SPACE GROWTH USING THE BIN-PACKING METHOD

RAKOTONDRASOLO Andrianandraina Marijaona[1], RAKOTOMIRAHO Soloniaina[2], RANDRIAMAROSON Rivo Mahandrisoa[3]

[1]*PhD student, SE-I-MSDE, ED-STII, Antananarivo, Madagascar*
[2]*Thesis director and Laboratory Manager, SE-I-MSDE, ED-STII, Antananarivo, Madagascar*
[3]*Thesis co-director, SE-I-MSDE, ED-STII, Antananarivo, Madagascar*

ABSTRACT

This paper provides methods of reengineering articles in one or more boxes called "bins". The reengineering problem raised from operations research and combinatorial optimization. It is about finding the most economical storage possible for a set of objects. Thus, bin-packing type problems consists in placing articles characterized by their shape in one or more boxes. The variants are distinguished by size, prior knowledge of the items, the shape of the items and boxes, the possibility of modifying the orientation of the items. The re-registration can be done in several dimensions, but in this paper, the problem of bin-packing in two dimensions will be the object of the study, algorithms of re-registration will be proposed. In this article, two different methods will be detailed including approximate Methods and exact methods. The Shelf-type algorithm is chosen because it offers an opportunity to classify each bin into a tier, so the approximate methods is more suitable.

Keyword: *bins, bin-packing, 1BP, 2BP, unidirectional, bi-directional, shelf, heuristic, algorithm*

1. INTRODUCTION

Given a set of rectangular shaped objects of any known dimensions and given a larger rectangular shaped bin of known dimensions, the two-dimensional bin-packing (2BP) problem is to determine the minimum number of bins needed to store all of these objects without overlapping (the objects do not overlap the bins and do not overlap). The objects are arranged in such a way that their edges are parallel to those of the bins that contain them (we are talking about orthogonal arrangement). This problem has many industrial applications. It is mainly found in the fabric, steel, wood, or glass industry in the form of a cutting problem. There are also other applications such as scheduling tasks with resource constraints or optimizing the placement of newspaper advertisements. Each real application has its own specificities and practical constraints. A particular extension of this problem is the ability to rotate objects 90 degrees.

2BP is a generalization of the one-dimensional bin-packing (1BP) problem that is known to be an NP-hard problem. The same goes for 2BP. The bibliography of the 2BP problem is rich. Indeed, this problem has been of great interest during the last ten years. Most of the work focuses on the problem with fixed orientation of objects.

2. CLASSIFICATION AND PRACTICAL CONSTRAINTS

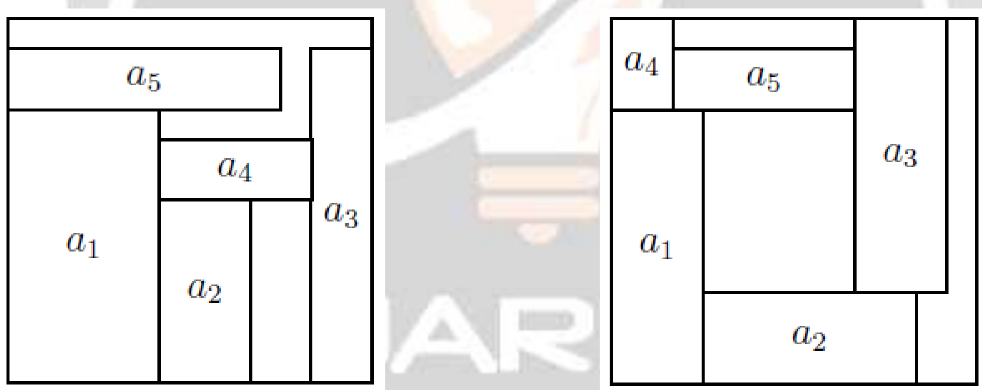
The classification and the constraints vary according to the dimension, but it is the case of two dimensional bin-packing that interests us the most in this study, i.e. the weight of an object does not part of the parameters to be studied. Many practical problems model themselves as a bin-packing problem. However, each real problem has its own specificities such as:

- the specific feature of the object :
  - objects of homogeneous or non-homogeneous shapes ;

  - objects of uniform or different sizes ;
  - deformable or non-deformable objects.
- the specifics of the problem:
  - number of dimensions of the problem ;
  - have only one bin (decision problem or maximization problem) ;
  - seek to minimize the number of bins to use ;
  - seek to minimize the surface area or the overall volume of the objects to be placed.
- the constraints specific to the problem:
  - balance constraints between objects ;
  - constraints on the order in which objects should be removed from the bin ;
  - orientation constraints of an object ;
  - constraints of placing objects in an axis following the shape of the bin to avoid overlap.

