LAGRANGEAN RELAXATION WITH CLUSTERS AND COLUMN GENERATION FOR THE MANUFACTURER'S PALLET LOADING PROBLEM

Glaydston Mattos Ribeiro[1]
Luiz Antonio Nogueira Lorena[2*]
LAC – Computer and Applied Mathematics Laboratory
INPE – Brazilian Space Research Institute
12.227-010, São José dos Campos – SP
Tel: +55 12 3945-6555 FAX: +55 12 3945-6357
E-mail: {glaydston[1], lorena[2]}@lac.inpe.br

Abstract

We consider in this paper a new lagrangean relaxation with clusters for the Manufacturer's Pallet Loading Problem (MPLP). The relaxation is based on the MPLP formulated as a Maximum Independent Set Problem (MISP) and represented in a conflict graph that can be partitioned in clusters. The edges inter clusters are relaxed in a lagrangean fashion. Computational tests attain the optimality for some instances considered difficult for a lagrangean relaxation. Our results show that this relaxation can be a successful approach for hard combinatorial problems modeled in conflict graphs. Moreover, we propose a column generation approach for the MPLP derived from the idea behind the lagrangean relaxation proposed.

Keywords: Pallet loading; Lagrangean relaxation; Column generation;

[*] Corresponding author

1. Introduction

The Pallet Loading Problem (PLP) is a well-known optimization problem and consists in arranging the maximum number of boxes onto a pallet without overlapping. The boxes can be rotated by 90^o and the edges must be orthogonal to the pallets' edges. According to Dyckhoff [15], this problem is classified as 2/B/O/C (Two-dimensional, Selection of items, One object, Identical items), therefore, this problem allows a special case of cut and packing problems.

The PLP frequently appears in goods and logistics distribution. Any increase in the number of boxes packed onto the pallet can represent a decrease in logistics costs (Pureza and Morabito [35]). In the literature, there are two types of PLP (Hodgson [22]): The Manufacturer's Pallet Loading Problem (MPLP) and Distributor's Pallet Loading Problem (DPLP). The MPLP considers boxes with identical dimensions and the DPLP deals with boxes of different dimensions. In both cases, the boxes are packed in horizontal layers. Figure 1 shows an example of how the boxes can be arranged according to the problem type.

Figure 1 – Types of PLP: (a) Manufacturer's Pallet Loading Problem - MPLP and (b) Distributor's Pallet Loading Problem - DPLP (Morabito and Morales [31]).

In this paper we consider the MPLP, where given a fixed layer height *h*, the problem consists in arranging the maximum number of identical boxes (*l*,*w*) onto the pallet (*L*,*W*). The boxes faces (*l*,*w*) can be packed in two different orientations in each layer: (*l*,*w*) and (*w*,*l*).

Several optimization methods have been developed to MPLP. The exact algorithms work, basically, with a tree search structure (Dowsland [14]; Bhattacharya et al. [8]; and Alvarez-Valdez et al. [3]). Heuristics can be constructive, dividing the pallet in blocks (Young-Gun and Maing-Kyu [40]), recursive methods (Morabito and Morales [31]) and techniques based in identified structures known as G4 (Scheithauer and Terno [37]) and L (Lins et al. [28] and Birgin et al. [9]). Some other works applied metaheuristics, such as Tabu Search (Pureza and Morabito [35] and Alvarez-Valdes et al. [2]) and Genetic Algorithms (Herbert and Dowsland [19]).

There are also upper bounds that consider the problem's geometry, and they allowed us to state the solution quality of relaxations and heuristics. Letchford and Amaral [27] presented a good review of the known upper bounds for the MPLP and conducted a detailed analysis to determine which bounds dominate others. They compared the area bound, Barnes bound (Barnes [6]), Isermann bound (Isermann [24]) and the packing bound that is a linear relaxation of the formulation proposed by Beasley [7]. Their results have shown that the linear relaxation dominates the other examined bounds. More details can be obtained in Letchford and Amaral [27].

The MPLP can also be seen as a Maximum Independent Set Problem (MISP) (Dowsland [14]). The MPLP can be represented by a conflict graph where each vertex indicates the left-lower-corner of a box placed on the pallet, and the edges represent the possible overlapping between these vertices.

Some problems represented in conflict graphs are well-adapted for a previous partitioning phase (clustering). This feature generates small scale sub-graphs (clusters) that are similar to the original one. Thus, if we remove the edges that are connecting all sub-graphs, the sub-problems can be independently solved providing bounds for the original problem. Besides, these edges correspond to constraints, and if they are relaxed in a lagrangean way, the bound can be improved and used to efficiently search optimal solutions.

Thus, this paper explores this characteristic. The conflict graph for the MPLP is generated and partitioned in clusters. The edges connecting these clusters are relaxed in a lagrangean way. Each cluster is a sub-problem and can be solved independently by some commercial solver. The bound is obtained and the lagrangean multipliers are updated using a subgradient method. Again, the sub-problems are solved independently, and so on until some stopping test is reached.

This lagrangean relaxation is called lagrangean relaxation with clusters, or simply LagClus. It was applied in point-feature cartographic label problems with better results than all reported in the literature (Ribeiro and Lorena [36]). The LagClus application to MPLP could ensure the optimality for instances that are considered difficult for lagrangean and linear relaxations.

Moreover, given that sub-problems generate solutions for each cluster independently, we also present a column generation approach for the MPLP. We present a Dantzig-Wolfe decomposition for the MPLP and some results for instances reported in the literature. The results show that the restricted master problem, obtained at the final of the column generation process, provided the optimal solution for all tests.

The structure of the paper is as follows. In Section 2, we present the MPLP and MISP formulation and a brief literature review of the MISP. In Section 3 we present the lagrangean relaxation with clusters proposed for the Beasley's MPLP formulation. In Section 4 we show the Dantzig-Wolfe decomposition proposed for MPLP using an analogy with MISP. In Section 5 we present computational results of these two proposed approaches, and finally, some comments are discussed in Section 6.

2. The MPLP and MISP formulation

As mentioned by several works in the literature (Morabito and Morales [31]; Morabito and Farago [29]; Alvarez-Valdes et al. [3]), the MPLP can be formulated using a particular case of the Beasley's [7] formulation for the two-dimensional non-guillotine cutting problem. Let L and W be the pallet length and width, respectively, such that $L \geq W$, and, l and w, the box length and width, respectively, such that $l \geq w$ and $l \leq Min(L,W)$. To represent all possible ways to packing a box, let be $(l_1,w_1)=(l,w)$ and $(l_2,w_2)=(w,l)$. Thus, these possible positions can be represented by $(l_i,w_i)_{i=1,2}$ that indicates the box length and width considering the orientation i.

To represent the boxes position onto the pallet, let X and Y be two sets that are used to define the coordinates (p,q) of the box left-lower-corner. These sets can be described by:

$$X = \left\{ p \in Z^{+} \mid p = \sum_{i=1}^{2} l_i b_i,\ 0 \le p \le L - w,\ b_i \ge 0 \text{ and integer},\ i = 1,2 \right\} \tag{1}$$

$$Y = \left\{ q \in Z^{+} \mid q = \sum_{i=1}^{2} w_i b_i,\ 0 \le q \le W - w,\ b_i \ge 0 \text{ and integer},\ i = 1,2 \right\} \tag{2}$$

These sets were introduced by Christofides and Whitlock [13] and they are called *normal sets*. The restriction of the boxes positions to these sets does not imply in loss of generality.

