding additional side-constraints and thirdly the methods differ in terms of computational cost. The method proposed by Moura and Oliveira (2005) was chosen as the basic module for the procedure LBOP. On the one hand this method achieves a good volume utilization for homogeneous box sets with a single type of boxes (cf. constraint (C1) above). On the other hand the method from Moura and Oliveira proved to be particularly sensitive in terms of load stability and this is a further important issue in the given situation. Last but not least solutions are provided very rapidly and this could encourage the O&I dispatchers to apply the corresponding decision support tool directly when negotiating with customers to perform what-if-computations etc.

As stated in Section 3 the boxes of different customers and/or different box types have to be packed on different pallets (cf. constraint (C1)). So it is sufficient to specify how the boxes of a single customer and of a single box type are loaded on one or more pallets (see Figure 4). The CLP method from Moura and Oliveira is now denoted as module CLGR1. For each customer and each box type the belonging boxes are loaded separately following the shown flowchart.

Since module CLGR1 is a container loading procedure the constraints (C2) and (C3) (see Section 3) are automatically satisfied if the maximum loading height of a pallet is taken as the container height. Moreover, the module CLGR1 is able to observe the support constraint (C4) as well as the orientation constraint (C5) without any modification.

In the first packing stage there are two critical success factors. On the one hand pallets should be loaded in such a way that the mean volume utilization of all packed pallets is as large as possible. The volume utilization of a single loaded pallet is computed as the quotient (total volume of loaded boxes) / (volume of pallet space) (in %) where the pallet space volume is given as product (pallet bottom area) × (loading height – height of empty pallet) (see Section 3, (C3)). Furthermore the number of generated cap pallets (see Section 3, (C7)) should be as small as possible since a higher number of cap pallets will lead to a higher number of used stacks. Later it will be seen whether procedure LBOP does consider these critical success factors.

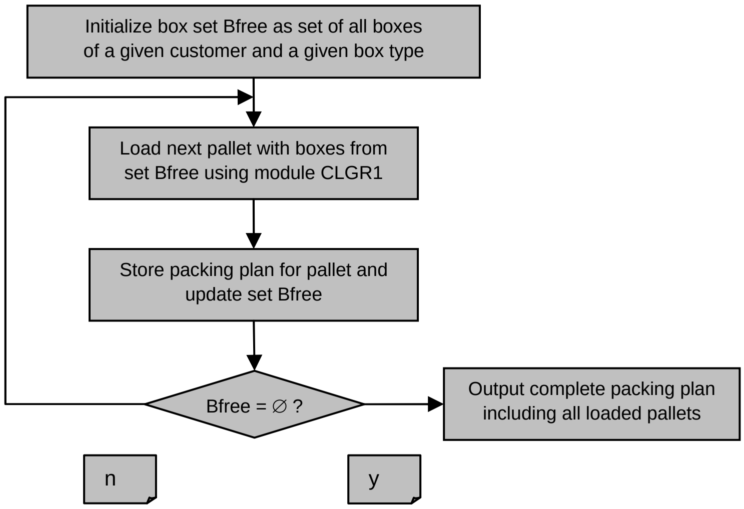

Figure 4 – Loading the boxes of a single customer and a single box type onto pallets

6 Loading pallets into trucks

All pallets that were packed with boxes in the first packing stage have to be loaded then into the minimum number of trucks during the second packing stage. Furthermore the constraints (C6) to (C8) and the second optimization criterion – minimization of the maximum stack height – must also be taken into account. In Table 2 the input and output data of the second packing stage are listed. Each *packed* pallet is given by its loading height, its total top area and the corresponding customer. Each *stacked* pallet is defined by a pallet index and by four indices regarding truck, strip, stack place and position in stack (see Section 3).

Table 2 – Input and output data of second packing stage

Input data	Output data
no. (*nC*) and sequence of customers	no. of loaded trucks (*nTrucks*)
truck height (*hT*),	maximum stack height (*maxStackHeight*)
pallet bottom area	set of stacked pallets
no. of stack places per strip (*nStPl*)	
requ. support rate per pallet (*minSupTopArea*)	
no. (*nP*) and set of packed pallets	

To our knowledge the considered problem of loading pallets into trucks has not been dealt with in the literature so far. However, it can be viewed as a special variant of 1D-BPP where the stack places act as bins with truck height as bin size and the heights of the packed pallets act as small pieces. Therefore, a tree search procedure was devised that adopts a well-known branching schema for the 1D-BPP. It is described now with its essential characteristics.

