Information Sciences and Computer Engineering, Vol. 1, No. 2, (2010) 21–25

PHARUS

International Journal of
Information Sciences and Computer Engineering

journal homepage: http://www.ijisce.org

Improved Algorithm for Orthogonal Rectangular Packing Problem

Rachid Ouafi*, Isma Dahmani

Faculty of Mathematics, Houari Boumedienne University of Sciences and Technology (USTHB), BP 32 El Alia 16111, Algiers, Algeria

Abstract– **In this paper, we develop a modified version of the Best First Branch and Bound algorithm (BFBB) proposed in [5] for solving exactly the Orthogonal Rectangular Packing problem (ORP). The ORP consists to pack a given set of small rectangles in an enclosing final rectangle. In our proposed version, we introduce a new upperbound in order to reduce the problem space search. We also propose new strategies that eliminate several duplicate packing patterns. Extensive computational testing onseveral randomly generated problem instances shows the effectiveness of the proposed algorithm.**

Keyword: Branch and Bound, Combinatorial Optimization, Heuristics, Rectangular Packing.

1. Introduction

Cutting and Packing problems belong to an old and very well-known family, called CP in Dyckhoff [1], Wäscher et al. [2]. The CPbelongs to the combinatorial optimization problems. It is mainly based on the geometrical aspect, which makes increase its complexity by classifying it among the class NP-Hard problems [3]. The CP involves many industrial applications [4] from computer science, industrial engineering, logistics, manufacturing, production process, etc.

In this paper, we study one of the most interesting problems of Cutting and packing, the Orthogonal Rectangular Packing Problem (shortly ORP) [5]. The ORP consists in joining a given set of n small rectangular pieces, into an enclosing final rectangle. Each type of piece $i, i = 1, \dots, n$, is characterized by a length l_i and a width w_i Moreover, a demand constraint b_i is imposed on each type of piece i, such as a piece i must be appeared exactly b_i times in the solution. The aim is then to minimize the area of the enclosing final rectangle. There are various options on packing rule, in our case; the orientation of each piece is fixed i.e., a piece of length l and width w is different from a piece of length w and width l (when l≠ w). The ORP can be either weighted or unweighted, we consider only the unweighted case in which the profit of a piece is equal to its area ($c_i = l_i w_i, i = 1, \dots, n$).

*Corresponding author:
Email address: rachid_ouafi@hotmail.com, Ph: +21 3770432863

To solve this problem, many algorithms based on different strategies various have been proposed. These algorithms can be categorized into two categories: the heuristic algorithms and meta-heuristic algorithms. The aspect of heuristic algorithms is to determine the packing rules. Liu et al. presented an improved heuristic algorithm based on the bottom-left method [6] The less flexibility first principle [7] was introduced by Wu and al. Wei et al [8] suggested a least first strategy which evaluates the positions used by the rectangles. The meta-heuristic algorithms use meta strategies such as genetic algorithm, neural networks, tabu search list and simulated annealing in order to guide the process search [9].

This paper is organized as follows: Section 2 describes the Orthogonal Rectangular Packing problem and the principle of the exact method Best First Branch and Bound [10]. In section 3, we give an improved version of BFBB based on the development of a new upper bound (greedy heuristic), we also adapt the new representation of the closed list and introduce the dominated and duplicated models strategies. Finally, the performance of our algorithm is presented in Section 4. A set of problem instances is considered and benchmark results are given. Conclusions are drawn in Section 5.

2. Improved algorithm for the BFBB

2.1. Orthogonal Rectangular Packing (ORP)

2.1.1. Presentation of the Orthogonal Rectangular Packing problem

Given a set of n small rectangular pieces $S = \{(l_1, w_1), (l_2, w_2), \dots, (l_n, w_n)\}$, each piece i ($i = 1, .., n$), is represented by a lengthl_i and a widthw_i. Moreover, each type of piece i is associated with a demand constraint bi, i.e. the piece i must be included exactly b_i times in the solution.