Let a be a function that describe overlapping constraints between boxes. This function can be obtained in advance for each vertex (p,q) in relation to some other vertex (r,s), for each orientation i, where $p \in X | p \le L-l_i$, $q \in Y | q \le W-w_i$, $r \in X$, $s \in Y$, and $i=1,2$. Thus, this function can be expressed by:

$$a_{ipqrs} = \begin{cases} 1, & \text{If } \ 0 \le p \le r \le p + l_i - 1 \le L - 1 \ \text{ and } \ 0 \le q \le s \le q + w_i - 1 \le W - 1 \\ 0, & \text{Otherwise} \end{cases} \tag{3}$$

Now, let $x_{ipq} \in \{0,1\}$ be a decision binary variable for all $p \in X | p \le L-l_i$, $q \in Y | q \le W-w_i$, and $i=1,2$. If $x_{ipq}=1$, one box is placed in pallet coordinates (p,q) with orientation i, otherwise, $x_{ipq}=0$.

Then the MPLP can be formulated as (Beasley [7]):

$$v(MPLP) = Max\left(\sum_{i=1}^{2} \sum_{\{p \in X | p \le L - l_i\}} \sum_{\{q \in Y | q \le W - w_i\}} x_{ipq} \right) \tag{4}$$

Subject to:

$$\sum_{i=1}^{2}\sum_{\{p\in X \mid p\leq L-l_i\}}\sum_{\{q\in Y \mid q\leq W-w_i\}} a_{ipqrs}\, x_{ipq} \leq 1, \quad \forall r\in X \text{ and } s\in Y \tag{5}$$

$$x_{ipq} \in \{0,1\} \;\forall i = 1 \ldots 2,\; p\in X \mid p \leq L-l_i, \text{ and } q\in Y \mid q \leq W-w_i \tag{6}$$

The constraints set (5) avoids overlapping between positions. Each individual constraint ensures that a particular “square” is covered by at most one box. The constraints set (6) ensures that all variables are binaries.

As mentioned before in Section 1, this problem can also be formulated as a Maximum Independent Set Problem (MISP). It is a classic problem, quite studied in the literature. The MISP normally appears embedded in applications and arises in several fields such as in coding theory, combinatorial auctions, computer vision and protein chemistry (see Bomze et al. [10]).

Due to MISP wide application area, there are several approaches proposed in the literature. Exact techniques include explicit enumeration of maximal independent sets (Bron and Kerbosch [11]), Branch-and-Bound (Balas and Xue [4]; Östergard [33]), Branch-and-Price (Hicks et al. [21]) and continuous formulations under Branch-and-Bound (Barnes [5]). Besides, several heuristics were proposed such as vertices contraction algorithms (Hertz [20]), and the greedy heuristic of Kopf and Ruhe [26]. There are still local search heuristics that try improving some solution given by another method, for example, by a greedy metaheuristic (see Feo et al [16]).

There are also several applications of metaheuristics for solving the MISP. Aarts and Korst [1] have used a Simulated Annealing, Bui and Eppley [12] Genetic Algorithms, and Gendreau et al. [17] have applied Tabu Search.

Among all works related before, the Branch-and-Price of Hicks et al. [21] is interesting and the idea behind our work is based in this paper. They have worked with the Maximum Weight Independent Set Problem (MWISP) that differs of the MISP because the MWISP considers weight in the edges. The authors generated a conflict graph for the MWISP and partitioned it. All sub-graphs (sub-problems) are considered in a Branch-and-Price algorithm, where each sub-problem generates columns for a Restricted Master Problem (RMP). Their results were good for several instances reported in the literature for the MWISP.

So, the MISP can be modeled as following. Let $G=(V,E)$ be a graph where V is a set of vertices v, and E a set of edges (u,v) such that $u,v\in V$ and $u\neq v$. Consider that there are no weights assigned to the vertices or edges. Thus, the MISP consists in obtain a subset $V'\subseteq V$ such that all pairs of vertices of V' are not adjacent, that is, if $r,s\in V'$, then $(r,s)\notin E$. Therefore, the MISP can be formulated by:

$$v(MISP) = Max\left(\sum_{v\in V} x_v\right) \tag{7}$$

Subject to:

$$x_u + x_v \leq 1 \quad \forall\,(u,v)\in E \tag{8}$$

$$x_u \in \{0,1\} \quad \forall\, u \in V \tag{9}$$

If $x_v=1$ the vertex v is in the independent set, otherwise, $x_v=0$. The constraints set (8) ensures that two adjacent vertices cannot be simultaneously in the independent set. The constraint set (9) indicates that all variables x_v are binaries.

The formulation (7)–(9) for the MISP can be used for the MPLP as mentioned by Dowsland [14], however it produces more constraints than in formulation defined in (4)-(6). It happens because the Beasley's [7] formulation uses cliques, reducing the number of constraints. For instance, consider a pallet with dimensions $(L,W)=(5,4)$ and boxes $(l,w)=(3,2)$. Figure 2(a) shows the formulation produced by model (4)-(6), Figure 2(b) shows the conflict graph obtained from formulation shown in Figure 2(a), and Figure 2(c) shows the formulation produced by (7)-(9). As expected, formulation (7)–(9) produces more constraints than formulation (4)-(6), but all constraints are considered implicitly in MPLP formulation.

Figure 2 – Comparison between MPLP and MISP formulation. (a) MPLP formulation, (b) conflict graph, and (c) MISP formulation.

3. The Lagrangean relaxation with clusters (LagClus)

The LagClus takes advantage that some conflict graphs are well-adapted for previous partitioning phase. So, from Beasley's [7] formulation, a conflict graph can be obtained as we showed at Figure 2(b).

The LagClus can be applied to the MPLP by following the steps:

a) Create the conflict graph from MPLP formulation and apply a graph partitioning heuristic to divide the conflict graph in $\overline{P}$ clusters. This step generates $\overline{P}$ sub-graphs (sub-problems);

b) Relax the constraints present in MPLP formulation that correspond to vertices in different clusters. In each relaxed clique, verify if there are pairs of vertices that are in the same cluster, and if they exist, add to respective cluster one adjacent constraint between each pair found;

c) The Lagrangean relaxation obtained is divided in $\overline{P}$ sub-problems and solved.

Note what happens at step b). If some clique constraint is relaxed, it must be decomposed and each one of their edges must be analyzed. If some edge is connecting two vertices in the same cluster, it must be appended to the respective cluster. This procedure is essential to become the relaxation stronger and to avoid invalid solution for some cluster.

The example in the Figure 3 explains the partitioning phases. Figure 3(a) has two well-defined clusters. Figure 3(b) shows all edges connecting the clusters that are relaxed in LagClus, and Figure 3(c) shows the two sub-graphs (or two sub-problems) similar to the original problem that can be separated and solved independently.

Figure 3 – Lagrangean relaxation with clusters. (a) Conflict graph, (b) edges connecting the clusters, and (c) clusters or sub-problems.