6.1 Schema of tree search

Solutions are constructed stepwise and in each step a solution is extended by stacking *one* further pallet. A depth first search is applied, i.e. the next solution to be extended is one of the existing partial solutions with the largest number of pallets already stacked. A pseudo-code representation of the tree search is shown in Figure 5.

```
extendSolution(nStackedPallets, solution)
    if nStackedPallets >= nP then  // solution complete
        update bestSolution where necessary;
    else // solution not complete
        generate nStVar stacking variants for pallet (nStackedPallets+1);
        oldSolution := solution;
        for i := 1 to nStVar do
            solution := oldSolution ⊕ ith stacking variant;
            extendSolution(nStackedPallets+1, solution);
        endfor;
    endif;
end.
```

Figure 5 – Schema of the tree search

Figure 5 needs some comments:

- The recursive function *extendSolution*() takes a partial solution and the corresponding number of stacked pallets as arguments. If the solution is already complete the best solution so far is updated where necessary. Otherwise the current partial solution is extended by stacking another pallet in alternative variants and the function is called again.
- To reduce the redundancy of the search the order in which the pallets are stacked is fixed. Before function *extendSolution*() is called for the first time (with zero values as arguments) the packed pallets are sorted according to the visiting sequence of the customers. Pallets that belong to customers to be visited last are stacked first (near the driver cabin) and vice versa. Pallets of the same customer are sorted according to the following criteria. A cap pallet has to be stacked before a non-cap pallet. If both of two pallets are either cap pallets or non-cap pallets then the pallet with greater loading height has to be stacked first.
- For the next pallet to be stacked (with index *nStackedPallets*+1) all feasible variants are determined to place this pallet on top of an existing stack (with one or more pallets). Additionally a new stack is opened in front of the existing ones and the current pallet is placed there. This proceeding corresponds to the branching schema applied to the 1D-BPP by Martello and Toth (1989). However, the number of tested stacking variants *nStVar* is limited by a parameter *maxSucc*. All feasible stacking variants are sorted according to the residual height left in the filled stack in ascending order. Then only the first *maxSucc* stacking variants with the lowest residual heights (in the stacks just filled) are considered. In this way stacking variants with highly utilized stacks are favored.

To limit the total effort of the procedure a time limit is used. Furthermore some bounds of the objective function serve to shorten the search.

6.2 Bounding the search

Two lower bounds are derived for the number of needed trucks. The first one (called static bound) refers to an "empty" solution, i.e. to a problem instance without any additional information. The second one (called dynamic bound) is based on a partial solution with one or more stacked pallets. Similarly, a static and a dynamic lower bound is derived for the second objective criterion, namely the maximum stack height.

a) Static bound of number of trucks

The following notation is agreed (see also Table 2). Let the customers be indexed from 1 to *nC* so that a customer *c* is visited as the *c*th customer. Let *hp*(*c*) denote the total loading height of all pallets of customer *c* and let *ncp*(*c*) be the number of cap pallets of customer *c* (*c* = 1,…,*nC*). Finally, let be $nst1(c) := \sum_{i>c} ncp(i)$ and $nst2(c) := \lceil \sum_{i \leq c} hp(i)/hT \rceil$, *c* = 0,1,...,*nC*.

Proposition 1

(i) $LBnStacks(c) := nst1(c) + nst2(c)$, *c* = 0,1,...,*nC*,
is a lower bound of the number of filled stacks.

(ii) $LBnStacks := \max\{ LBnStacks(c), c = 0,1,...,nC \}$
is a lower bound of the number of filled stacks.

(iii) $LBnTrucks := \lceil LBnStacks/(2*nStPl) \rceil$
is a lower bound of the number of needed trucks.