The set of all feasible solutions of *ORP* problem is denoted as $T = \{T_1, T_2, \dots, T_m\}$ consisting of m enclosing final rectangles, such as each solution contains all the pieces with their copies. A feasible solution $T^* \in T$ is said to be optimal if it realizes the minimum wastage.

We use a strategy of orthogonal build [11] in order to reduce the feasible patterns of packing. It is applied by combining the pieces and their copies by horizontal and vertical builds. For this purpose, we adopt the following definitions.

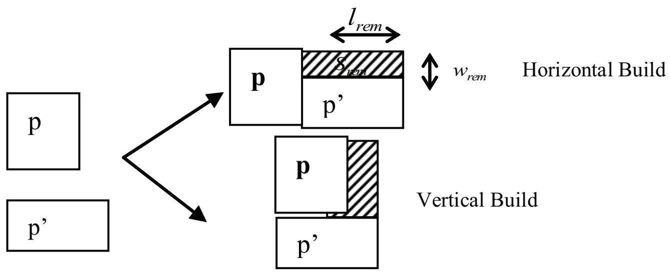

Fig. 1. Horizontal and vertical builds.

Definition 1. Let p and p' be two pieces of S, with dimensions (l, w) and (l', w') respectively. We say that $R = (l+l', \max\{w, w'\})$ (resp. $R = (\max\{l, l'\}, w + w')$), is a rectangular packing (denoted r-packing) obtained by joining the pieces p and p' by a horizontal build (resp. vertical). Moreover, the number of occurrences of each type of pieces in each build does not exceed the upper bound b_i.,$i = 1, \dots, n$ (figure 1)

S_{rem}: Remaining area (empty space) after the combination, of lengthl_{rem} and width w_{rem}.

Definition 2. R is called a terminal packing (denoted t-packing), if it contains all pieces with their copies.

2.1.2. Best First Branch and Bound algorithm (BFBB)

1. Upper bound

We propose an initial procedure (greedy) to produce a primary upper bound for the exact algorithm. The suggested heuristics begin with two constructed obvious t-packing:

– The first one is a horizontal t-*packing* satisfying the following value:

$$E_h = (\sum b_i l_i)(\max_{1<i<n}\{w_i\}) \quad (1)$$

– The second one is a vertical t-packing with the value:

$$E_v = (\sum b_i w_i)(\max_{1<i<n}\{l_i\}) \quad (2)$$

The initial evaluation of the greedy heuristic consists in choosing one of the two previoust-packing realizing the minimal value:

$$E_{\text{sup}} = \min\{E_h, E_v\} \quad (3)$$

The value E_{sup} is considered as a first upper bound. It is improved by a greedy procedure 'Algo1' using a horizontal build. The same process is used to produce another upper bound with vertical build. Thus, the initial E_{sup} of the exact algorithm is the minimum value of the two previous upper bounds obtained by the greedy procedure.

Algo1— *The greedy procedure for the PAOR problem*

Input: A set of rectangular pieces bounded by b_i, $1 \leqslant i \leqslant n$
Output: A suboptimal solution E_{sup}

Initial step:

1) Compute E_{sup} using (3)
2) Set ξ be the set of all pieces with their copies and R be a piece of ξ with the highest width $w_R = max_{R' \in \xi}\{w_{R'}\}$
3) Construct $Init = (R_{Init}, l_{rem}, w_{rem})$ which represents the list composed by *r-packing*. $R_{Init} = (b_R l_R, w_R)$, l_{rem} and w_{rem} are respectively, the length and the width of the sub rectangle represented by S_{rem}.

Main step:
Repeat
Choose the piece $R' \in \xi$ with the longest length $l_{R'} = max_{R' \in \xi}$ $\{l_R\}$ and form a horizontal build with R_{Init}.

- Set $\xi = \xi \backslash \{R'\}$
- Compute E_{Init} the value of the surface of R_{Init} and $S_{rem} = (l_{rem}, w_{rem})$ with $l_{rem} = l_{R'}$and $w_{rem} = w_{Init} - -w_{R'}$.
- Fill up the rectangle (l_{rem}, w_{rem}) with the pieces of the current set ξ using a greedy procedure which consists in putting randomly some pieces without overlapping and update the set ξ.