Dyckhoff and Finke [1] have proposed a typology which makes it possible to organize the problems of cuts and placements by taking into account four main characteristics:

- the number of dimensions of the problem ;
- the type of task: all objects and a selection of bins, or a selection of objects and all bins ;
- the characteristics of the bins: 1 single bin, bins of identical sizes, or bins of different sizes ;
- the characteristics of the objects: identical objects, few objects of different shapes, several objects of different shapes or even objects of relatively identical shapes.



(a) Guillotine constraint respected. (b) Guillotine constraint not respected.

Fig -01: Illustration of the guillotine constraint.

A more recent typology has been proposed by Wscher et al. [2] in order to include recent cutting and storage issues and establish a complete categorization of all known issues in the field.

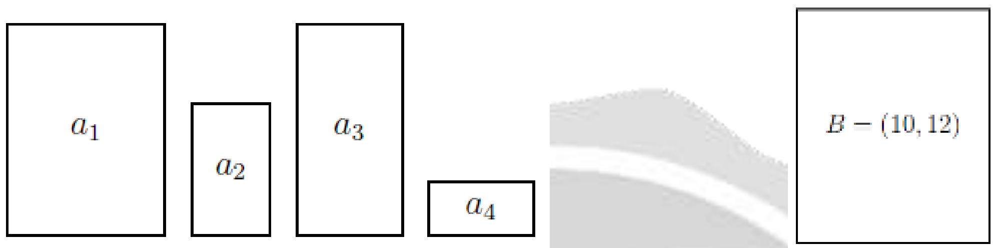
Looking at the problem of bin-packing in two dimensions (2BP), the two most encountered specificities are as follows:

- *Orientation:* the objects can be fixed orientation (It is about oriented case) or they can be rotated by 90 degrees (the non-oriented case). It is this specificity that interests us in particular. During this section we develop several works carried out around the 2BP problem, the oriented as well as the undirected case.

- *The guillotine* constraint: If it is imposed, we must have the possibility of restoring the objects stored by end-to-end cuts parallel to the dimensions of the bins. Figure 01 presents two examples of solutions: the solution illustrated in Figure 01 (a) respects the guillotine constraint, which is not the case of the solution illustrated in Figure 01 (b).

In [3] a classification of 2BP problems taking into account the two specificities mentioned above was proposed. A two-dimensional problem is then denoted by 2BP C1 (C1) C2. The C1 field takes the value R to indicate the non-oriented case and the value O for the oriented case. While field C2 indicates the presence or absence of the guillotine constraint (it thus takes the value G and F respectively).

The guillotine constraint is the subject of several recent studies. Throughout this document we do not consider the guillotine constraint, we will omit the field C2 and we will talk about 2BP O (respectively 2BP R) to denote the 2BP problem in the oriented case (respectively undirected).



(a) All objects (b) The bin B = (W, H)

Fig -02: An example of a 2BP instance.

3. MODELS FOR THE 2BP

This section describes two sample models for 2BP. A first model dedicated to 2BP was proposed by Gilmore and Gomory [4] as an extension of their 1BP approach [5] [6]. Their model is based on the enumeration of all the subsets of objects that can be stored in the same bin. Each of these sets is represented by a binary column vector V_j composed of n elements $v_{ij}, i = 1, \dots, n$ such as:

$$v_{ij} = \begin{cases} 1 \text{ if ai belongs to the subset j} \\ 0 \text{ if not} \end{cases} \quad (01)$$

Let M be the matrix composed of vectors $V_j(j = 1, \dots, m)$, m being the number of subsets that can be placed in the same bin. The matrix M therefore represents the set of all the possible storage configurations. The model is then written:

$$Minimize \sum_{j=1}^{m} x_j \quad (02)$$

By respecting the following constraints:

$$\sum_{j=1}^{m} v_{ij} x_j = 1 \quad (i = 1, \dots, n) \quad (03)$$

$$x_j \in \{0, 1\} \quad (j = 1, \dots, m) \quad (04)$$

The major drawback of this model is the exhaustive number of columns to be generated. Gilmore and Gomory [5] [6] presented for 1BP, a method to dynamically generate as needed columns. It consists of solving a one-dimensional backpack problem. Fekete and Schepers [7] proposed a model based on graph theory for the following decision problem: is it possible to place a given set of objects in a single bin?

