Abstract Code: 025-0545

Abstract Title: The problem of consolidation in the groupage process: formulation and solution approaches.

Authors:

`Tommaso Paletta`

1st Year Doctoral Research Candidates of the Kent Business School (KBS), University of Kent at Canterbury, United Kingdom. E-mail: tp245@kent.ac.uk

`Dr. Abey Kuruvilla`

School of Business and Technology, University of Wisconsin-Parkside, Kenosha, Wisconsin, 53144. E-mail: abey.kuruvilla@uwp.udu

`Dr. Francesca Guerriero`

Department of Electronics, Computer Science and Systems, University of Calabria, Arcavacata di Rende (CS), Calabria, Italy. E-mail: guerriero@deis.unical.it

`Dr. Luigi Di Puglia Pugliese`

Department of Electronics, Computer Science and Systems, University of Calabria, Arcavacata di Rende (CS), Calabria, Italy. E-mail: ldipuglia@deis.unical.it

POMS 23rd Annual Conference

Chicago, Illinois, U.S.A.

April 20 to April 23, 2011

Abstract

This paper addresses the problem of consolidation in the groupage process. Groupage is a particular kind of shipment where items from different suppliers, located in the same area and addressed to different clients in another area, are grouped (consolidated) in the same cargoes. This kind of shipment allows a reduction of cost to clients because they can order a smaller quantity of items, resulting in a reduction of inventory cost as well as transportation costs owing to the costs being shared among customers. In this paper, a new mathematical formulation of the problem of consolidation is considered: it is based on existing formulations in the literature (Knapsack and 3D Bin Packing problems) but it contains new considerations. It considers fragile items that cannot be under other items, the balancing of the weight in the bin (container) and the possibility of declaring a couple of items incompatible, that is to say they have to be inserted in two different bins or at least at some distance from each other. Furthermore, some lower bounds and two new heuristic algorithms to solve the problem are proposed and the heuristic solutions are compared with exact solutions obtained using the optimization software ILOG CPLEX. The results of a wide computational experimentation demonstrates that the proposed algorithms perform well.

Keywords. bin-packing, consolidation, formulation, balancing constraint, fragile items, knapsack, heuristics.

1 Introduction

The objective that we want to achieve using groupage is to reach a situation of Full Truck Load (FTL), that means cargoes that are completely full during transportation. The difference between the traditional FTL and the groupage is that in the latter there are no repositories but there is a node, absent in the former, where items are consolidated. In this paper, the groupage process will be tackled from the point of view of optimization of consolidation of items. These classes are NP-hard because they are a generalization of the 1D-bin packing problem. Following the description in [1] this is a 3d-bin packing problem ortogonal. Literature tackles this problem from different points of view and with different hypotheses:

- consolidation with considerations about scheduling: the main problem is about minimizing the makespan that is to say the time to complete the loading and unloading tasks (see [2] for further details).
- consolidation with considerations about Vehicle Routing: a set of vehicles is used to transport the items from vendors to customers in a cross-docking system. Each vehicle can pick up items from different vendors and deliver them to different customers in the same route. The picked up items are delivered first to a cross-docking center where they are first consolidated and then transported. (see [3] for further details).

- consolidation as a particular kind of bin packing: most papers in this area tackle the bin packing problem, knapsack problem and container loading problem (brief descriptions of the above problem is in [4]). The article [5] was the first paper that studied 3d-container loading. The authors developed a heuristic approach to load different-dimension items in a bin. This approach inserts items in the bin using the logic of layers.

An example of the use of groupage in real life is The Home Depot ([6]). "An American retailer of home improvement and construction products and services"[1], it has more than 2200 stores throughout the United States, Canada, China and Mexico. Its goal is "to turn The Home Depot's supply chain into a competitive advantage in the marketplace" and it is trying to have Full Truckload Shipments with good results.

2 The consolidation problem: formulation

In this section the formulation of the consolidation problem is discussed. The formulation is based on the bin packing, knapsack and container loading formulations and some considerations about groupage are added. There is a subsection about the hypotheses, one about the parameters, that is to say all the elements that are known when you try to solve the problem, one about

[1]See [7]

the variables and then one about the real formulation.

2.1 Hypotheses

The formulation described in the subsection 2.4 is based on the following hypothesis:

- all the items are available when we are solving the problem. This means that the locations of items is not a function of the time when the items reach the cross-docking center. In literature this hypothesis is called off-line problem.

- the cost to locate an item in a container is independent of the location (such as near the door of the container or at the rear of the container. The only cost we consider is the cost of using the container.

- it is possible to rotate the items but only around the height. This hyphothesis is real because usually items are on pallet so the orientation is given by the suppliers. (This hypothesis is included in [8] too)

- each container is cleared out when it reaches the cross dock center at destination so the locations of items is independent to the order in which *consignee* will be served.

  Another situation is when the container or carrier reaches the customers directly without going to a cross dock center at destination. It this case

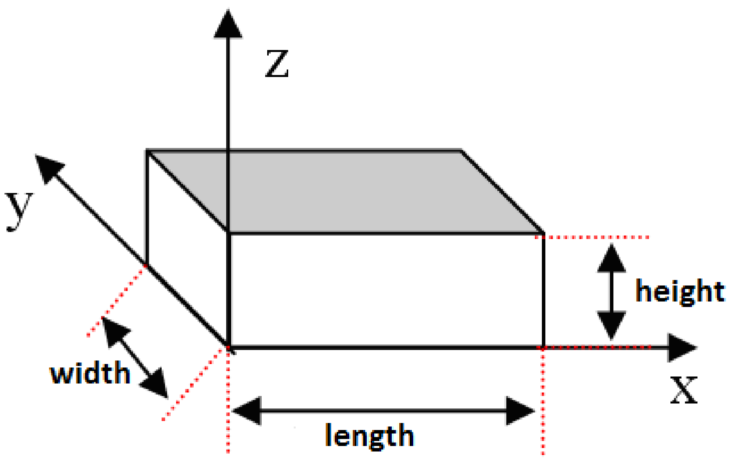


Figure 1: Definition of the origin of the coordinate system in a bin

we need to locate items in the bin in such a way that it is easy to unload each item when it is needed. That is to say items that will be unloaded first should be loaded last. (LIFO) . (*Vehicle Routing on cross-dock*).

- the origin of the coordinate system is in the vertex below and left of the bin so x axis coincides with the length of the bin, y axis with the width and z axis with the height. To better understand see figure **1**.

- length, width and height of each item must be parallel to one of the three dimensions of the bin that contains it. (the problem is **orthogonal**).

- the location of each item is described by the triple (x, y, z) that indicates the location of the vertex of the item closer to the origin of the coordinate system. To better understand see figure **2**.