Until: $(\xi = \emptyset)$ or$(E_{sup} \leqslant E_{Init})$
Terminal step:
If $E_{Init} < E_{sup}$, then the set $E_{sup} = E_{Init}$
End

2. Lower bound

Given an instance of the ORP problem and an r-packing R [12], we expect to find a lower bound for the value of the best$t$-packing including R.

Let be:

$g(R)$: The internal value of R equal to the total area of pieces in R.
$c(R)$: The non-used area in R, which is the difference between total surface of the rectangle R and the total area of the pieces contributing to the construction of R.
$h(R)$: The smallest area which is the surface covering the rest of the demand constraints without tacking in account the set of pieces contained in R.
$f(R) = G(R) + h(R)$ is the minimal value of a feasible t-packing containing R, where $G(R) = g(R) + c(R)$
$h'(R)$: The estimation of $h(R)$, smallest area representing the surface which covers the rest of the demand constraints.

In general, it is not easy to find such estimation. Our estimation $h'(R)$ is computed as follows:

$$h'(R) = \sum_{i=1}^{n} c_i(b_i - x_i) \quad (4)$$

Where x_i is the number of occurrences of the i^{th} piece in R and $c_i = l_i w_i$ is the area of the ith piece.

Where, $f'(R) = G(R) + h'(R)$ representing the lower bound of t-packing containing R.

2.1.3. Exact algorithm BFBB for the ORP problem

Algo2 describes the main step of the exact algorithm to solve *ORP* problem. The BFBB Algorithm uses two lists ξ_1 and ξ_2: The initial list ξ_1 contains n elements such that each element

R_i $(i = 1, \ldots, n)$, of dimensions $(l_R, w_R) = (l_i, w_i)$, has an internal value $g(R_i) = l_i w_i$, an estimated value $h'(R_i)$, a lower bound $f(R_i)$ and a vector d with $d_k \leq b_k (k = 1, \ldots, n)$ which is the number of occurrences of the k^{th} piece in R_i. The list ξ_2 is initialized by the empty set.

At each iteration, we select an element R of the set ξ_1, having the smallest lower bound f' then place it in ξ_2. A set ξ_3 of the new r-packing is created by combining the element R with elements R' of ξ_2 using horizontal and vertical builds. The elements of ξ_3 satisfy the following conditions: $d_k \leq b_k, k = 1, \ldots, n$ and $f'(R) <$ Opt (where Opt is the best current solution value). It is created by combining the element R with all the elements R' of ξ_2

The algorithm stops when the value $f'(R)$ of the element R selected from ξ_1 is greater than or equal to the best current solution value or when the list ξ_1 is reduced to the empty set.

Algo2— *Exact algorithm BFBB for the ORP*

ξ_1 :The set of subproblems;
ξ_2 :The list of stored best subproblems;
R and R': r-packing;
$f'(R)$: The lower bound of the subproblem containing R;
Opt: The best current solution value.

Input: An instance of the orthogonal rectangular packing problem
Output: An optimal solution value *Opt*

Initial step:
$\xi_1 = \{R_1, \ldots, R_n\}$; $\xi_2 \neq \emptyset$ and *finished=false* ;
Let E_{sup} be the upper bound obtained by applying the greedy procedure presented by Algo1 and $Opt = E_{\text{sup}}$
Main step:
Repeat
Choose the *r-packing* R with the smallest f' value; (denoted f'_{min})
If $Opt - f'_{\text{min}} \leqslant 0$ then *finished=true*
Else Begin
Transfer R from ξ_1 to ξ_2 and construct the elements of ξ_3 such that:

- ξ_3 is the set of orthogonal builds between R and all elements of ξ_2;
- Each element of ξ_3 satisfies the constraints $b_i (1 \leqslant i \leqslant n)$ and $f' <$ Opt;