For the computational tests, we have implemented a subgradient algorithm to solve the Lagrangean dual (Parker and Rardin [34]; Narciso and Lorena [32]). The step size control in the algorithm was the one proposed by Held and Karp [18], beginning with 2 and halving it whenever the upper bound does not decrease for 15 successive iterations. The stopping tests used are: step less or equal than 0.005; difference between the best lower and upper bounds less than 1; and the length of the subgradient vector equal to zero. The lagrangean multipliers are initialized with zero.

Figure 4 – Verify and improvement heuristic used in LagClus process.

Before the first iteration of the subgradient algorithm, we used a simpler form of the block heuristic proposed by Smith and De Cani [38] to generate an initial solution. This solution is used in step size definition and can be substituted by a solution provided by the LagClus, made feasible to MPLP. This heuristic, called VI, identifies all vertices present in relaxed solution that are in conflict, removing from this solution the vertex with the largest number of vertices in conflict. This process is repeated until the heuristic produces a feasible solution to MPLP. After that, it tries to introduce other vertices in this solution aiming to get the maximum number of independent vertices. These other vertices are the remaining vertices, not including the first vertices removed from the relaxed solution. The VI heuristic is shown in Figure 4. The step sizes of the subgradient algorithm are updated considering the LagClus solutions and the feasible solutions obtained with VI or the block heuristic.

4. Dantzig-Wolfe decomposition and column generation approach for the MPLP

The classic implementation of a column generation approach uses a coordinator problem and sub-problems that generate columns. The coordinator problem or Restricted Master Problem (RMP), guides the sub-problems by their dual variables for search new columns that introduce new information for the RMP.

Using the LagClus idea of partitioning, the Dantzig-Wolfe decomposition proposed by Hicks et al. [21] to MWISP can be reformulated to MISP, and consequently to MPLP. Let $\overline{P}$ be the number of clusters formed after the conflict graph partitioning, as shown at Figure 3(b). Thus, the MISP can be rewritten as:

$$v(MISP) = Max\left(\sum_{p=1}^{\overline{P}} x^p\right) \tag{10}$$

Subject to:

$$A^1x^1 + A^2x^2 + \cdots + A^{\overline{P}}x^{\overline{P}} \leq \mathbf{1} \tag{11}$$

$$\begin{array}{llll} D^1x^1 & & & \leq \mathbf{1} \\ & D^2x^2 & & \leq \mathbf{1} \\ & & \cdots & \leq \vdots \\ & & & D^{\overline{P}}x^{\overline{P}} \leq \mathbf{1} \end{array} \tag{12}$$

$$x^1 \in B^{n_1} \quad \cdots \quad \cdots \quad x^{\overline{P}} \in B^{n_{\overline{P}}} \tag{13}$$

Where:

- x^p is a decision variables vector assigned to the cluster p;

- A^p is a binary matrix with dimensions M x $|V|$ that represents the variable coefficients x^p assigned to cluster p and also appearing at the M adjacent constraints between cluster;
- D^p is a binary matrix with dimensions $|E|$-M x $|V|$ that represents the variable coefficients x^p assigned to adjacent constraints that are inside the cluster p;
- the B^{n_p} is a vector of binary variables assigned to cluster p of dimension n_p.

Note that if we remove the constraint set defined by Equation (12), it allows us to divide the problem in $\overline{P}$ distinct sub-problems as shown at Figure 3(c).

Now, applying the Dantzig-Wolfe decomposition (DW) for the linear relaxation (LP) of the problem (10)-(13), we have the following problem:

$$v(MISP_{DW})_{LP} = Max\left(\sum_{p=1}^{\overline{P}} \sum_{j \in J_p} \lambda_{jp} \overline{x}^{jp}\right) \tag{14}$$

Subject to:

$$\sum_{p=1}^{\overline{P}} \sum_{j \in J_p} \lambda_{jp}\left(A_p \overline{x}^{jp}\right) \leq 1 \tag{15}$$

$$\sum_{j \in J_p} \lambda_{jp} = 1 \qquad \forall\, p \in \{1 \ldots \overline{P}\} \tag{16}$$

$$\lambda_{jp} \geq 0 \qquad \forall\, p \in \{1 \ldots \overline{P}\} and\ j \in J_p \tag{17}$$

Where:

- J_p is a set of extreme points of the cluster (sub-problem) p;

- $\bar{x}^{jp}$ is a vector of dimension $|V_p|$ that represents the extreme point $j \in J_P$;
- λ_{jp} is a decision variable that represents the extreme point $j \in J_p$.

The sub-problems $p \in \{1 \ldots \bar{P}\}$ are MISPs defined by

$$v(MISP^{p}) = Max\left\{\left(1 - A_p^T \Delta\right)\bar{x}^{jp}\right\} \quad (18)$$

Subject to:

$$D^{p}\bar{x}^{jp} \leq 1 \quad (19)$$

$$\bar{x}^{jp} \in B^{n_p} \quad (20)$$

Where Δ is an M-dimensional vector of dual variables corresponding to constraints set (16).

Considering the restricted master problem (RMP) of the decomposition above, i. e., a restricted number of columns, a new column provided by a sub-problem *p* is an improving column, if $v(MISP^{p}) - \beta_p > 0$, where β_p is a dual variable associated with the p^{th} convexity constraint (17).

The LagClus proposed in Section 3 can also be obtained directly from RMP model of (14)-(17) using the formulation:

$$v(L_{\Delta}MISP) = \sum_{p=1}^{\bar{P}}\left\{v(MISP^{p})\right\} + \sum_{\delta_i \in \Delta}\delta_i \quad (21)$$

We used a RSF (Recursive-Smallest-First) heuristic, proposed by Yamamoto and Lorena [39] for point-feature cartographic label problem, to generate the initial pool of columns that is composed only by feasible solutions for the MPLP. Originally, RSF begins choosing the smallest vertex degree, and turns inactive this vertex and its adjacent vertices. Considering the list of active vertices, the degree for each vertex is calculated and the algorithm is repeated upon this new set of vertices until there are no more active vertices. The selected vertices form an independent set and a feasible solution for the MPLP.

We have modified the original RSF to generate different solutions for MPLP, and, consequently, generate a good initial set of columns. Instead of starting the RSF choosing the smallest vertex degree, we randomly select one vertex, and the algorithm continues as the original version. This process is repeated until a number of desired solutions. Let *ND* be this number, then, the number of columns appended to RMP is $ND * \overline{P}$ because each cluster generates one column.

To give a better idea of the decomposition and column generation approach described above, Figure 5 shows a diagram with flows and the main steps used. Note that the shadow area represents the iteration necessary to produce improved columns.

Figure 5 – Diagram of the steps used for the column generation approach

5. Computational results

There are several works that present instances of MPLP, as in Dowsland [14], Letchford and Amaral [27], Morabito and Morales [31], Alvarez-Valdez et al. [2] and Pureza and Morabito [35]. Other works propose instances obtained from real problems arising in carriers, such as in Morabito et al. [30] and in Morabito and Farago [29]. In this work, we present results for some of these instances that are divided in three groups. The first one has 10 instances (L1-L10) proposed by Letchford and Amaral [27] and they are considered difficult for lagrangean and linear relaxation approaches. The second has 10 randomly selected instances (L11-L10) from COVER II proposed by Dowsland [14], and the last has 10 randomly selected instances (L21-L30) obtained from COVER III, proposed by Alvarez-Valdez et al. [2], that do not present known optimal solutions.