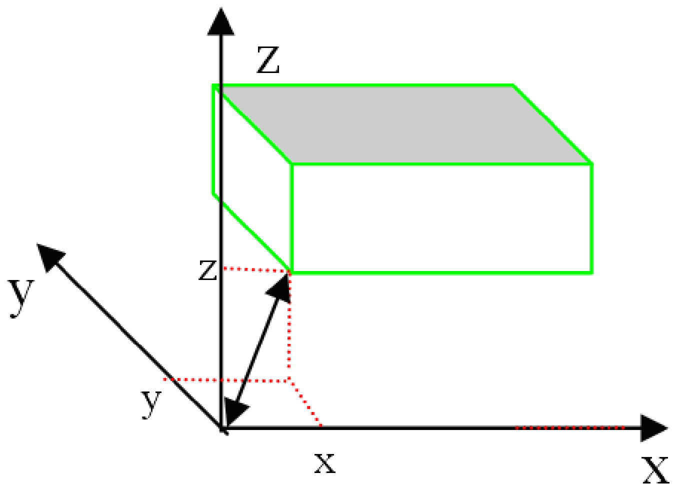
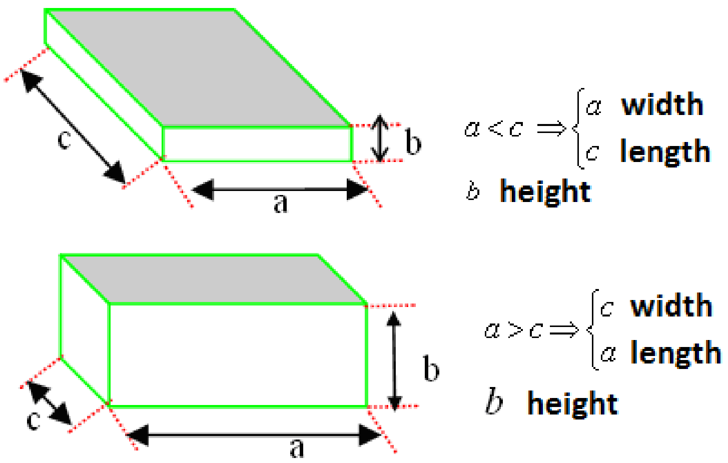


Figure 2: Definition of the location of a item in a bin

2.2 Parameters

- the set $Item = \{1, 2, \ldots, \mathbf{N}\}$ is the set of items that we must transport.
- the set $Bin = \{1, 2, \ldots, \mathbf{m}\}$ is the set of available bins. Each kind of bin can be used more than one time so **m** is considered equal to $N \times b$ where b the number of typologies of bins available.
- for each **item** we consider:
  - its length p, its width q and its height r. For the generic item i we have: p_i, q_i, r_i. For an item we know the height because it is not possible to knock over the items but we must define width and length: we define *length* the biggest dimension between the



c
b
a
$a < c \Rightarrow \begin{cases} a & \text{width} \\ c & \text{length} \end{cases}$
b height
c
b
a
$a > c \Rightarrow \begin{cases} c & \text{width} \\ a & \text{length} \end{cases}$
b height

Figure 3: Samples to understand which dimension is the length

two remaining (height excluded) and *width* the other. To better understand see figure **3**

- the weight. For the generic item i we have $weight_i$.
- a numerical value that indicates whether the item can tollerate a weight above. For he generic item i we have:

$$NoSopra_i = \begin{cases} 0 & \text{if above the item i no items can be placed} \\ 1 & \text{otherwise.} \end{cases}$$

A item with $NoAbove = 0$ will be called easily **fragile**.

- for each **Container** we have:
  - the length L, the width W and the height H. For the generic bin j we have: L_j, W_j, H_j.

  - the cost. For the generic bin j we have c_j.
  - the capacity, that is to say the greatest weight that it is possible to insert in the bin. For the generic bin j we have $Capacity_j$.

- for each couple of **Items** we have:
  - a numerical value that indicates if the two items are compatible or not. Two items are *incompatible* if they cannot be located in the same container or even if they can be located in the same container there must be away from each other. For the generic couple of items (α, β) we have:

$$comp_{\alpha,\beta} = \begin{cases} d_{\alpha,\beta} & \text{if items } \alpha \text{ and } \beta \text{ must be distant at least at } d_{\alpha,\beta} \\ M & \text{item } \alpha \text{ is incompatible with item } \beta \end{cases}$$

    If item α is compatible with item β then $d_{\alpha,\beta} = 0$.

  - a numerical value that indicates that the two items of the couple must be located in the same container. For the generic couple (α, β) we have:

$$insieme_{\alpha,\beta} = \begin{cases} 1 & \text{if items } \alpha \text{ and } \beta \text{ must be located in the same bin} \\ 0 & \text{otherwise.} \end{cases}$$

- M is a big number. It is possible to set it to:

$$M = \max_{j \in \text{Bin}} \{\max\{L_j, W_j, H_j\}\}.$$

- Mom_{max} indicates the greatest torque that could be present at a bin because of the disposition of the items.

- quantity, indicates the greatest weight that an item can be located not on the ground. That is to say if the weight of the item is over that quantity it must be located on the ground.

2.3 Variables

- for the generic **Item** i we have:

  - $lx_i = \begin{cases} 1 & \text{if the length of the item i is parallel to the x axis} \\ 0 & \text{otherwise.} \end{cases}$

    $ly_i = \begin{cases} 1 & \text{if the length of the item i is parallel to the y axis} \\ 0 & \text{otherwise.} \end{cases}$

  - $wx_i = \begin{cases} 1 & \text{if the width of the item i is parallel to the x axis} \\ 0 & \text{otherwise.} \end{cases}$

    $wy_i = \begin{cases} 1 & \text{if the width of the item i is parallel to the y axis} \\ 0 & \text{otherwise.} \end{cases}$

  - the triple (x_i, y_i, z_i) indicates the location of item i in the bin in which it is located. It is the point indicated in the section **2.1** and in figure **2**.

To better understand the meaning of the variables $l_{x,i}, l_{y,i}, w_{x,i}, w_{y,i}$ see figure **4**.

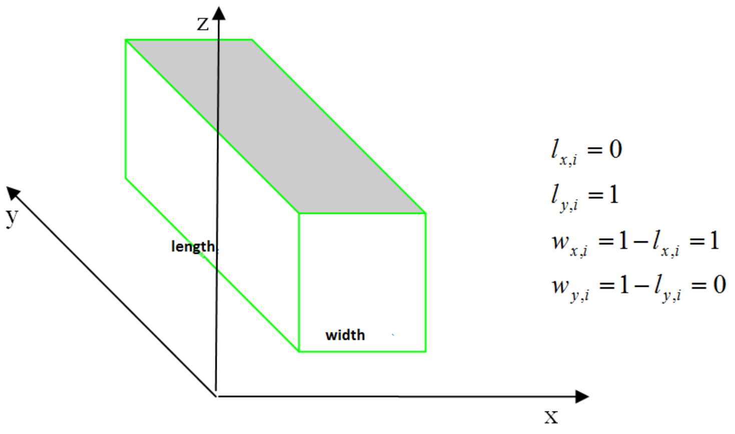


Figure 4: Definition of variables $l_{x,i}, l_{y,i}, w_{x,i}, w_{y,i}$ (sample)