The advantage of the proposed model is that it avoids a large number of symmetrical configurations. Two graphs $G_W = (V, E_W)$ and $G_H = (V, E_H)$, each of size n, are considered, n being the number of objects. Each vertex v_i of G_W and G_H is associated with an object a_i stored in the bin. The two graphs are constructed as follows: an edge is added in G_W (respectively in G_H) between the two vertices v_i and v_j if and only if the projections of the objects a_i and a_j on the horizontal axis (respectively vertical) overlap. Fekete and Schepers [7] show that a solution associated with a pair of interval graphs (G_W, G_H) is feasible if the following three conditions are true:

1. For each stable set S_W of G_W, we have $\sum_{v_i \in S_W} w_i \le W$
2. For each stable set S_H of G_H, we have $\sum_{v_i \in S_h} h_i \le H$
3. $E_w \cap E_h = \emptyset$

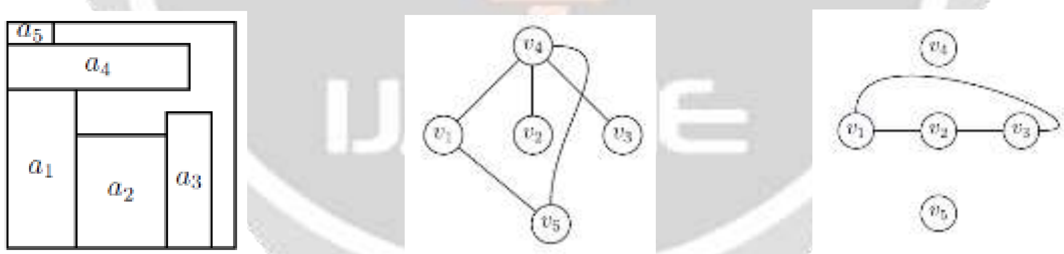
The first two conditions mean that all the objects fit in the bin, while the third condition means that there is no overlap between any two objects stored in the bin.

Other models have been proposed for the bin-packing problem and its variants. Beasley [8] proposed a linear model for the characterization of 2D slicing problems that associate a profit with each object, the objective being to store a subset of objects in a bin so as to maximize the sum of the profits of the stowed objects (backpack problem). Hadjiconstantinou and Christofides [9] worked on a similar model. A modeling by linear programming of the two-dimensional guillotine strip packing problem was recently proposed in the thesis of Ben Messaoud [10].

In the next section we describe different preprocessings for the bin-packing problem.

4. PRE-TREATMENTS

In difficult problems like the case of the bin-packing problem, it is often interesting to apply preprocessing on the instance to study before launching a costly resolution method in terms of computation time. The preprocessing can also contribute to the improvement of the lower bounds by valuing the spaces which will be lost independently of the method of solving the problem. To our knowledge, there are no pre-treatments dedicated to 2BP A. In this section, we describe the pre-treatments proposed in the literature for 2BP O.



(a) Example solution (b) Graph.G_W (c) Graph G_H

Fig -03: Example of application of the model of Fekete and Scheppers.

Among the existing pre-treatment models, two of these models will be explained in the following paragraphs.

4.1 Martello and Vigo pretreatment

Martello and Vigo [11] proposed in 1998 a pretreatment that consists in seeking to determine optimal assemblies. Let's a_i a given object of the considered instance, we try to find the set of objects to be placed with such that the storage of the bin containing a_i is optimal. Let's Q_i the set of objects compatible with a_i:

$$Q_i = a_i \in A: \left(h_i + h_j\right) \le H \ ou \ \left(w + w_j\right) \le W \qquad (05)$$

There are two cases, in the first case $Q_i = \emptyset$, this means that no object of $A \setminus a_i$ can be stored with a_i. Optimal storage of the object a_i is trivial, it must be stored alone in a bin. In the second case, $Q_i \neq \emptyset$, an upper bound on the number of objects that can be arranged with a_i is then calculated as follows. The objects of Q_i are considered in decreasing order of surfaces. The maximum value such that the sum of the last the objects of Q_i does not exceed the value $HV - h_i v_i$, is an upper bound on the number of objects that can be arranged with a_i. Either a_{j*} the first (largest) object in q_i, a preprocessing is applied in the following two cases:

- l = 1 and for all a_j in $Q_i \setminus a_{j*}, h_{j*} \geq h_j \; et \; w_{j*} \geq w_j$
- l = 2 and any pair of objects $Q_i \setminus a_{j*}$ that can be placed with a_i, can be placed in a bin of dimensions equal to the dimensions of a_{j*}

In either case, the best choice is to store $a_{j*}$with a_i.