The code in C++ and the tests are performed in a computer with Pentium IV processor and 512 MB of RAM memory. The sub-problems, either for LagClus or for sub-problems and RMP, were solved by CPLEX 7.5 (ILOG [23]). For the graph partitioning task, we have used the METIS (Karypis and Kumar [25]) that is a well-known heuristic for graph partitioning problems. Given a conflict graph G and a pre-defined number $\overline{P}$ of clusters, the METIS divides the graph in $\overline{P}$ clusters minimizing the number of edges with terminations in different clusters.

Tables 1, 2 and 3 report results for LagClus using the instances defined before. The columns are:

- Instance – Name of the instance;

- L and W – Pallet length and width, respectively;
- l and w – Box length and width, respectively;
- Optimal solution – Known optimal solution;
- Best known solution– Best feasible solution reported in literature;
- Area Bound – Area bound given by $\lfloor (L*W)/(l*w) \rfloor$ (where $\lfloor z \rfloor$ denotes rounding down to the nearest integer);
- Barnes bound – Bound provided by Barnes [6];
- LP bounds – Linear relaxation of model (4) –(6) solved by CPLEX;
- Lower bound – Lower bound found by VI heuristic or block heuristic;
- Upper bound – Upper bound provided by LagClus;
- GAP LB (%) – Percentage deviation gap from the optimal/best known solution to the best lower bound:

$$Gap\ LB = \frac{(Optimal\ solution/Best\ solution - Lower\ bound)}{Optimal\ solution/Best\ solution} * 100;$$

- GAP UB (%) – Percentage deviation gap from the optimal/best known solution to the best upper bound:

$$Gap\ UB = \frac{(Upper\ bound - Optimal\ solution/Best\ solution)}{Optimal\ solution/Best\ solution} * 100;$$

- Time – Time in seconds elapsed by LagClas reaching some of the stop conditions;
- Iterations – Number of iterations used by LagClus.

The number of clusters for all sets was previously obtained. We analyzed the tradeoff between the quality of the upper bounds and the computational times. We used number of clusters that provide good upper bounds with an acceptable time.

As can be seen at Tables 1, 2 e 3, the lower bounds provided are very close to the optimal or to the best known solution. In worst case, our results differ in one box. The LagClus upper bounds are almost the same as the LP bounds, and the computational times are comparable.

Table 1 presents all results for instances L1-L10 with 2 clusters. These instances are considered difficult for a lagrangean relaxation, but the LagClus was able to prove the optimal solution for one instance (L7) and for the others, the dual bound are very close to the optimal solution (almost 1 box).

Table 1 – Computational results for 10 examples proposed by Letchford and Amaral [27], considered difficult for a lagrangean relaxation.

Table 2 reports results for instances L11-L20 and for this group we have used 5 clusters. The LagClus was able to verify the optimality in 60% of instances: L13, L14, L16, L18, L19 and L20. Confirming the results shown in Table 1, the dual bounds are very close to the optimal solution.

Table 2 – Computational results for 10 instances randomly obtained from COVER II (Dowsland [14]).

Table 3 presents all results provided for the last ten instances with 15 clusters. Differently from the results in Table 1 and 2, the LagClus was not able to find the optimality of the lower bound. In fact, this means that the last group is hard to solve. Table 2 has shown that the LagClus is a stronger relaxation, but it did not found optimal solutions for these last 10 instances.

Table 3 – Computational results for 10 randomly examples obtained from COVER III (Alvarez-Valdez [2]), upon all instances that do not present optimal solution known.

To show the relation between the quality of the upper bounds and the time consuming requirements, we decided to modify the number of clusters. The instance L7 was used for tests and the results are shown at Table 4. As expected, when the number of clusters is increased, the LagClus provided poor upper bounds but the time reduces.

Table 4 – Computational results for instance L7 varying the number of clusters.

The results found using the column generation approach for instances L1-L10 are shown at Table 5. The columns are:

- Instance – Instance name;
- Initial number of columns – Initial number of columns considered in RMP;
- Initial RMP – Initial value provided by RMP;
- Final number of columns – Final number of columns considered in RMP;
- Final RMP – Final value provided by RMP;
- Time CG (s) – Time consumed at the column generation process;
- Solution (IP) – Value obtained by RMP solved using integer variables;
- Time IP (s) – Time consumed for solving the integer RMP.

We considered the same partitioning done as shown at Table 1, 2 and 3. In this work, we preferred stop the column generation approach when no more columns are appended to the RMP, that is, when no improving column is found, and we also did not use tests to remove unproductive columns from the RMP.

The results shown at Table 5 validate the Dantzig-Wolfe decomposition, although the computational times were high in some cases. Indeed, the last RMP solved considering only integer variables, provided the optimal solutions in reasonable times (less than 5,10s for all instances).

Table 5 – Computational results using the column generation approach.

Figure 6 shows the column generation approach behavior and the LagClus bound obtained in Equation (22). Note that the RMP and LagClus bounds tend to be equivalent and are close to the area/Barnes bound.

Figure 6 – LagClus and RMP behavior.

The column generation results for the last two groups L11-L30 were in course, but can be obtained in reasonable times only after removing unproductive columns or to stop the column generation process using the converging RMP and LagClus bounds. They will be reported in a further work.

6. Conclusions

This work has presented a new lagrangean relaxation and a new column generation approach for the manufacturer's pallet loading problem. The LagClus deals the MPLP as a Maximum Independent Set Problem (MISP) presenting a conflict graph that can be partitioned in clusters. The partitioning also permits a column generation approach to MPLP.

The LagClus reaches to the optimality of some solutions and provided goods bounds for instances considered difficult for a lagrangean relaxation. The column generation has also presented good results for some instances as shown at Table 4, and we have demonstrated that the LagClus can be obtained using the dual variables provided by the column generation approach.

Continued efforts are intended for a column generation algorithm to solve large scale MPLP instances. Besides, a desired complement to our studies will be a Branch-and-Price algorithm for the MPLP.

Acknowledgments: The authors acknowledge Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for partial financial research support. The authors thank the anonymous referees for their useful suggestions and comments.

References

1) Aarts, E. and Korst, J. **Simulated Annealing and Boltzmann Machines**. Chichester, UK: J. Wiley & Sons, 1989.
2) Alvarez-Valdez, R.; Parreño, F. and Tamarit, J. M. A tabu search algorithm for pallet loading problem. **OR Spectrum**, 27(1): 43-61, 2005.