- for each couple of **Item** (for the generic couple of items (α, β) with $(\alpha \leq \beta)$) we have:
  - to indicate the reciprocal locations of the items on the base of the x axis:
    - $Left_{\alpha,\beta} = \begin{cases} 1 & \text{if item } \alpha \text{ is on the } LEFT \text{ of the item } \beta \\ 0 & \text{otherwise.} \end{cases}$
    - $Right_{\alpha,\beta} = \begin{cases} 1 & \text{if item } \alpha \text{ is on the } RIGHT \text{ of the item } \beta \\ 0 & \text{otherwise.} \end{cases}$
  - to indicate the reciprocal locations of the items on the base of the y axis:
    - $Ahead_{\alpha,\beta} = \begin{cases} 1 & \text{if item } \alpha \text{ is on the } AHEAD \text{ of the item } \beta \\ 0 & \text{otherwise.} \end{cases}$

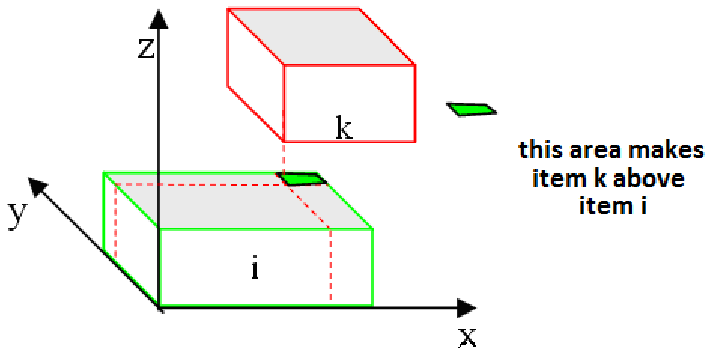


Figure 5: Definition variables Above,Below, Ahead, Behind, Left, Right (Sample 1)

* $Behind_{\alpha,\beta} = \begin{cases} 1 & \text{if item } \alpha \text{ is } BEHIND \text{ item } \beta \\ 0 & \text{otherwise.} \end{cases}$

- to indicate the reciprocal locations of the items on the base of the z axis:

  * $Above_{\alpha,\beta} = \begin{cases} 1 & \text{if item } \alpha \text{ is } ABOVE \text{ the item } \beta \\ 0 & \text{otherwise.} \end{cases}$

  * $Below_{\alpha,\beta} = \begin{cases} 1 & \text{if item } \alpha \text{ is } BELOW \text{ the item } \beta \\ 0 & \text{otherwise.} \end{cases}$

- to indicate if two items are in the same bin:

  * $same_{\alpha,\beta} = \begin{cases} 1 & \text{if item } \alpha \text{ and item } \beta \text{ are in the same bin} \\ 0, & \text{otherwise.} \end{cases}$

• for each **Carrier** (for the generic carrier j) we have:

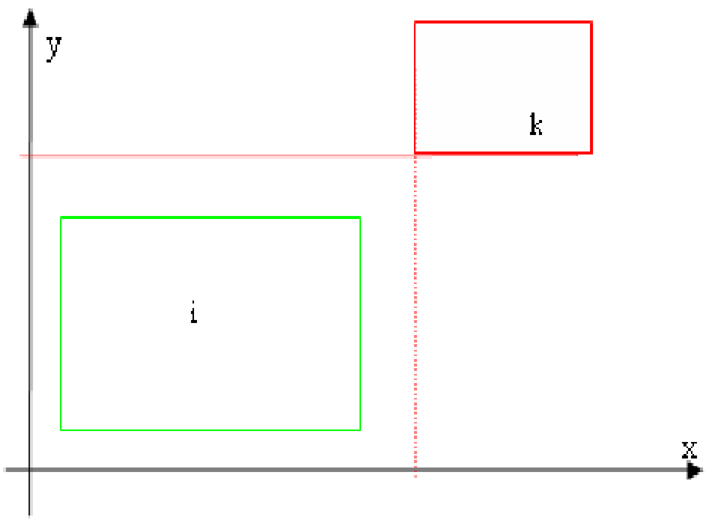


Figure 6: Definition variables Above,Below, Ahead, Behind, Left, Right (Sample 2)

– $n_j = \begin{cases} 1 & \text{if bin j is used} \\ 0 & \text{otherwise.} \end{cases}$

- for each **Carrier** and each **Item** (for the generic carrier j and the generic item i) we have:

  – $s_{i,j} = \begin{cases} 1 & \text{if item } i \text{ is located in the bin } j \\ 0 & \text{otherwise.} \end{cases}$

  – $SIlx_{i,j} = \begin{cases} 1 & \text{if item i is in the bin j and if its } \textit{length} \\ & \text{is parallel to x axis} \\ 0 & \text{otherwise.} \end{cases}$

- $$SIwx_{i,j} = \begin{cases} 1 & \text{if item i is in the bin j and if its } \textit{width} \\ & \text{is parallel to x axis} \\ 0 & \text{otherwise.} \end{cases}$$

- x_i^j indicates the location of the item i in the bin j only regarding the x axis. This variable is used to calculate the torque (see constraint **17**).

  $$x_i^j = \begin{cases} x_i, & \text{if item i is in the bin j } (s_{i,j} = 1) \\ 0, & \text{otherwise.} \end{cases}$$

2.4 Mathematical Formulation

$$Min \sum_{j=1}^{m} c_j \cdot n_j \tag{1}$$

$$\sum_{i=1}^{N} weight_i \cdot s_{i,j} \leq Capacity_j \cdot n_j \quad \forall j \tag{2}$$

$$\sum_{j=1}^{m} s_{i,j} = 1 \quad \forall i \tag{3}$$

$$x_i + p_i \cdot l_{x,i} + q_i \cdot w_{x,i} + comp_{i,k} \leq x_k + (1 - Left_{i,k}) \cdot 2M, \forall i,k|i < k \tag{4}$$

$$x_k + p_k \cdot l_{x,k} + q_k \cdot (1 - l_{x,k}) + comp_{i,k} \leq x_i + (1 - Right_{i,k}) \cdot 2M, \forall i,k|i < k \tag{5}$$

$$y_i + p_i \cdot l_{y,i} + q_i \cdot w_{y,i} + comp_{i,k} \leq y_k + (1 - Behind_{i,k}) \cdot 2M, \forall i,k|i < k \tag{6}$$

$$y_k + p_k \cdot (1 - l_{x,k}) + q_k \cdot l_{x,k} + comp_{i,k} \leq y_i + (1 - Ahead_{i,k}) \cdot 2M, \forall i,k|i < k \tag{7}$$

$$z_i + r_i + comp_{i,k} \leq z_k + (1 - Below_{i,k}) \cdot 2M, \forall i, k | i < k \tag{8}$$

$$z_k + r_k + comp_{i,k} \leq z_i + (1 - Above_{i,k}) \cdot 2M, \forall i, k | i < k \tag{9}$$

$$Above_{i,k} + Below_{i,k} + Right_{i,k} + Left_{i,k} + Ahead_{i,k} + Behind_{i,k} \geq s_{i,j} + s_{k,j} - 1, \quad \forall i, k, j | i < k \tag{10}$$

$$x_i + p_i \cdot l_{x,i} + q_i \cdot w_{x,i} \leq L_j + (1 - s_{i,j}) \cdot M, \quad \forall i, j \tag{11}$$