Proof: (i) Since two cap pallets cannot be packed into one stack at least *nst1*(*c*) pallets are needed for the customers *i* = *c*+1,...,*nC*. Packing a pallet of a customer $j \leq c$ and a cap pallet of a customer $i > c$ into one stack would violate one of the constraints (C7) or (C8). Hence at least *nst2*(*c*) additional stacks are necessary to load all pallets. The special cases *c* = 0 and *c* = *nC* are trivial. (ii) and (iii) are obvious. □

b) Static bound of maximum stack height

A lower bound of the maximum stack height is formulated for a given number of trucks in a complete solution. Let *maxhp* denote the maximum loading height of a pallet.

Proposition 2

$LBmaxStHeight := \max\{ maxhp, \lceil \sum_{c=1}^{nC} hp(c) /(2*nTrucks*nStPl) \rceil \}$
is a lower bound of the maximum stack height for all solutions with *nTrucks* trucks.

Proof: Obvious. □

c) Dynamic bound of number of trucks

Now let a partial solution with *nStacks* used stacks and *nPallets* (*nPallets* < *nP*) loaded pallets be given. Hence the pallets with indices *nPallets*+1,...,*nP* are free, i.e. still to be packed. The total pallet height of all free pallets is denoted as *hpres*. An existing (partially filled) stack is called usable if it can still accommodate at least one free pallet. Let *hstres* be the sum of the residual heights of all usable stacks.

Proposition 3

(i) $LBnAddStacks := \lceil (hpres\text{-}hstres)/hT) \rceil$ is a lower bound of the number of additional stacks needed to load the residual pallets.

(ii) $LBnTrucksD := \lceil (nStacks+ LBnAddStacks)/(2*nStPl) \rceil$ is a lower bound of the number of trucks needed to load all pallets.

Proof: (i) Note that the height difference *hpres* – *hstres* represents a lower bound of the total pallet height that can't be accommodated in existing stacks. (ii) Obvious. □

d) Dynamic bound of maximum stack height

Again a partial solution with *nStacks* used stacks and *nPallets* (*nPallets* < *nP*) loaded pallets is assumed. A lower bound of the maximum stack height is formulated for all *complete* solutions with *nTrucks* trucks. Let *maxhst* denote the maximum height of all used stacks. *nStacksRes* stands for the difference *nTrucks**nStPl* – *nStacks*, i.e. for the maximal number of additional stacks.

Proposition 4

$LBmaxStHeightD := \max\{\, maxhst,\ maxhp,\ \lceil (hpres - hstres)/nStacksRes \rceil \,\}$
is a lower bound of the maximum stack height for all solutions with *nTrucks* trucks.

Proof: Cf. proof of prop. 3. □

The static lower bounds are used each time a new complete solution was generated. In this case the search is aborted if the complete solution equals both of the lower bounds, i.e. if a globally optimal solution was reached. The dynamic lower bounds are used each time before the current solution is extended. The extension is saved if the best solution so far can't be improved any longer.

7 Computational test

The solution procedure for the considered two-stage packing problem, consisting of the procedures LBOP and LPIT, was implemented in C and tested by means of an Intel PC (Core 2 Duo 1.8 GHz). In this section first two sets of problem instances are introduced before the corresponding computational results are reported and commented.

7.1 Two sets of problem instances

The first set includes 12 instances of the (complete) two-stage packing problem and these instances are based on real data taken from the distribution process of company O&I. In Table 3 the main characteristics of the 12 instances are listed.

Table 3 – Characteristics of the 12 instances of set 1 (all dimensions in cm)

Characteristic	Data
Truck loading space	length: 800, width: 250, height: 260
Pallet type (empty pallet)	length: 120, width: 80, height: 15
Stack places per strip *nStPl*	10
Requested pallet support *minSupTopArea*	70%
No. of routes per instance	1 (constantly)
No. of customers per instance	3 – 5
No. of demanded box types per customer	1 – 4
Mean no. of boxes per instance	17,000

The maximum loading height of a pallet is stipulated individually per customer. This is illustrated in Figure 6 that shows some pallets which were loaded using different limits of the loading height.

The second set of test instances comprises 126 instances of the packing problem that occurs at the second packing stage. Table 4 shows the main characteristics of these instances that serve to subject the procedure LPIT a more thorough test.