4.2 Boschetti and Mingozzi pre-treatment

Boschetti and Mingozzi [12] proposed in 2003 a preprocessing that consists in updating the dimensions of the objects taking into account the surface that will be lost in the bin where they will be affected. Given an object a_i, it is a question of calculating the width w_i^* necessarily lost when a_i is placed in a bin. This value can be calculated by solving a variant of the subset-sum problem:

$$w_j^* = w_j + \max\left\{ \sum_{a_i \in A - \{a_j\}} w_i \xi_i : \sum_{a_i \in A - \{a_j\}} w_i \xi_i \leq W - w_j, \quad \xi_i \in \{0,1\} \right\} \tag{06}$$

w_{j*} can be calculated in O (nW) by means of a classical pseudo-polynomial method.
The width w_j of the object a_j can then be updated as follows:

$$w_j \leftarrow w_j + (W - w_j^*) \tag{07}$$

The preprocessing is also applied, in a similar fashion, to the height h_i of the object a_i. This preprocessing is applied to all objects in the processed instance, but the order in which the objects are considered is important. Indeed, if the width of an object a_j comes into play in the calculation of the optimal value w_i^* of an object a_i, w_j^*, will have as value W and w_j cannot be updated in turn. A heuristic rule that consists in considering the objects in ordercreasing of height (respectively width) before applying the preprocessing on the heights (respectively widths) of the objects is applied in order to maximize the number of preprocessed objects.

5. METHODS OF SOLVING THE BIN-PACKING PROBLEM

There are many methods of solving a bin-packing problem. There are two main categories of combinatorial optimization problem solving methods: exact methods and approximate methods. The exact methods get the optimal solution every time, but the computation time can be long if the problem is complicated to solve. Approximate methods, also called heuristics, make it possible to quickly obtain an approximate solution, and therefore not necessarily optimal. We will detail an example of the algorithm for solving each category.

5.1 Approximate Methods

In difficult problems, like the case of the bin-packing problem, it is often interesting to apply heuristics that give good quality solutions in a reasonable time. In this paper, the resolution of the 2BP is considered. The two-dimensional bin-packing problem is a classic problem for which several approximate methods have been proposed, these methods are in general heuristics, most of which were inspired by the strategies used in 1BP. These heuristics can be divided into two main families [13], including one-phase algorithms and two-phase algorithms. Two-phase algorithms will be detailed in the following paragraphs.

Two-phase algorithms first start by arranging the objects in a bin of infinite height while looking for a solution for the strip-packing (2SP) problem. In the second phase, an algorithm of resolution for the problem of 1BP consists in using the solution of the 2SP obtained to build the arrangement in the bins to be used effectively (solution for the 2BP).

The two-phase methods work as follows: the first step (strip-packing) consists in sorting the objects according to their decreasing heights and placing them successively in a container of infinite height, filling them layer by layer. The first object, the longest, is placed in the first level which corresponds to the lower edge of the bin.

The other objects are arranged according to an NF (Next-Fit), FF (First-Fit) or BF (Best-Fit) strategy with respect to the width of the bin, we are talking respectively about an NFDH (Next-Fit Decreasing Height), FFDH (First-Fit Decreasing Height) and BFDH (Best-Fit Decreasing Height) strategy.