If $\exists$ a terminal packing $R' \in \xi_3 \backslash f'(R') < Opt$, then Opt $= f'(R')$; update the set ξ_1 by $\xi_1 \cup \xi_3$
Replace the set ξ_1 by ξ_1/{*non-terminal packings with the evaluation* $f' \geqslant Opt$ };
If $\xi_1 = \emptyset$, then *finished=true*; End if
Until *finished=true*;
End

2.2. *The new version of the algorithm BFBB*

In our modified version of the algorithm, we start by determining a new upper bound (initial feasible solution) in order to speed up the algorithm by pruning unsearched areas which cannot yield better results than already found. We also, introduce a new representation of the closed list which permits to reduce problem space search without affecting the quality of the obtained solution. Finally, we present the dominated and duplicated patterns strategies in order to discard some symmetrical pattern packing

2.2.1. *The new upper bound*

We expand a new greedy procedure (Algo 3) in order to produce an initial upper bound for the orthogonal rectangular packing problem. The algorithm begins by the computation of the initial upper bound (using equation (3)) which will be improved thereafter. We consider, a list ξ containing a set of pieces with their copies sorted in the decreasing order of width. We state by build a rectangular R_{rec} by the pieces with the highest width. In the main step of the procedure, we realize a successive horizontal builds in the following way: We introduce an intermediate list ξ' composed by pieces of the highest width, then we select the longest piece R of the listξ' and pack it on the right of R_{rec} .We carry out thereafter, a filling with the remaining pieces on the sub rectangle (*empty space)* EV. The procedure stops when $E_{sup} \leqslant E_{rec}$ where E_{rec} is the surface of the resulting rectangle R_{rec} , or when the list ξ is reduced to the empty set. If $E_{sup} \leqslant E_{rec}$ we assign the value of E_{rec} to E_{sup}, E_{sup} is the best upper bound obtained by a horizontal build.

Algo.3—*A greedy procedure for the ORP*

Input: A set of the rectangular pieces bounded by b_i, $1 \leqslant i \leqslant n$
Output: Suboptimal solution denoted E_{sup}

Initial step:

1) Compute E_{sup} such that $E_{sup} = min\{E_h, E_v\}$ where $E_h = (\sum_{i=1}^{n} b_i l_i)(\max_{1 \leqslant i \leqslant n} \{w_i\})$ and $E_v = (\sum_{i=1}^{n} b_i w_i)(\max_{1 \leqslant i \leqslant n} \{l_i\})$
2) ξ: Set of the pieces with their copies sorted according to the decreasing order of width: $w_1 > w_2 \geqslant \ldots \geqslant w_{\sum_{i=1}^{n} b_i}$
3) Let R_{rec} be the initial rectangle obtained by a horizontal build of the piece R_1 with its copies: $R_{rec} = (b_1 l_1, w_1)$, with: $\xi = \xi \quad \backslash \{R_1 / lR_k = / lR_1 et\, wR_k = wR_1\}$

Main step:
Repeat

1. Let ξ' be the set of the pieces with highest width:
2. $\xi' = \{R_k \ : wR_k = maxR_j \in \xi\{wR_k\}, \forall\, lR_k\}$
3. Choose the piece $R' \in \xi' / l_{R'} = maxR_k \in \xi'\{lR_k\}$, construct the horizontal build of R' with R_{rec} and set: $\xi = \xi \backslash \{R'\}$

Compute the empty space EV, result of the previous construction, such that $EV = (l_{R'}, wR_{rec} - w_{R'})$
while *(EV :*$(l_{EV}, w_{EV}) \neq (0,0)$*))* and (if $\exists$ a piece which re-enters in *EV)*
begin
Place the piece R_i in the space *EV*
Set $\xi = \xi \backslash \{R_i\}$
Compute the new empty space *EV*
end;
Until: $(\xi = \emptyset)$ or $(E_{sup} \leqslant E_{rec})$;
Final step: if$E_{rec} < E_{sup}$ then $E_{sup} = E_{rec}$.
End.