$$y_i + p_i \cdot l_{y,i} + q_i \cdot w_{y,i} \leq W_j + (1 - s_{i,j}) \cdot M, \quad \forall i, j \tag{12}$$

$$z_i + r_i \leq H_j + (1 - s_{i,j}) \cdot M, \quad \forall i, j \tag{13}$$

$$x_i^j \leq x_i + (1 - s_{i,j}) \cdot M, \quad \forall i, j \tag{14}$$

$$x_i^j \geq x_i - (1 - s_{i,j}) \cdot M, \quad \forall i, j \tag{15}$$

$$x_i^j \leq s_{i,j} \cdot M, \quad \forall i, j \tag{16}$$

$$\begin{aligned} -Mom_{max} \leq \sum_{i=1}^{N} weight_i \cdot \Bigg[\left(\frac{L_j}{2} - x_i^j - \frac{p_i}{2}(1 - SIwx_{i,j}) - \frac{q_i}{2}(1 - SIlx_{i,j}) \right) + \\ - \left(\frac{L_j}{2} - \frac{p_i}{2} - \frac{q_i}{2} \right)(1 - s_{i,j}) \Bigg] \leq Mom_{max} \quad \forall j \end{aligned} \tag{17}$$

$$SIlx_{i,j} \leq l_{x,i} \quad \forall i, j \tag{18}$$

$$SIlx_{i,j} \leq s_{i,j} \quad \forall i, j \tag{19}$$

$$SIlx_{i,j} \geq l_{x,i} + s_{i,j} - 1 \quad \forall i, j \tag{20}$$

$$SIwx_{i,j} \leq w_{x,i} \quad \forall i, j \tag{21}$$

$$SIwx_{i,j} \leq s_{i,j} \quad \forall i,j \tag{22}$$

$$SIwx_{i,j} \geq w_{x,i} + s_{i,j} - 1 \quad \forall i,j \tag{23}$$

$$same_{i,k} \leq s_{i,j} - s_{k,j} + 1 \quad \forall i,k,j,i<k \tag{24}$$

$$same_{i,k} \leq s_{k,j} - s_{i,j} + 1 \quad \forall i,k,j,i<k \tag{25}$$

$$same_{i,k} \geq s_{i,j} + s_{k,j} - 1 \quad \forall i,k,j|i<k \tag{26}$$

$$Above_{k,i} \leq NoAbove_i + Ahead_{k,i} + Behind_{k,i} + Left_{k,i} + Right_{k,i} + (1 - same_{k,i}) \quad \forall i,k|k<i \tag{27}$$

$$Below_{i,k} \leq NoAbove_i + Ahead_{i,k} + Behind_{i,k} + Left_{i,k} + Right_{i,k} + (1 - same_{i,k}) \quad \forall i,k|i<k \tag{28}$$

$$same_{i,k} \geq Together_{i,k} \quad \forall i,k|i<k \tag{29}$$

$$z_i = 0 \quad \forall i|weight_i \geq quantity \tag{30}$$

$$z_i \leq M \cdot \left(\sum_{k \in Item|k>i} Above_{i,k} + \sum_{k \in Item|k<i} Below_{k,i} \right) \quad \forall i \tag{31}$$

$$l_{x,i} + l_{y,i} = 1 \quad \forall i \tag{32}$$

$$w_{x,i} + w_{y,i} = 1 \quad \forall i \tag{33}$$

$$l_{x,i} + w_{x,i} = 1 \quad \forall i \tag{34}$$

$$l_{y,i} + w_{y,i} = 1 \quad \forall i \tag{35}$$

$$l_{x,i}, l_{y,i} = \{0,1\} \quad \forall i \tag{36}$$

$$w_{x,i}, w_{y,i} = \{0,1\} \quad \forall i \tag{37}$$

$$Above_{i,k}, Right_{i,k}, Ahead_{i,k} \quad = \quad \{0,1\} \quad \forall i,k,k \neq i \tag{38}$$

$$Below_{i,k}, Left_{i,k}, Behind_{i,k} \quad = \quad \{0,1\} \quad \forall i,k,k \neq i \tag{39}$$

$$s_{i,j} = \{0,1\} \quad \forall i,j \tag{40}$$

$$n_j = \{0,1\} \quad \forall j \tag{41}$$

$$x_i^j \geq 0 \quad \forall i,j \tag{42}$$

$$x_i, y_i, z_i \geq 0 \quad \forall i \tag{43}$$

$$same_{i,k} = \{0,1\} \quad \forall i,k,k \neq i \tag{44}$$

$$SIwx_{i,j}, SIlx_{i,j} = \{0,1\} \quad \forall i,j \tag{45}$$

The objective function (1) of the problem is to minimize the cost of the use of carriers. Other objects are possible, for example to reduce wasted space. **2** indicate that the total weight of the items located in a bin must be less than its capacity. The constraints avoid that an item is located in a

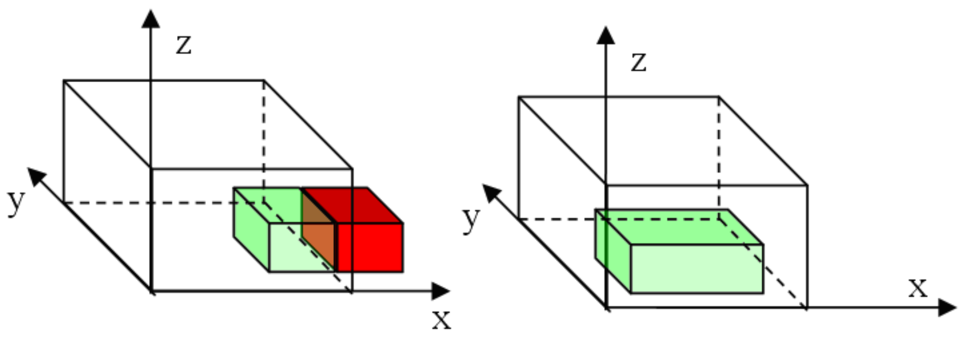


Figure 7: Correct and incorrect position of items in a bin

bin if the bin is not part of the used bins. **3** indicate that each item must be located in one and only one bin. Constraints from **4** to **10** are used to avoid that two items located in the same bin are superimposed, that is to say they guarantee that the intersection in the space of each couple of items is empty. These constraints exist in literature but they do not consider the compatibility between each couple of items and consider instead the possibility of knocking items over (see [9]). **11**, **12** and **13** localize items completely in a bin. They avoid that an item is localized in such a position that part of the item is out of the border of the bin. To better understand see figure **7**. The constraints from **14** to **16** define x_i^j. **17** indicate that in each carrier the torque generated by the weight of items around the central axis of the carrier (the axis that divides the carrier in two equal parts on the x axis) must be from a minimum value to a maximum value. The torque is