- NFDH (Next-Fit Decreasing Height) algorithm

In the NFDH algorithm, rectangles are sorted according to their height in a decreasing fashion. The first rectangle (the highest) is placed at the bottom left, which determines the height of the floor (definitively fixed). Then another rectangle is placed next to this one (which is the highest rectangle among the rest), and we continue until the current rectangle no longer fits by its width. In this case, this rectangle is placed above to the left of the shelf, creating a new floor [14]. This operation is repeated until no longer having the rectangle. (Fig-04)

- FFDH (First-Fit Decreasing Height) algorithm

The principle of FFDH is very similar to that of NFDH. However, before placing the next rectangle on the current level, it is necessary to check the possibility of placing it in the lower level (as low as possible), having enough space to accommodate it. In FFDH, it is therefore possible to revisit a lower level, which is not allowed in NFDH. (Fig-04)

- BFDH (Best-Fit Decreasing Height) algorithm

The BFDH algorithm acts the same as DDFH, but the search for a location of the current rectangle is done exhaustively on the shelf. [15]

Thus, the horizontal space lost by the placement of a rectangle is calculated for all the shelves. The rectangle is effectively placed in the shelf for which this wasted space is minimal. This algorithm is therefore logically slower than FFDH which places the rectangles in the first shelf found. (Fig-04)

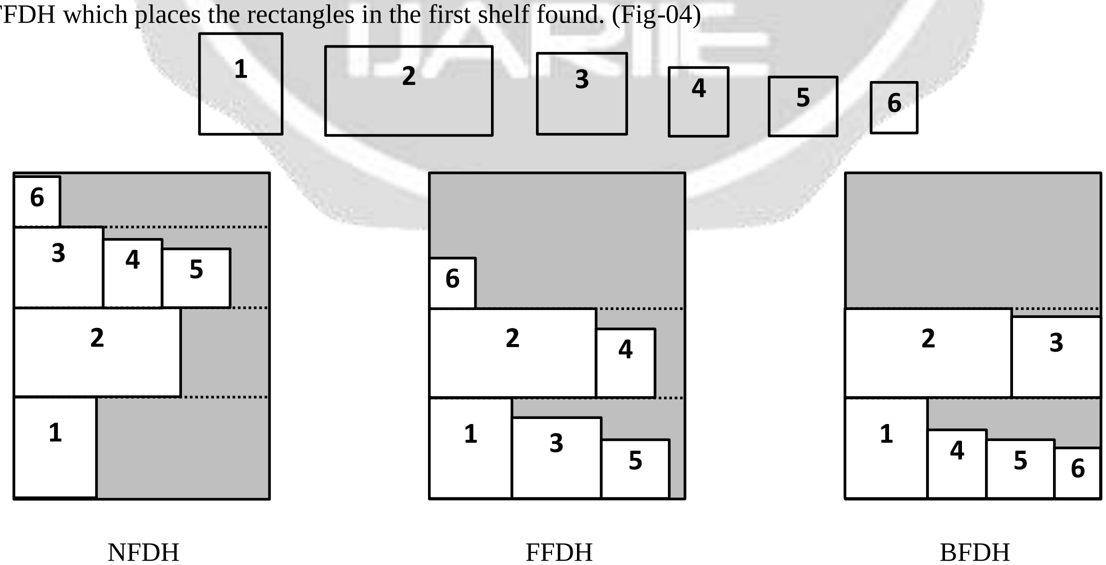


Fig -04: Illustration of examples for NFDH, FFDH and BFDH.

5.2 Exact methods

The principle of exact methods is to find the best solution in the set of solutions to a problem. Exact optimization concerns all methods of obtaining a result that is known to be optimal for a specific problem. We can classify the exact methods in three main classes: dynamic programming, linear and quadratic programming and branch and bound search methods.

In this article, the branch and bound method will be detailed. The branch and bound method (procedure by evaluation and progressive separation) consists in enumerating these solutions in an intelligent way in the sense that, by using certain properties of the problem in question, this technique manages to eliminate partial solutions which do not lead to the solution we are looking for. As a result, it is often possible to obtain the desired solution in a reasonable time. Of course, in the worst case, we always fall back on the explicit elimination of all solutions to the problem. To do this, this method has a function that allows you to put a limit on certain solutions to either exclude them or maintain them as potential solutions. Of course, the performance of a branch and bound method depends, among other things, on the quality of this function (its ability to exclude partial solutions early).

Several enumeration algorithms were derived from the bottom-left strategy proposed by Baker, Coffman, and Rivest [16] which produce an approximate solution by placing one item at a time, in the lowest possible feasible position, left justified. Hadjiconstantinou and Christofides [9] developed a tree-search exact algorithm for the 2D-KP that packs the next item in every possible position such that the item's left and bottom edges touch either the bin or edges of other items (left-most downward strategy). Martello and Vigo [11] adopted this strategy in a branch-and-bound algorithm for the 2D-OPP, and embedded it in a two-level enumeration algorithm for the 2D-BPP.This strategy later evolved into two classes of implicit enumeration schemes, namely the *staircase placement* and the *niche placement*. Both strategies enumerate the possible packing of the items with their bottom-left corner in particular positions (corner points), induced by the current (partial) packing, whose definition depends on the specific strategy.