2.2.2. New representation of the closed list

The algorithm looks for the smallest surface which contains all the pieces and their copies. It is based on two mains lists closed and open ones. These lists occupy a very important place memory during implementation. The new representation of the closed list [3] allows reorganizing the closed list in an intelligent way to avoid any useless computation.

Indeed, we fix L and W according to the feasible initial solution of Algo1 such as:

L: Length of the rectangle solution obtained by Algo1, using horizontal build, and

W: Width of the rectangle solution obtained by Algo1, using vertical build.

In each iteration, we select R from the set ξ_1, having the smallest lower bound f' and we place it in ξ_2. The set ξ_3 of the new r-packing, satisfying $d_k \leqslant b_k, (k = 1, \ldots, n)$ and $f'(R) <$ Opt (where Opt is the best current solution value), is created by the horizontal builds of R with all elements R' of the set ϑl_R and the vertical builds of R with all elements R'' of the set ϑw_R where ϑl_R and ϑw_R are subsets of ξ_2 such that:

$$\begin{cases} \vartheta_{l_R} = \{q/l_q = l_R + l_p \leqslant L, p \in \xi_2\} \\ \vartheta_{w_R} = \{q/w_q = w_R + w_P \leqslant W, p \in \xi_2\} \end{cases}$$

Where l_p and w_p are respectively, the length and the width of the element p.

The algorithm stops when the value $f'(R)$ of R selected from ξ_1 is greater than or equal to the best current solution value or when the list ξ_1 is reduced into the empty set.

2.2.3. Notion of the dominated model and the order search

The dominated patterns: The dominated pattern notion [13] is adapted to BFBB, to eliminate some useless patterns as follow: Let R and R' be two r-packings, the orthogonal construction between R and R' known by *dominated* pattern *if* there exists $R'' \in \xi_1$ which occupies the empty spaces S_{rem} obtained by orthogonal construction between R and R'. This technique is introduced in BFBB, such that for each new pattern produced by an orthogonal build, we compute the empty space. If, there is a piece of ξ_1 which can occupy this space without violation of the constraints, then this pattern is eliminated.

Order search: The order search [13] is applied to the *Best First Branch and Bound* algorithm, in order to eliminate some symmetrical patterns. It is provided that in the initial open list, at each piece R_i (i∈I), we associate two order searches, horizontal and vertical, where $i=\theta_h=\theta_v=1,2,\ldots, n$.

For each r-packing A, obtained by horizontal build between K and Q, we introduce a new order search such that $\theta_{h(A)} = \min\{\theta_{h(K)}, \theta_{h(Q)}\}$ and $\theta = \theta_{v(A)} = \max_{E \in \xi_1 \cup \xi_2} (\theta_{v(E)}) + 1$

In each iteration, we test (test of the duplicated patterns) for the combined patterns whether the obtained pattern are duplicated, then they wouldn't be constructed.

Test of the duplicated patterns. Let R and R' be two r-packing and θh(R) and θh(R') , respectively two horizontal Order searches of R, R'. $\theta_{v(R)}$ and $\theta_{v(R')}$ respectively a two Vertical Order searches of R and R'.

If R is taken from an Open list and is composed at least by two pieces and if R' is taken from a Closed list and is composed by a unique piece type. Then:

Horizontal build between R and R' is a duplicate model if $\theta_{h(R)} < \theta_{h(R')}$

Vertical build between R and R' is a duplicate model if $\theta_{v(R)} < \theta_{v(R')}$

Improved exact algorithm (IBFBB (:

ξ_1: A set of subproblems;

ξ_2: The list of stored best subproblems;

R, R' and R'': r-pickings;

$f'(R)$: The lower bound of the subproblem containing R;

Opt : The best current solution value;

S_{rem}: The empty space obtained by a horizontal or vertical build between R and R;

$\theta_{h(R)}$: The horizontal order search of R;

$\theta_{v(R)}$: The vertical order search of R.