3) Alvarez-Valdez, R.; Parreño, F. and Tamarit, J. M. A branch-and-cut algorithm for the pallet loading problem. **Computer and Operations Research**, 32(11): 3007-3029, 2005.
4) Balas, E. and Xue, J. Weighted and unweighted maximum clique algorithms with upper bounds from fractional coloring. **Algorithmica**, 15(5): 397-412, 1996.
5) Barnes, E. R. A branch-and-bound procedure for the largest clique in a graph. In: Pardalos, P. M. (ed.) **Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems**. Boston: Kluwer Academic Publishers, 2000.
6) Barnes, F. W. Packing the maximum number of *m x n* tiles in a large *p x q* rectangle. **Discrete Mathematics**, 26: 93-100, 1979.
7) Beasley, J. An exact two-dimensional non guillotine cutting tree search procedure. **Operations Research**, 33: 49-64, 1985.
8) Bhattacharya, R.; Roy, R.; and Bhattacharya, S. An exact depth-first algorithm for the pallet loading problem. **European Journal of Operational Research**, 110: 610-625, 1998.
9) Birgin, E. G.; Morabito, R. and Nishihara, F. H. A note on an L-approach for solving the manufacturer's pallet loading problem. **Journal of the Operational Research Society**, 2005. To Appear.
10) Bomze, I. M.; Budinich, M.; Pardalos, P. M. and Pelilo, M. The maximum clique problem. In: Du, D. and Pardalos, P. M. (eds) **Handbook of Combinatorial Optimization**. Boston: Kluwer Academic Publishers, 1999.
11) Bron, C. and Kerbosch, J. Finding all cliques of an undirected graph. **Communications of the ACM**, 16(9): 575-577, 1973.
12) Bui, T. N. and Eppley, P. H. A hybrid genetic algorithm for the maximum clique problem. In: Proceedings 6th International Conference on Genetic Algorithms. **Annals**, p. 478-484, 1995.
13) Christofides, N. and Whitlock. C. An algorithm for two-dimensional cutting problems. **Operations Research**, 25: 30-44, 1977.
14) Dowsland, K. An exact algorithm for the pallet loading problem. **European Journal of Operational Research**, 31: 78-84, 1987.
15) Dyckhoff, H. A typology of cutting and packing problems. **European Journal of Operational Research**, 44: 145-159, 1990.
16) Feo, T. A.; Resende, M. G. C. and Smith, S. H. A greedy randomized adaptive search procedure for maximum independent set. **Operations Research**, 42: 860-878, 2000.
17) Gendreau, M.; Salvail, L. and Soriano, P. Solving maximum clique problem using a tabu search approach. **Annals of Operations Research**, 41: 385-403, 1999.
18) Held, M. and Karp, R. M. The traveling salesman problem and minimum spanning trees: part II. **Mathematical Programming**, 1: 6-25, 1971.
19) Herbert, A. and Dowsland, K. A family of genetic algorithm for the pallet loading problem. In: Osman, I. H. and Kelly, J. P., editors. **Metaheuristics: theory and applications**. Dordrecht: Kluwer Academic Publishers, p. 378-406, 1996.
20) Hertz, A. A fast algorithm for coloring Meyniel graphs. **Journal of Combinatorial Theory B**, 50(2): 231-240, 1990.
21) Hicks, I. V.; Warren, J. S.; Warrier, D. and Wilhenlm, W. E. **A branch-and-price approach for the maximum weight independent set problem**. Texas A & M University: Department of Industrial Engineering, 2004. Available at http://ie.tamu.edu/People/faculty/Hicks. Accessed January 22th, 2004.
22) Hodgson, T. A combined approach to the pallet loading problem. **IIE Transactions**, 14(3): 176-182, 1982.
23) **ILOG CPLEX 7.5 Reference Manual**. 7.5v. 610p. ©Copyright by ILOG, France, 2001.
24) Isermann, H. Ein planungssytem zur optimierung der palettenbeladung mit kongruenten rechteckigen versandgebinden. **OR Spectrum**, 9: 235-249, 1987.
25) Karypis, G. and Kumar, V. Multilevel k-way partitioning scheme for irregular graphs. **Journal of Parallel and Distributed Computing**, 48(1): 96-129, 1998.
26) Kopf, R. and Ruhe, G. A computational study of the weighted independent set problem for general graphs. **Found. Control Engi.**, 12(4): 167-180, 1987.
27) Letchford, A.N. and Amaral, A. Analysis of upper bounds for the pallet loading problem. **European Journal of Operational Research**, 3(132): 582-593, 2001.
28) Lins, L.; Lins, S. and Morabito, R. An L-approach for packing (l,w)-rectangles into rectangular and *L*-shaped pieces. **Journal of the Operational Research Society**, 54: 777-789, 2003.

29) Morabito, R. and Farago, R. A tight lagrangean relaxation bound for the manufacturer's pallet loading problem. **Studia Informatica Universalis**, 2(1): 57-76, 2002.
30) Morabito, R.; Morales, S. R. and Widmer, J. A. Loading optimization of palletized products on trucks. **Transportation Research Part E**, 36: 285-296, 2000.
31) Morabito, R. and Morales, S. R. A simple and effective heuristic to solve the manufacturer's pallet loading problema. **Journal of the Operational Research Society**, 49: 819-828, 1998.
32) Narciso, M. G. and Lorena, L. A. N. Lagrangean/Surrogate relaxation for generalized assignment problems. **European Journal of Operational Research**, 114: 165-177, 1999.
33) Östergard, P. R. J. A fast algorithm for the maximum clique problem. **Discrete Applied Mathematics**, 120: 197-207, 2002.
34) Parker, R. G. and Rardin, R. L. **Discrete Optimization**. New York: Academic Press, 1988.
35) Pureza, V. and Morabito, R. Some experiments with a simple tabu search algorithm for the manufacturer's pallet loading problem. **Computers and Operations Research**, 33(3):804-819, 2006.
36) Ribeiro, G. M. and Lorena, L. A. N. Lagrangean relaxation with clusters for point-feature cartographic label placement problems. **ECCO XVII post conference special issues**, 2005. Submitted.
37) Scheithauer, G. and Terno, J. The G4-heuristic for the pallet loading problem. **Journal of the Operational Research Society**, 47: 511-522, 1996.
38) Smith, A. and De Cani, P. An algorithm to optimize the layout of boxes in pallets. **Journal of the Operational Research Society**, 31: 573-578, 1980.
39) Yamamoto, M. and Lorena, L. A. N. A constructive genetic approach to point-feature cartographic label placement. In: Ibaraki, T.; Nonobe, K.; Yagiura, M. (eds) **Metaheuristics: Progress as Real Problem Solvers**. Netherlands: Kluwer Academic Publishers, pages 285-300, 2005.
40) Young-Gun, G. and Maing-Kyu, K. A fast algorithm for two-dimensional pallet loading problems of large size. **European Journal of the Operational Research**, 134: 193-202, 2001.