(a) (b)

Fig- 05: Corner points of (a) staircase placement; (b) niche placement.

5.3 Comparison of approximate method and exact method

Combinatorial problems are distinguished by the finite number of solutions. The exact methods are used for solving Cutting and Packing (C&P) problems with components of rectangular or parallelepiped geometry. For large instances of these problems, approximate resolution methods are used. These methods are mainly metaheuristics such as evolutionary algorithms or even dedicated methods specific to each problem.

The exact methods then allow to obtain the optimal solution every time, but the computation time can be long if the problem is complicated to solve.

The approximate methods, also called heuristics, make it possible to obtain an approximate solution, therefore not necessarily optimal.

6. CONCLUSION

In this article we described some of the space layout work. We note that there is little work on the non-oriented case, whether in resolution methods or default assessments. However, the ability to turn objects is a specificity frequently encountered in real problems, which therefore deserves to be studied. This bin-packing method is applicable in the field of parking space design by considering a polygonal-shaped space. In a parking lot, a maneuverability envelope of a vehicle is represented by a bin. To properly arrange the given space, it is necessary to manage the overlap of the envelopes. So, this article highlights how the bin-packing method manages parking space management. Among these methods, the shelf algorithm is the most suitable.

7. REFERENCES

[1]. U. Dyckhoff, H.and Finke, “Cutting and Packing in Production and Distribution: Typology and Bibliography.”, (eds. Mueller, W.A. and Schuster, P.), Springer-Verlag, New York, 1992.

[2]. G. Wäscher, H. Haussner, H. Schumann, “An improved typology of cutting and packing problems.”, European Journal of Operational Research, to appear, 2006.

[3]. A. Lodi, S. Martello, D. Vigo, “Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems.”, INFORMS J. Comput., 11 :345–357, 1999.

[4]. P. Gilmore, R. Gomory, « Multistage cutting stock problems of two and more dimensions. », Ops. Res., 13 :94–120, 1965.

[5]. P. Gilmore, R. Gomory, « A linear programming approach to the cutting stock problem. », Ops. Res., 9 :849–859, 1961

[6]. P. Gilmore, R. Gomory, « A linear programming approach to the cutting stock problem – part II. », Ops. Res., 11 :863–888, 1963.

[7]. S. Fekete, J. Schepers, « A combinatorial characterization of higher-dimensional orthogonal packing. », Mathematics of Operations Research, 29 :353–368, 2004.

[8]. J. E. Beasley, “An exact two-dimensional non-guillotine cutting tree search procedure.”, Operations Research, 33 :49–64, 1985.

[9]. E. Hadjiconstantinou, N. Christophides, « An exact algorithm for general, orthogonal, two-dimensional knapsack problem. », European Journal of Operational Research, 83 :39–56, 1995.

[10]. S. Ben Messaoud, « Caractérisation, Modélisation et Algorithmes pour des Problèmes de Découpe Guillotine. », PhD thesis, Université de Technologie de Troyes, France, 2005.

[11]. S. Martello, D. Vigo, “Exact solution of the two-dimensional finite bin packing problem.”, Management Science., 44 :388–399, 1998.

[12]. M. A. Boschetti, A. Mingozzi, « The two-dimensional finite bin packing problem. Part I: New lower bounds for the oriented case. » 4OR, 1 :27–42, 2003.

[13]. A. Lodi, S. Martello, D. Vigo, « Recent advances on two-dimensional bin packing problems », Discrete Applied Mathematics, 123 :379–396, 2002.

[14]. E. E. Bischoff, G. Wascher, « Cutting and Packing », The European Journal of Operational Research, 84, 503-505, 1982.

[15]. A. Lodi, S. Martello, D. Vigo, « Neighborhood search algorithm for the guillotine non-oriented two-dimensional bin packing problem. », kluwer Academic Publishers, Boston :125–139, 1998.

[16]. B.S. Baker, E.G. Coffman, Jr., and R.L. Rivest. « Orthogonal packing in two dimensions. » SIAM Journal on Computing, 9(4):846{855, 1980.