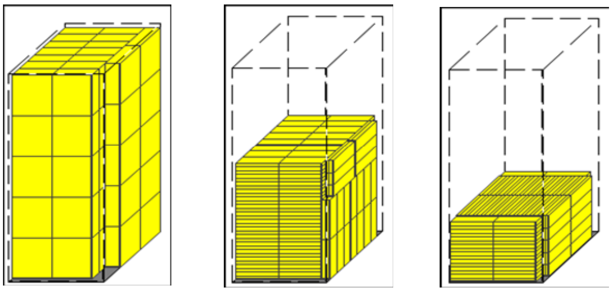

Figure 6 – Examples of loaded pallets

The instances of set 2 are generally considerably larger regarding the number of pallets than the corresponding LPIT-instances that result from the 12 instances of set 1 (see Table 5).

Table 4 – Characteristics of the 126 instances of set 2 (all dimensions in cm)

Characteristic	Data
Truck height (height of loading space)	230
Stack places per strip *nStPl*	10
Requested pallet support *minSupTopArea*	80%
No. of customers per instance	1 – 3
No. of pallets per instance	51 – 105
Pallet heights (as rate of truck height)	16% – 33%
Percentage of cap pallets (see Section 3)	2% – 30%

7.2 Computational results

The time limit for the procedure LBOP was set to 1 minute and one iteration is performed for each problem instance. Ten different values for α parameter are considered, ranging from 0 to 1 in a step of 0.1 and for each α we run the GRASP algorithm 20 times. The time limit for procedure LPIT was set to 3 seconds while the number of stacking variants was bounded by *maxSucc* = 5.

In Table 5 the main results for the 12 instances of set 1 are shown that were achieved in the first packing stage by procedure LBOP. The following data are offered per instance: number of customers, total number of loaded pallets, mean volume utilization over all loaded pallets (in %), number and rate (in %) of generated cap pallets.

Considering the critical success factors, namely the mean volume utilization and the number (or rate) of cap pallets, it seems that procedure LBOP yields sufficient results. In particular the numbers of cap pallets will not contribute to increasing stack numbers.

Table 6 shows the main results for the 12 instances of set 1 that were achieved in the second packing stage by procedure LPIT. For each instance the following data are indicated: number of customers, number of pallets to be packed, number of stacks used, number of loaded trucks (*nTrucks*), truck span, maximum stack height (*maxStackHeight*, in cm) and height gap. The truck span is based on the (static) lower bound of the truck number *LBnTrucks* (see Section 6) and is given as difference (*nTrucks* – *LbnTrucks*). The height gap is based on the (static) lower bound of the maximum stack height *LBmaxStHeight* (see Section 6) and is computed as ratio (*maxStackHeight* – *LBmaxStHeight*)/*LBmaxStHeight* (in %).

The core result is that in 6 of 12 cases the optimal truck number was reached. The height gaps are moderate: they do never exceed 32% and amount to 17.6% on average. It should be stressed that the results are compared only with lower bounds (that are more or less weak) while the optimal truck number is unknown for 6 of 12 instances and the optimal stack height is even unknown in all cases.

Table 5 – Overall results for the 12 instances of set 1, first packing stage

Instance	No. of customers	No. of loaded pallets	Mean volume utilization (%)	No. of cap pallets	Rate of cap pallets (%)
1	5	40	94	0	0.0
2	4	50	90	5	10.0
3	5	60	91	5	8.3
4	4	40	93	0	0.0
5	3	49	89	10	20.4
6	4	60	86	8	13.3
7	5	48	89	4	8.3
8	3	40	90	0	0.0
9	4	47	94	0	0.0
10	5	50	91	6	12.0
11	5	52	93	10	19.2
12	5	51	91	6	11.8
Avg.	**-**	**48.9**	**91**	**4.5**	**8.6**

A further result of the test of set 1 is that the computational effort per instance is rather low. The mean total computing time over the 12 instances amounts to ca. 65 seconds. The main part of the computing time falls upon the loading of pallets in the first stage while the computing time for the second stage never exceeds 5 seconds.