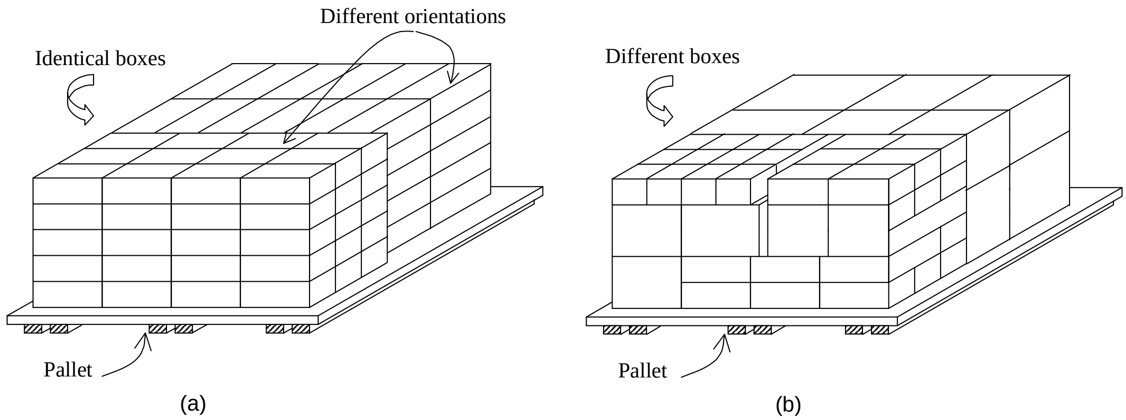


Figure 1 – Types of PLP: (a) Manufacturer's Pallet Loading Problem - MPLP and (b) Distributor's Pallet Loading Problem - DPLP (Morabito and Morales [31]).

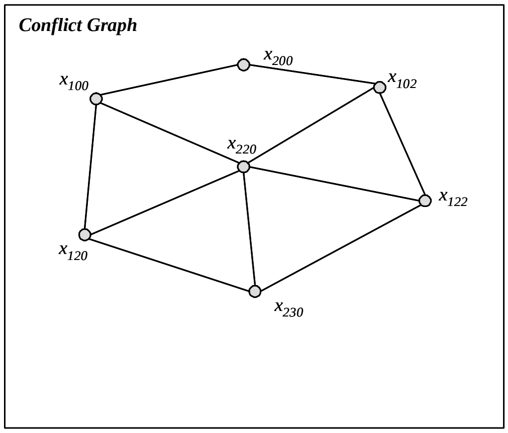
MPLP

$$v(MPLP) = Max\ (x_{100} + x_{102} + x_{120} + x_{122} + x_{200} + x_{220} + x_{230})$$

Subject to:

$$x_{100} + x_{200} \leq 1$$
$$x_{102} + x_{200} \leq 1$$
$$x_{100} + x_{120} + x_{220} \leq 1$$
$$x_{102} + x_{122} + x_{220} \leq 1$$
$$x_{120} + x_{220} + x_{230} \leq 1$$
$$x_{122} + x_{220} + x_{230} \leq 1$$
$$x_{100}, x_{102}, x_{120}, x_{122}, x_{200}, x_{220}, x_{230} \in \{0,1\}$$

(a)



(b)

MISP

$$v(MISP) = Max\ (x_{100} + x_{102} + x_{120} + x_{122} + x_{200} + x_{220} + x_{230})$$

Subject to:

$$x_{100} + x_{120} \leq 1$$
$$x_{100} + x_{200} \leq 1$$
$$x_{100} + x_{220} \leq 1$$
$$x_{102} + x_{122} \leq 1$$
$$x_{102} + x_{200} \leq 1$$
$$x_{102} + x_{220} \leq 1$$
$$x_{120} + x_{220} \leq 1$$
$$x_{120} + x_{230} \leq 1$$
$$x_{122} + x_{220} \leq 1$$
$$x_{122} + x_{230} \leq 1$$
$$x_{220} + x_{230} \leq 1$$
$$x_{100}, x_{102}, x_{120}, x_{122}, x_{200}, x_{220}, x_{230} \in \{0,1\}$$

(c)

Figure 2 – Comparison between MPLP and MISP formulation. (a) MPLP formulation, (b) conflict graph, and (c) MISP formulation.

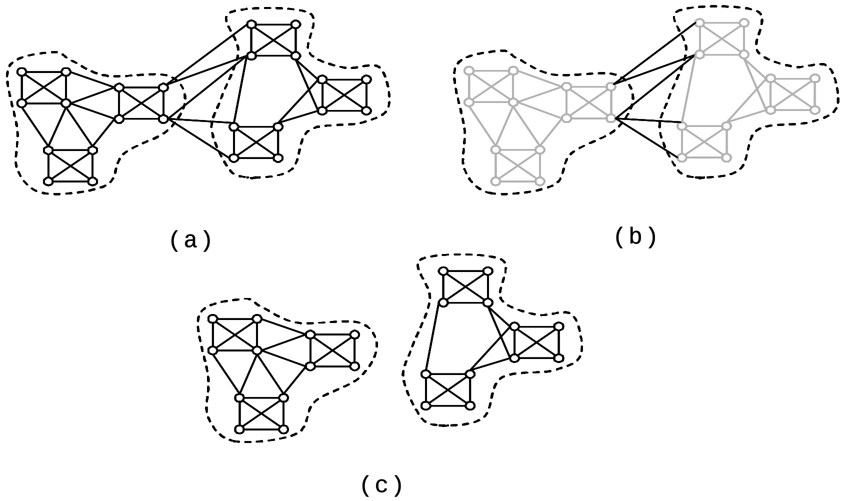


Figure 3 – Lagrangean relaxation with clusters. (a) Conflict graph, (b) edges connecting the clusters, and (c) clusters or sub-problems.

```
Verify and Improvement Heuristic - VI
1. Make feasible solution vector equal to the relaxed
     solution given by LagClus;
2. While not obtain a feasible solution Do
     3. For each vertex i in feasible solution vector,
     define the number of vertices j that are in
     conflict with i;
     4. Sort in decrease order the feasible solution
     vector according to number of conflicts;
     5. Remove the first vertex from de feasible
     solution vector;
6. End while;
7. Verify among the other vertices not present in
     feasible solution and not present in that set
     removed from the feasible solution in step 2, if
     there are vertices that can be inserted in
     feasible solution.
```

Figure 4 – Verify and improvement heuristic used in LagClus process.