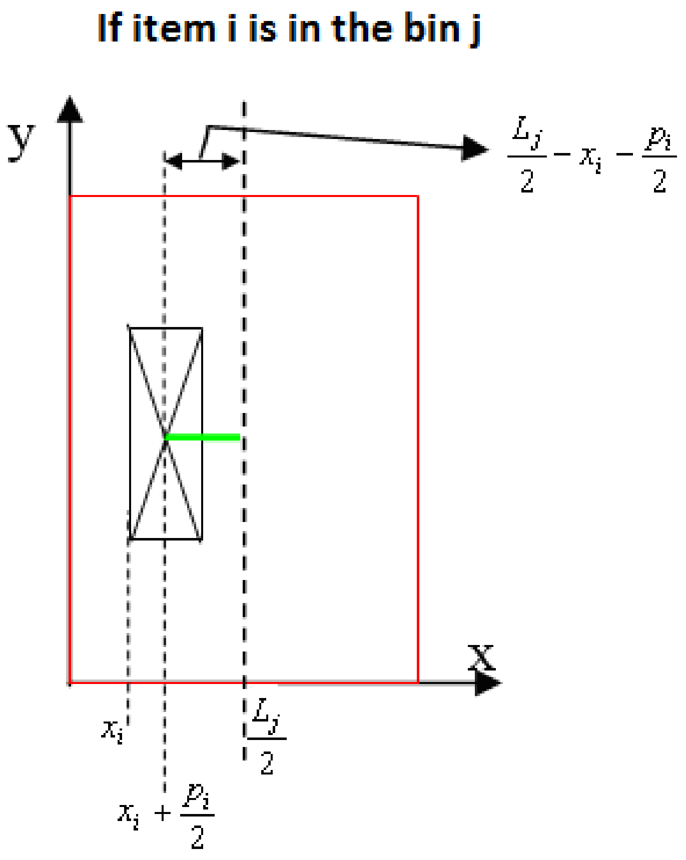

If item i is in the bin j
y
$\frac{L_j}{2} - x_i - \frac{p_i}{2}$
x
x_i
$\frac{L_j}{2}$
$x_i + \frac{p_i}{2}$

Figure 8: Definition of lever arm for the calculation of torque in the bin

calculated as

$$\sum_{\text{item in the bin}} [\text{lever arm} \cdot \text{item's weight}]$$

where the lever arm is the distance between the center of gravity of the item and the axis. See figure **8** for further details.

24, **25** and **26** define the variables $stesso_{i,k}$. **27** and **28** avoid that items are located above fragile items. **29** impose that two items i and k, with $Insieme_{i,k} = 1$ must be located in the same bin. **30** locate heavy items, the items with a weight higher than the greatest weight allowed (*quantity*), on the ground of the bin. **31** locate the generic item i on the ground if there are no other items under it. It is possible to not considerate these constraints and after the solution of the problem each item should be translated in the direction of the ground. The constraints from **32** to **35** connect lx_i, ly_i, wx_i, wy_i to each other. The constraints from **36** to **45** define the domains of the variables.

In the above formulation we use $7 \cdot N + 4 \cdot N \cdot m + \frac{7 \cdot N \cdot (N-1)}{2} + m$ variables and among these $4 \cdot N + 3 \cdot N \cdot m + \frac{7 \cdot N \cdot (N-1)}{2} + m$ are integers. We have $2 \cdot m + 6 \cdot N + \frac{N \cdot (N-1)}{2} \cdot (9 + 4 \cdot m) + 12 \cdot N \cdot m$ constraints.

3 Lower Bounds

Considering the simplification that the containers are all the same, the lower bounds confederated refer to:

- **Available volume in a bin**: $LB1 = \left\lceil \frac{\sum_{i=1}^{N} p_i \cdot q_i \cdot r_i}{L \cdot W \cdot H} \right\rceil$.

  See figure **9** where it is possible to see that items in this lower bound are considered only as a volume and not as a geometrical figure.

  There are some improvements in this lower bound: for example if an item is so high that no item can go over it, its height will be considered equal to the high of the bin. The same is for the width and the length.

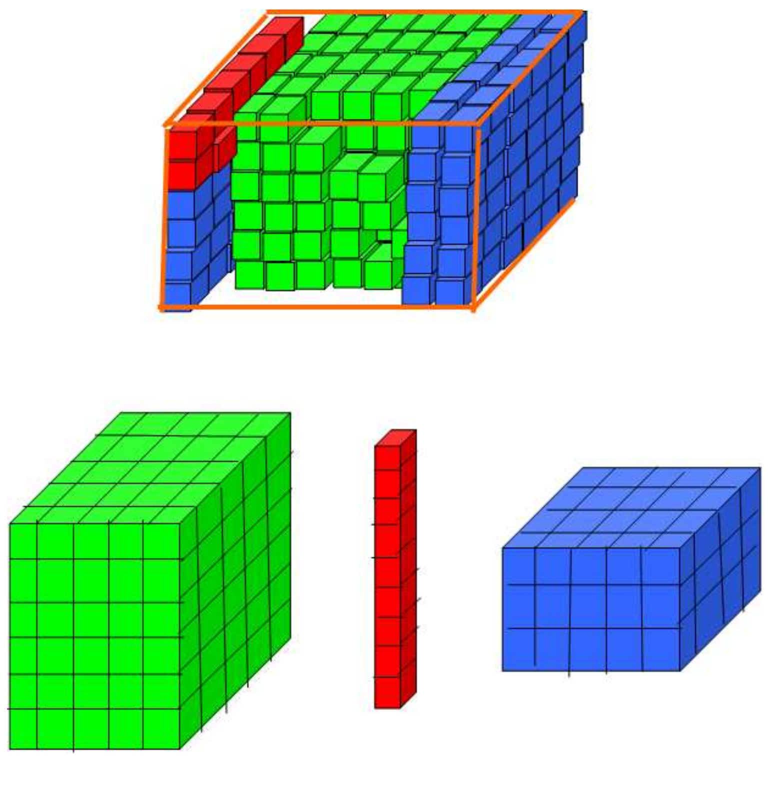


Figure 9: Lower Bound Volume

- **The capacity of a bin**: $LB2 = \left\lceil \frac{\sum_{i=1}^{N} peso_i}{\text{Capacity}} \right\rceil$.

- **The greatest available area for the NoAbove items**: $LB3 = \left\lceil \frac{\sum_{i=1}^{N} p_i \cdot q_i \cdot (1 - NoSopra_i)}{L \cdot W} \right\rceil$. This lower bound uses the characteristic of some items that are fragile and cannot support items over them. In each container the greatest area of a fragile item is equal to the area of the base of the bin $(L \cdot W)$.

- **The greatest available area for items**: $LB4 = \left\lceil \frac{\sum_{i=1}^{N} p_i \cdot q_i}{L \cdot W \cdot \left\lfloor \frac{H}{hmin} \right\rfloor} \right\rceil$.
  There is a partition of the bin in layers: we consider the item with the smallest height and that height will be the height of each layer. We know the number of layers in each container and as the area of each layer is $L \cdot W$, in each container we know the area available.
  See figure **10** to better understand: in the figure **f.** we have the items that we must insert in the bins with the smallest height (in red). In the figure **g.** a bin is partitioned in layers, each of them with a height equal to the red height. Figures **a.**, **b.** and **c.** are the plans of the three layers of the bin. Figures **d.** and **e.** are respectively the area of violet items and azure items. In this sample the violet item cannot be inserted in the bin because there is no space in the three layers so we need another layer and so another bin. The lower bound is 2.