Table 6 – Overall results for the 12 instances of set 1, second packing stage

Instance	No. of customers	No. of pallets	No. of stacks	No. of trucks	Truck span	Max. stack height (cm)	Height gap (%)
1	5	40	20	1	0	256	0.4
2	4	50	37	2	1	170	31.8
3	5	60	20	1	0	253	2.9
4	4	40	20	1	0	256	0.4
5	3	49	39	2	1	164	27.1
6	4	60	20	1	0	249	1.6
7	5	48	39	2	1	165	27.9
8	3	40	20	1	0	256	0.4
9	4	47	37	2	1	166	28.7
10	5	50	38	2	1	168	30.2
11	5	52	39	2	0	170	31.8
12	5	51	40	2	1	165	27.9
Avg.	**-**	**48.9**	**30.7**	**1.5**	**0.5**	**-**	**17.6**

Regarding the second set of 126 instances, only overall results are reported here. Based on the (static) lower bound of the truck number *LBnTrucks* (see Section 6) it can be stated that for 115 of 126 instances the optimal truck number was reached. For the residual instances the lower bound was failed by one truck. The values of the height gap (as defined above) turn out to be moderate. Now the height gap never exceeds 31% and it amounts to 9.3% on average. The computational effort amounts to a few seconds per instance. All in all, the test by means of the representative second instance set proves that LPIT is an effective and efficient solution procedure for loading pallets into trucks with predefined stack places.

In Figure 7 pallet charts are presented for an arbitrarily chosen instance of the second packing stage. Each chart visualizes the loading of pallets within one strip (left or right) of one truck. Since only one truck was filled two strips are shown. Pallets of different customers are differently shaded. Besides a pallet index (left index) each pallet has a customer position index (right index) that indicates the position of the customer in the given (single) route of the instance. So it can be realized that the LIFO constraint is completely satisfied (see Section 3, (C8)).

Figure 7 – Pallet charts for one loaded truck

8 Conclusions

In this case study a two-stage packing problem is considered that occurs in the distribution process of a Portuguese trading company. Boxes including ordered goods are to be packed onto identical pallets which in turn are to be loaded into a minimal number of uniform trucks. Each truck provides a set of places for stacking pallets that are organized in two strips. Within each of the problem stages multiple packing constraints, as LIFO policy and support of boxes or pallets, are to be observed. A solution method was developed consisting of two procedures that cover the two problem stages. The procedure for packing boxes onto pallets was derived from the GRASP procedure for container loading from Moura and Oliveira (2005) while a new tree search procedure was devised for loading pallets into trucks. The (total) solution method was tested by means of problem instances whose data were taken from the distribution process of the company. The method proved to be able to reach near optimal results regarding the number of needed trucks in short running times of about one minute. One can expect that the implementation of the solution method in the distribution process would be advantageous for the company: on the one hand suitable stowage plans including visualization are provided quickly and automatically. On the other hand the dispatchers can make more effective packing decisions. In particular they are enabled to make proposals to customers very rapidly that help to establish a suitable balance of customer service and utilization of truck volume. All in all, the developed method could serve as a core module of an effective decision support tool.

Bibliography

Bischoff, E.E. (2004). Three dimensional packing of items with limited load bearing strength. *Eur. J. of Oper. Res.*, 168, 952–966.

Bortfeldt, A.; Gehring, H. (2001). A hybrid genetic algorithm for the container loading problem. *Eur. J. of Oper. Res.,* 131, 143–161.

Eley, M. (2002). Solving container loading problems by block arrangement. *Eur. J. of Oper. Res.* 141, 393–409.

Fanslau, T.; Bortfeldt, A. (2010). A Tree Search Algorithm for Solving the Container Loading Problem. *INFORMS J. on Computing,* 22, 222–235.

Mack, D.; Bortfeldt, A.; Gehring, H. (2004). A parallel hybrid local search algorithm for the container loading problem. *Internat. Trans. in Oper. Res.* 11, 511–533.

Martello, S.; Toth, P. (1990). Knapsack problems: Algorithms and Computer Implementations. John Wiley & Sons, ISBN: 0 471 92420 2, chapter 8.

Moura, A.; Oliveira, J.F. (2005). A GRASP approach to the Container-Loading Problem. *IEEE Intelligent Systems* 20, 50–57.

Parreño, F.; Alvarez-Valdes, R.; Oliveira, J.F.; Tamarit, J.M. (2007). A maximal-space algorithm for the container loading problem. *INFORMS J. on Computing*, 20, 412–422.

Pisinger, D. (2002). Heuristics for the container loading problem. *Eur. J. of Oper. Res.* 141, 382–392.