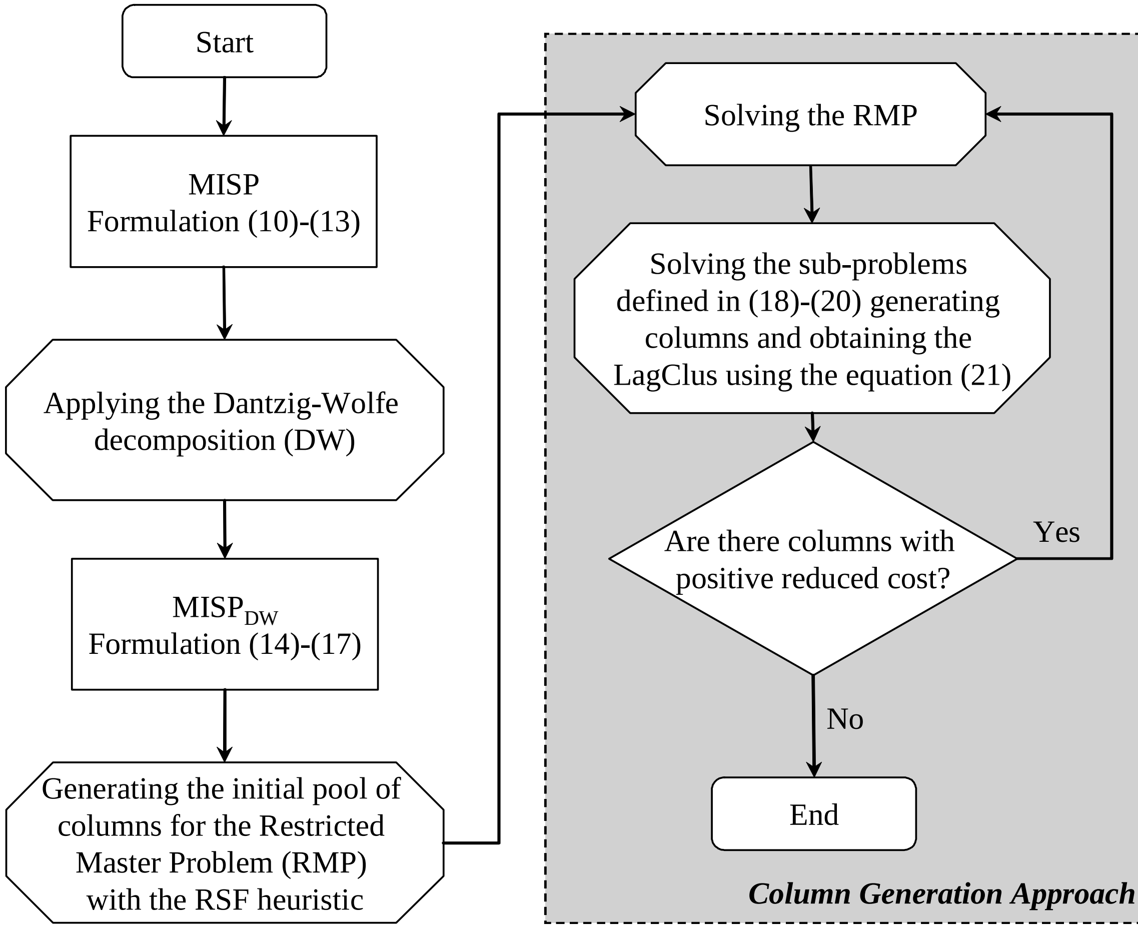


Figure 5 – Diagram of the steps used for the column generation approach

Table 1 – Computational results for 10 examples proposed by Letchford and Amaral [2], considered difficult for a lagrangean relaxation.

Instance	L	W	l	w	Optimal solution	Area bound	Barnes bound	Linear Relaxation		LagClus					
								LP bound	Time (s)	Lower bound	Upper bound	GAP LB (%)	GAP UB (%)	Time (s)	Iterations
L1	32	22	5	4	34	35	35	35.0000	0.27	34	35.0022	0.00	2.95	25.00	145
L2	32	27	5	4	42	43	43	43.0000	0.26	42	43.0023	0.00	2.39	88.00	145
L3	40	26	7	4	36	37	37	37.0000	0.15	36	37.0020	0.00	2.78	98.00	145
L4	40	33	7	4	46	47	47	47.0000	0.82	46	47.0023	0.00	2.18	276.00	145
L5	53	26	7	4	48	49	49	49.0000	0.59	48	49.0110	0.00	2.11	318.00	145
L6	37	30	8	3	45	46	46	46.0000	0.73	45	46.0334	0.00	2.30	202.00	145
L7	81	39	9	7	49	50	50	50.0000	1.34	49	49.9936	0.00	2.03	257.00	67
L8	100	64	17	10	36	37	37	37.0000	0.25	36	37.0022	0.00	2.78	114.00	145
L9	100	82	22	8	45	46	46	46.0000	1.90	45	46.0250	0.00	2.28	404.00	145
L10	100	83	22	8	45	47	46	46.0000	1.86	45	46.0250	0.00	2.28	403.00	145

The columns contain:

- Instance – Name of the instance;
- L and W – Pallet length and width, respectively;
- l and w – Box length and width, respectively;
- Optimal solution – Known optimal solution;
- Area Bound – Area bound given by $\lfloor (L*W)/(l*w) \rfloor$ (where $\lfloor z \rfloor$ denotes rounding down to the nearest integer);
- Linear relaxation – Bound and Time in seconds;
- Barnes bound – Bound provided by Barnes [6];
- Lower bound – Lower bound by VI or block heuristic;
- Upper bound – Upper bound provided by LagClus;
- GAP LB (%) – Percentage deviation from the optimal solution to the best lower bound:

$$Gap\ LB = \frac{(Optimal\ solution - Lower\ bound)}{Optimal\ solution} * 100$$

- GAP UB (%) – Percentage deviation from the optimal/best known solution to the best upper bound:

$$Gap\ UB = \frac{(Upper\ bound - Optimal\ solution)}{Optimal\ solution} * 100$$

- Time – Time in seconds elapsed by LagClas reaching some of the stop conditions;
- Iterations – Number of iterations used by LagClus.

Table 2 – Computational results for 10 instances randomly obtained from COVER II (Dowsland, 1987).

Instance	*L*	*W*	*l*	*w*	Optimal solution	Area bound	Barnes bound	Linear Relaxation		LagClus					
								LP bound	Time (s)	Lower bound	Upper bound	GAP LB (%)	GAP UB (%)	Time (s)	Iterations
L11	57	53	7	5	85	86	86	86.1379	68.09	85	86.3574	0.00	1.60	418.00	145
L12	84	75	11	6	94	95	95	95.1678	72.56	94	95.6230	0.00	1.73	1192.00	145
L13	151	131	19	11	94	94	94	95.1667	209.39	93	94.4966	1.06	0.53	2755.00	145
L14	61	38	6	5	77	77	77	77.0698	22.01	76	77.2440	1.30	0.32	144.00	145
L15	100	53	9	7	83	84	84	83.9025	55.14	83	84.1221	0.00	1.35	365.00	145
L16	120	80	14	11	61	62	62	61.7500	6.03	61	61.9986	0.00	1.64	41.00	115
L17	51	38	11	3	57	58	58	57.7500	5.73	57	58.3649	0.00	2.39	118.00	145
L18	120	83	17	6	97	97	97	97.5000	90.68	97	97.9876	0.00	1.02	273.00	55
L19	131	86	16	7	100	100	100	100.1432	230.52	100	100.9866	0.00	0.99	195.00	35
L20	98	93	17	7	75	76	76	75.0000	89.97	75	75.9902	0.00	1.57	482.00	135

The columns contain:

- Instance – Name of the instance;
- *L* and *W* – Pallet length and width, respectively;
- *l* and *w* – Box length and width, respectively;
- Optimal solution – Known optimal solution;
- Area Bound – Area bound given by $\lfloor (L*W)/(l*w) \rfloor$ (where $\lfloor z \rfloor$ denotes rounding down to the nearest integer);
- Linear relaxation – Bound and Time in seconds;
- Barnes bound – Bound provided by Barnes [6];
- Lower bound – Lower bound by VI or block heuristic;
- Upper bound – Upper bound provided by LagClus;
- GAP LB (%) – Percentage deviation from the optimal solution to the best lower bound:

$$Gap\ LB = \frac{(Optimal\ solution - Lower\ bound)}{Optimal\ solution} * 100$$

- GAP UB (%) – Percentage deviation from the optimal solution to the best upper bound:

$$Gap\ UB = \frac{(Upper\ bound - Optimal\ solution)}{Optimal\ solution} * 100$$

- Time – Time in seconds elapsed by LagClas reaching some of the stop conditions;
- Iterations – Number of iterations used by LagClus.