The two first lower bounds are present in the literature but the second two are new.

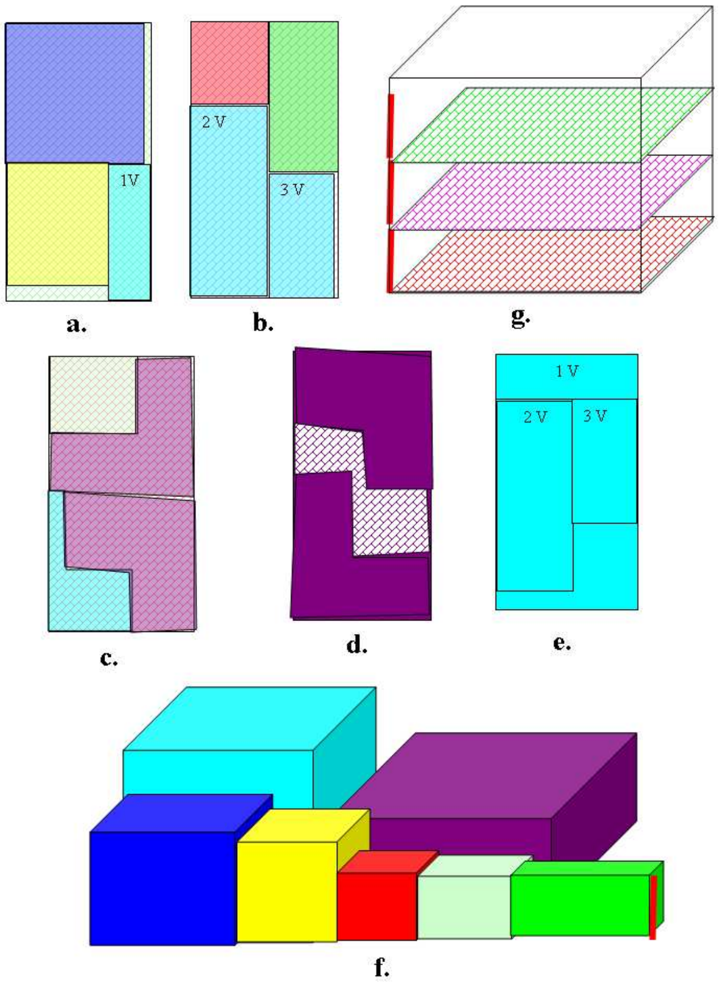


Figure 10: Lower Bound Area Disponibile

4 Algorithms

This section describes two new algorithms to solve the problem described above. The bodies of the algorithms are the same but they differ only in the way they verify the admissibility of each sub-problem where we define sub-problem the problem equal to verify if, given a bin that contains several items, it is possible to add another item in it. The first method verifies the admissibility by solving it in a exact way the problem using ILOG Cplex, the second verifies it by heuristic methods.

4.1 Body

Using the best among the lower bounds described above, we have the lowest number of bins that we have to use in order to deliver all the items. Executing the best of these lower bound algorithms you should put the items in the opened bins so at this point some bins contain some items then the algorithm starts dividing the remaining (outside the bins) items in different categories:

- Items that have to be in the same bin (Together)
- Items that cannot stay in the same bin of at least another item (Alone)

Then the Alone list should be ordered according to a criterion such as:

- increasing (or decreasing) height or length or width or weight
- increasing or decreasing maximum dimension (between length and width

- before the fragile items and then the others or vice versa
- increasing or decreasing identification numbers
- increasing or decreasing of the number of incompatibility (the number of items that cannot stay in the same bin of the considered item)

Now consider a bin.

Take a group of elements that should go together and verify if the problem is admissible[2]. If the problem is admissible, insert all the elements of the group in the bin and update the list of elements that cannot be in this bin. Repeat this action until the group of items together, ends. Then take an item that is alone and try to insert it in the bin[3]. If the problem is admissible, insert all the elements of the group in the bin and update the list of elements that cannot be in this bin. Do this for all the items that are alone. Then consider another bin and start again with the group of items that are in a group and alone..

If there are other items that are together or alone, repeat until the end of all the item in the two list these operations: open a new bin and try put these

[2]Start verifying if the four lower bounds described above are equal to 1. If there is one lower bound that is more than 1 the problem is not admissible. If all the lower bound is equal to 1 then solve it using ILOG CPLEX or using the heuristic procedure described in the next section.

[3]see footnote 2

item in it[4] and if it is possible insert it.

4.2 Solution of a sub-problem

There are two kinds of solutions to the sub-problem. The first is using Ilog Cplex: the formulation of the sub-problem is the same as the main problem where only one bin is considered and we are not interested in the value of the objective but only if it is admissible or not. The second method is a heuristic one and is described in the following. The algorithm uses different kinds of strategies and tries all of them until one of them finds a solution. Then they try to fix all the variables in order to satisfy the constraints.

5 Computational results

We discuss the results in this section.. Three methods were used to solve the problem:

- exact solution using ILOG CPLEX on the formulation described in section 2.4. (In the tables that problem is named as Exact)
- the hybrid solution using the heuristic methods but ILOG CPLEX to solve the sub-problems (In the tables that problem is named as Heur-Exact)

[4]see footnote 2

- the heuristic solution that uses the heuristic method to solve the sub-problems (In the tables that problem is named as Heur-Heur)

The algorithms are implemented by Java ver. 6 and a Intel(R) Core(TM)2 Duo CPU T5750 2.00 GHz, RAM 3,00 GB is used to solve it. The samples are divided in categories:

Class A items have low weight in accordance with the capacity of the bins. This class does not consider fragile Items.

Class B items have high weight in accordance to the capacity of the bins. This class does not consider the fragile Items

Class C the same as class B but it considers fragile items as well.

Class D in these instances the range in which the torque can vary is more binding

Class E the same as class C but the items have dimensions that vary in the range of $[1 \div \frac{L}{5}]$

Class F the same as class C but the items have dimensions that vary in the range of $[1 \div \frac{2 \cdot L}{5}]$

Class G the same as class C but the items have dimensions that vary in the range of $[1 \div \frac{3 \cdot L}{5}]$

Class H the same as class C but the items have dimensions that vary in the range of $[1 \div \frac{4 \cdot L}{5}]$ We considered the same samples two times: the first without incompatibility constraints and the second with a lot of incompatibility constraints.