Table 3 – Computational results for 10 randomly examples obtained from COVER III (Alvarez-Valdez et al. [2]), upon all instances that do not present optimal solution known.

Instance	L	W	l	w	Optimal solution	Area bound	Barnes bound	Linear Relaxation		LagClus					
								LP bound	Time (s)	Lower bound	Upper bound	GAP LB (%)	GAP UB (%)	Time (s)	Iterations
L21	99	88	12	5	144	145	145	145.0000	1370.07	144	145.8324	0.00	1.27	998.00	145
L22	99	75	13	5	113	114	114	114.0000	233.54	113	114.7045	0.00	1.51	342.00	145
L23	97	95	9	7	145	146	146	146.1436	1184.67	145	146.8228	0.00	1.26	620.00	145
L24	98	98	10	7	136	137	137	137.1338	1480.91	136	137.6772	0.00	1.23	468.00	145
L25	98	88	10	7	122	123	123	123.1186	594.73	122	123.6647	0.00	1.36	357.00	145
L26	97	96	11	6	140	141	141	141.0000	891.98	140	141.8625	0.00	1.33	932.00	145
L27	96	87	8	7	148	149	149	149.0000	1232.13	148	149.9519	0.00	1.32	638.00	145
L28	99	70	15	4	114	115	115	115.0000	520.66	114	116.1673	0.00	1.90	348.00	145
L29	91	70	12	5	105	106	106	106.0000	136.84	105	106.5183	0.00	1.45	156.00	145
L30	93	84	11	6	117	118	118	118.0000	248.46	117	118.9239	0.00	1.64	387.00	145

The columns contain:

- Instance – Name of the instance;
- L and W – Pallet length and width, respectively;
- l and w – Box length and width, respectively;
- Best solution – Best feasible solution reported in literature;
- Area Bound – Area bound given by $\lfloor (L*W)/(l*w) \rfloor$ (where $\lfloor z \rfloor$ denotes rounding down to the nearest integer);
- Barnes bound – Bound provided by Barnes [6];
- Linear relaxation – Bound and Time in seconds;
- Lower bound – Lower bound by VI or block heuristic;
- Upper bound – Upper bound provided by LagClus;
- GAP LB (%) – Percentage deviation from the best known solution to the best lower bound:

$$Gap\ LB = \frac{(Best\ solution - Lower\ bound)}{Best\ solution} * 100$$

- GAP UB (%) – Percentage deviation from the best known solution to the best upper bound:

$$Gap\ UB = \frac{(Upper\ bound - Best\ solution)}{Best\ solution} * 100$$

- Time – Time in seconds elapsed by LagClas reaching some of the stop conditions;
- Iterations – Number of iterations used by LagClus.

Table 4 – Computational results for instance L7 varying the number of clusters.

Number of clusters	Iterations	VI heuristic	LagClus	Time (s)
2	67	49	49,9936	257
3	145	49	50,0612	86
4	145	49	50,0429	31
5	145	49	50,1019	33
6	145	49	50,1316	27
7	145	49	50,1360	18
8	145	49	50,1512	18
9	145	49	50,1241	19
10	145	49	50,1229	22
11	145	49	50,1785	21
12	145	49	50,2020	21
13	145	49	50,2199	16
14	145	49	50,1731	18
15	145	49	50,2645	15

The columns contain:

- Number of clusters – Number of clusters used in LagClus;
- VI heuristic - Result obtained by VI heuristic described in Figure 4 or block heuristic;
- LagClus – Upper bound provided by LagClus;
- Iterations – Number of iterations used by LagClus;
- Time – Time in seconds elapsed by LagClas reaching some of the stop conditions.

Table 5 – Computational results using the column generation approach.

Instance	Column generation approach					Solving the last RMP as integer	
	Initial number of columns	**Initial RMP**	**Final number of columns**	**Final RMP**	**Time CG (s)**	**Solution (IP)**	**Time IP (s)**
L1	500	34.00	626	35.00	17	34	0.00
L2	500	41.00	674	43.00	45	42	0.20
L3	500	35.16	697	37.00	62	36	0,00
L4	500	45.00	680	47.00	183	46	2.00
L5	500	47.20	691	49.00	286	48	1.00
L6	500	44.00	829	46.00	194	45	0.00
L7	500	49.00	815	49.85	2166	49	1.00
L8	500	36.00	645	37.00	27	36	0.00
L9	500	44.00	813	46.00	331	45	4.01
L10	500	44.00	813	46.00	326	45	5.10

The columns contain:

- Instance – Instance name;
- Initial number of columns – Initial number of columns considered in RMP;
- Initial RMP – Initial value provided by RMP;
- Final number of columns – Final number of columns considered in RMP;
- Final RMP – Final value provided by RMP;
- Time CG (s) – Time consumed at the column generation process;
- Solution (IP) – Value obtained by RMP solved using integer variables;
- Time IP (s) – Time consumed for solving the integer RMP.

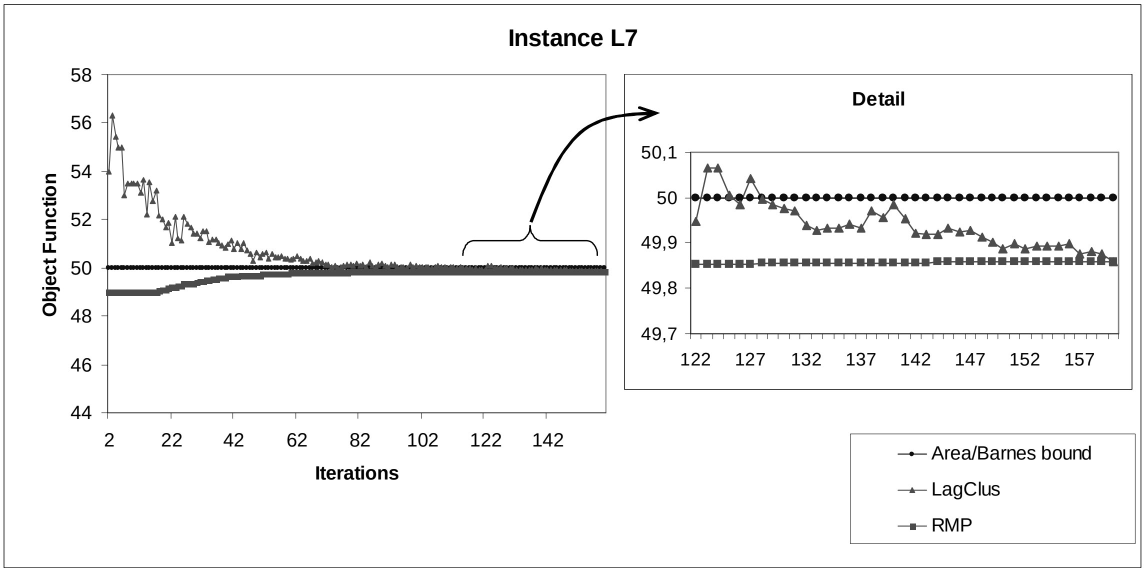


Figure 6 – LagClus and RMP behavior.