The algorithms are implemented by Java ver. 6 and a Intel(R) Core(TM)2 Duo CPU T5750 2.00 GHz, RAM 3,00 GB is used to solve it. The samples are divided in categories:

- S=number of items in the solutions
- C=times to load the problems on cplex
- T=solution time
- TL=times to obtain the lower bound

	Number of items in the sample							
Class	10	20	30	40	50	100	150	200
A	0	8	16	24	32	40	48	56
B	1	9	17	25	33	41	49	57
C	2	10	18	26	34	42	50	58
D	3	11	19	27	35	43	51	59
E	4	12	20	28	36	44	52	60
F	5	13	21	29	37	45	53	61
G	6	14	22	30	38	46	54	62
H	7	15	23	31	39	47	55	63

Table 1: Identification number of the samples

Samples with a lot of incompatibility constraints (1)										
	Lower	Exact			Heur Heur			Heur Exact		
Sample	Algorithm	S	C	T	S	TL	T	S	TL	T
0 M	2+0=2	2	0.202	0.453	2	0.001	0.003	2	0.042	0.021
1 M	2+0=2	2	0.063	0.359	2	0	0.01	2	0.027	0.023
2 M	2+0=2	2	0.032	0.686	2	0.003	0.012	2	0.027	0.029
3 M	2+0=2	2	0.031	0.421	2	0.005	0.003	2	0.031	0.021
4 M	2+0=2	2	0.048	0.624	2	0.005	0.003	2	0.025	0.028
5 M	2+0=2	2	0.031	0.483	2	0.007	0.002	2	0.037	0.04
6 M	2+0=2	2	0.063	0.577	2	0.01	0.004	2	0.056	0.03
7 M	2+0=2	2	0.031	0.499	2	0.01	0.002	2	0.032	0.025
8 M	3+0=3	3	0.11	85.232	3	0.006	0.153	3	0.065	0.083
9 M	3+0=3	3	0.125	82.945	3	0.006	0.134	3	0.091	0.088
10 M	3+0=3	3	0.14	57.299	3	0.006	0.132	3	0.066	0.121
11 M	3+0=3	3	0.126	91.735	3	0.006	0.152	3	0.093	0.123
12 M	3+0=3	3	0.125	95.245	3	0.006	0.022	3	0.138	0.11
13 M	3+0=3	3	0.125	70.59	3	0.003	0.025	3	0.123	0.138
14 M	3+0=3	3	0.109	69.383	3	0.002	0.069	3	0.127	0.15
15 M	3+0=3	3	0.126	80.714	3	0.008	0.034	3	0.103	0.151
16 M	3+0=3	3	0.234	156.072	3	0.004	0.838	3	0.122	1.02
17 M	3+0=3	3	0.266	24.055	3	0.001	0.523	4	0.168	0.309
18 M	3+0=3	3	0.265	485.248	4	0.002	0.446	4	0.203	0.327
19 M	3+0=3	3	0.234	383.967	4	0.004	0.463	4	0.153	0.438
20 M	3+0=3	3	0.25	20.094	3	0.002	0.048	3	0.184	0.384
21 M	3+0=3	3	0.317	46.934	3	0.003	0.042	3	0.265	0.519
22 M	3+0=3	3	0.25	23.93	3	0.002	0.208	4	0.153	0.468
23 M	3+0=3	3	0.234	24.71	3	0.016	0.313	4	0.149	0.472
24 M	3+0=3	4	0.219	109.057	4	0.006	1.531	4	0.229	2.772
25 M	3+1=4	4	0.203	15.553	4	0.004	0.857	4	0.347	0.581

Table 2: Results for sample with a lot of incompatibility (1)

Samples with a lot of incompatibility constraints (2)										
	Lower	Exact			Heur Heur			Heur Exact		
Sample	Algorithm	S	C	T	S	TL	T	S	TL	T
26 M	3+1=4	4	0.187	28.854	4	0	0.894	4	0.336	0.858
27 M	3+1=4	4	0.218	16.421	4	0.001	0.941	4	0.3	1.01
28 M	3+1=4	4	0.219	1.981	4	0.006	0.065	4	0.317	0.772
29 M	3+1=4	4	0.203	2.387	4	0.006	0.088	4	0.268	0.844
30 M	3+1=4	4	0.202	1.592	4	0.003	0.157	4	0.307	0.685
31 M	3+1=4	4	0.203	2.761	4	0.012	0.383	4	0.391	0.724
32 M	4+0=4	n.	n.	n.	5	0.002	2.169	5	0.393	11.552
33 M	4+1=5	5	0.546	106.704	5	0.015	0.847	5	0.459	1.19
34 M	4+1=5	5	0.374	84.568	5	0.012	1.343	5	0.509	1.542
35 M	4+1=5	5	0.312	149.229	5	0.012	1.407	5	0.572	1.725
36 M	4+1=5	5	0.39	242.69	5	0.002	0.094	5	0.884	2.398
37 M	4+1=5	5	1.169	80.87	5	0.003	0.094	5	0.93	2.435
38 M	4+1=5	5	0.375	32.666	5	0.004	0.236	5	1.026	2.282
39 M	4+1=5	5	0.343	38.096	5	0.002	1.186	5	1.245	10.801
40 M	4+0=4	n.	n.	n.	7	0.005	12.739	6	0.449	102.477
41 M	3+5=8	n.	n.	n.	8	0.009	6	8	0.296	2.768
42 M	3+5=8	n.	n.	n.	8	0.005	6.147	8	0.317	5.652
43 M	3+5=8	n.	n.	n.	8	0.004	6.107	8	0.346	3.711
44 M	3+5=8	n.	n.	n.	8	0.004	0.216	8	0.533	2.459
45 M	3+5=8	n.	n.	n.	8	0.009	0.485	8	0.577	2.688
46 M	3+5=8	n.	n.	n.	8	0.016	0.645	8	0.764	2.948
47 M	3+5=8	n.	n.	n.	8	0.006	4.619	8	0.836	3.456
48 M	3+2=5	n.	n.	n.	9	0.061	43.441	n.d.	n.d.	n.d.
49 M	3+10=13	n.	n.	n.	13	0.007	6.165	13	0.456	1.563
50 M	3+10=13	n.	n.	n.	13	0.027	4.404	13	0.722	2.296

Table 3: Results for sample with a lot of incompatibility constraints (2)

Samples with a lot of incompatibility constraints (3)										
	Lower	Exact			Heur Heur			Heur Exact		
Sample	Algorithm	S	C	T	S	TL	T	S	TL	T
51 M	3+10=13	n.	n.	n.	13	0.012	4.04	13	1.036	3.242
52 M	3+10=13	n.	n.	n.	13	0.006	0.302	13	0.823	3.009
53 M	4+9=13	n.	n.	n.	13	0.002	0.559	13	0.576	2.198
54 M	4+9=13	n.	n.	n.	13	0.002	0.9	13	0.85	2.455
55 M	4+9=13	n.	n.	n.	13	0.006	4.157	13	1.132	3.57
56 M	4+3=7	n.	n.	n.	13	0.003	77.274	7*	2.435*	324.996*
57 M	4+12=16	n.	n.	n.	16	0.002	22.654	16	1.901	6.695
58 M	4+14=18	n.	n.	n.	18	0.004	7.379	18	2.877	8.247
59 M	4+14=18	n.	n.	n.	18	0.003	7.536	18	2.964	7.807
60 M	4+14=18	n.	n.	n.	18	0.004	0.387	18	2.928	8.128
61 M	4+14=18	n.	n.	n.	18	0.001	0.486	18	1.004	3.245
62 M	4+14=18	n.	n.	n.	18	0.007	1.321	18	0.006	1.934
63 M	4+14=18	n.	n.	n.	18	0.004	6.719	18	0.021	12.299

Table 4: Results for sample with a lot of incompatibility constraints(3)

Samples with few incompatibility constraints (1)							
	Lower	Exact			Heur Heur		
Sample	Algorithm	S	C	T	S	TL	T
0	2+0=2	2	0.187	0.468	2	0.003	0.001
1	2+0=2	2	0.062	0.39	2	0.003	0.003
2	2+0=2	2	0.033	0.686	2	0.001	0.009
3	2+0=2	2	0.031	0.405	2	0.001	0.005
4	2+0=2	2	0.047	0.64	2	0.001	0.003
5	2+0=2	2	0.031	0.484	2	0.021	0.002
6	2+0=2	2	0.047	0.578	2	0.009	0.001
7	2+0=2	2	0.031	0.499	2	0.008	0.005
8	2+0=2	2	0.265	1.217	2	0	0.444
9	2+0=2	2	0.14	1.201	2	0.004	0.45
10	2+0=2	2	0.125	83.398	3	0.001	0.422
11	2+0=2	2	0.109	85.55	2	0.002	0.491
12	2+0=2	2	0.126	79.778	2	0.006	0.03
13	2+0=2	2	0.141	80.184	2	0.001	0.041
14	2+0=2	2	0.124	75.224	2	0.001	0.167
15	2+0=2	2	0.109	87.313	2	0.006	0.176
16	2+0=2	2	0.421	1881.431	3	0	2.579
17	2+1=3	3	0.234	11.482	3	0.001	0.816
18	2+1=3	3	0.203	253.725	3	0.003	1.098
19	2+1=3	3	0.218	550.642	3	0.002	1.096
20	2+1=3	3	0.265	365.664	3	0.011	0.071
21	2+1=3	3	0.25	329.258	3	0.011	0.092
22	2+1=3	3	0.25	280.17	3	0.001	0.28
23	2+1=3	3	0.234	30.685	3	0.001	0.692

Samples with a few incompatibility constraints (2)							
	Lower	Exact			Heur Heur		
Sample	Algorithm	S	C	T	S	TL	T
26	2+2=4	4	0.203	33.524	4	0.003	0.713
27	2+2=4	4	0.188	14.242	4	0	0.703
28	2+2=4	4	0.234	1.654	4	0	0.096
29	2+2=4	4	0.234	1.669	4	0.013	0.144
30	2+2=4	4	0.202	1.545	4	0.001	0.195
31	2+2=4	4	0.202	1.436	4	0.002	0.562
32	3+0=3	n.	n.	n.	4	0.002	6.482
33	3+2=5	5	0.593	3.853	5	0	2.452
34	3+2=5	5	0.422	78.967	5	0.009	1.133
35	3+2=5	5	0.39	72.446	5	0.003	1.484
36	3+2=5	5	0.375	89.122	5	0.01	0.101
37	3+2=5	5	0.405	96.096	5	0.006	0.136
38	3+2=5	5	0.359	68.422	5	0.004	0.401
39	3+2=5	5	0.343	56.409	5	0.01	3.261
40	3+0=3	n.	n.	n.	7	0.015	31.019
41	3+5=8	n.	n.	n.	8	0.012	13.29
42	3+5=8	n.	n.	n.	8	0.008	5.107
43	3+5=8	n.	n.	n.	8	0.009	5.932
44	3+5=8	n.	n.	n.	8	0.008	0.257
45	3+5=8	n.	n.	n.	8	0.016	0.51
46	3+5=8	n.	n.	n.	8	0.002	0.866
47	3+5=8	n.	n.	n.	8	0	9.683
48	3+2=5	n.	n.	n.	9	0.003	75.236
49	3+10=13	n.	n.	n.	13	0.004	6.495

Samples with a lot of incompatibility constraints (3)							
	Lower	Exact			Heur Heur		
Sample	Algorithm	S	C	T	S	TL	T
51	3+10=13	n.	n.	n.	13	0.002	6.434
52	3+10=13	n.	n.	n.	13	0.008	0.372
53	3+10=13	n.	n.	n.	13	0.008	0.657
54	3+10=13	n.	n.	n.	13	0.014	1.483
55	3+10=13	n.	n.	n.	13	0.008	4.69
56	3+4=7	n.	n.	n.	13	0.006	127.329
57	3+13=16	n.	n.	n.	16	0.001	21.51
58	3+15=18	n.	n.	n.	18	0.002	16.692
59	3+15=18	n.	n.	n.	18	0.003	15.427
60	3+15=18	n.	n.	n.	18	0.006	0.482
61	3+15=18	n.	n.	n.	18	0.007	0.66
62	3+15=18	n.	n.	n.	18	0.006	1.934
63	3+15=18	n.	n.	n.	18	0.021	12.299

Table 7: Results for sample with few incompatibility constraints (3)

Sample with a lot of incompatibility constraints: average time (TM)			
	Exact	Heur Heur	Heur Exact
TM for samples solved exactly	69.633	0.353	0.916
TM for all samples		3.76	3.873

Table 8: Results for samples with a lot of incompatibility constraints: average time

Sample with few incompatibility constraints: average time (TM)		
	Ottimo	Euri Euri
TM for samples solved exactly	133.719	0.656
TM for all samples		6.199

Table 9: Results for samples with few incompatibility constraints: average time

6 Conclusions

This work adds some constraints in the formulations of bin packing, knapsack and container loading in order to represent the problem of consolidation in the process of groupage.

We added:

- incompatibility among items: some items cannot stay very close to each other.
- some items should stay together
- some items are fragile
- the stability in the bins
- some items should be on the floor of the bin because they are too heavy.

Then we presented two heuristic algorithms to solve the problems. They started with the calculation of the lower bound, leading to the final solution. The two algorithms divide the big problem into several smaller problems. They differ from each other in that one solves the sub-problem using ILOG CPLEX and the other using a heuristic algorithm.

References

[1] H. Dyckhoff. A typology of cutting and packing problems. *European Journal of Operational Research*, 44:145–159, 1990.

[2] Feng Chen and Chung-Yee Lee. Minimizing the makespan in a two-machine cross-docking flow shop problem. *European Journal of Operational Research*, 193(1):59–72, February 2009.

[3] Clausen J. Cordeau J.-F. Wen M., Larsen J. and Laporte G. Vehicle routing with cross-docking. *Journal of the Operational Research Society*, pages 1–1, 2008.

[4] S. Martello, D. Pisinger, and D. Vigo. The three-dimensional bin packing problem, 1997.

[5] J.A. George and D.F. Robinson. A heuristic for packing boxes into a container. *Computers and Operations Research*, 7:147–156, 1980.

[6] The Home Depot, 2012. [Online; 2-January-2012].

[7] Wikipedia. The home depot — wikipedia, the free encyclopedia, 2012. [Online; accessed 31-January-2012].

[8] Ito T. Jin Z. and Ohno K. Issues in the development of approaches to container loading. *JSME Int Journal*, 46(1):60–66, 2003.

[9] Lee S. M. Chen, C. S. and Q. S. Shen. An analytical model for the container loading problem. *European Journal of Operational Research*, 80(1):68–76, January 1995.