Input: An instance of the orthogonal rectangular packing problem

Output: The optimal solution value *Opt*

Initial step:

$\xi_1 = \{R_1,\ldots,R_n\}$; $\xi_2=\emptyset$ and *finished = false ;*

Let E_{sup} be the upper bound obtained by applying the greedy procedure presented by **Algo3**; $Opt = E_{sup}$

Principal step:

Repeat

Choose the *r-packing R* with the smallest value of f'; (denoted f'_{min})

If $Opt\text{-} f'_{min} \leqslant 0$; then *fin = true*

Else Begin

Transfer R fromξ_1 to ξ_2

construct all elements of ξ_3 such that:

Each element R'' of ξ_3 obtained by horizontal build between R and the elements $\vartheta l_R$of $\xi_2$such that $\vartheta l_R = \{q/l_R + l_p \leqslant L, p \in \xi_2\}$

Each element R'' of ξ_3 obtained by vertical build between R and the elements $\vartheta w_R$of $\xi_2$such that $\vartheta w_R=\{q/ w_q = w_R + w_p \leqslant W, p \in \xi_2\}$

Test R with all elements of ξ_2. If the model obtained is a duplicated, then the new model is not built.

Each element of ξ_3 satisfies the constraints b_i $(1 \leqslant i \leqslant n)$ and $f' < Opt$;

Each model of $\xi_3$which satisfies the test of dominance is eliminated.

Each element of $\xi_3$is labelled by an order search.

If it ∃ a final packing $R' \in \xi_3 \backslash f'(R') < Opt$, then $Opt = f'(R')$;

To update the set ξ_1 by $\xi_1 \cup \xi_3$ Replace the set ξ ξ_1 /{*non-final packing with the evaluation* $f' \geqslant Opt$ };

If $\xi_1 = \emptyset$ then *fin = true*;

End if

Until *fin = true* ;

End

3. Implementation & results

The proposed algorithm was coded in C and tested on a computer with Pentium 4 CPU 3.00 GHz, and 1 GO of RAM. The performance of our algorithm (IBFBB) is evaluated on several randomly generated problem instances. We consider a group of 50 instances; the number of used pieces is taken in the interval

Table 1. Performance of the improved algorithm BFBB.

Initial Bound AV. Ratio		BFBB		IBFBB	
UB(1)	UB(2)	AV. Time	AV. nodes	AV. Time	AV. nodes
1,21	1,15	86,88	3638,58	13,75	2532,24

Table 2. Performance of the IBFBB compared to the BFBB algorithm.

Gain	(%) AV. time	(%) AV. nodes
BFBB versus IBFBB	84,17	30,40

[3, 16]. The dimensions $(l_i w_i)$ of all pieces are fixed in the interval [1, 80], and the bound $b_i i = 1, \ldots, n$, is randomly taken in interval [1, 9].

Performance of the improved BFBB algorithm

In table 1

AV. Ratio: Average ratio represents the quality of both solutions obtained by the greedy procedure algo1 (UB (1)) and the new greedy procedure algo3 (UB (2))

The ratio is computed by an usual measure A(I)/Opt(I), where A(I) represents the solution value obtained by applying the algorithm A on the instance I and Opt(I) is the optimal solution value of this instance.

AV. time: The average execution time (measured in seconds) which represents the time that both BFBB and IBFBB algorithms need to reach the optimal solution.

AV. nodes: The average nodes which represent the total number of nodes (builds) generated by both BFBB and IBFBB algorithms.

According to the results obtained in table1, we notice that the new upper bound Algo3 is better than Algo1, since it performs (in average) from 1.21 to 1.15 the quality of solutions produced by the greedy procedure (see Algo3).

Table 2 shows the performance of the IBFBB compared to the BFBB algorithm. The computed average time and nodes evaluate the performance of the IBFBB when the dominated and duplicated models strategies are used. In the same table, we observe that the average computational time gain is considerably increased, it is equal to 84,17 and produces an average reduction of 30,40 in term of the number of generated nodes.

4. Conclusion

In this paper, we have proposed an improved version of the BFBB algorithm for solving the orthogonal rectangular packing problem; it is based on the following new proposals:

1. A new upper bound are used at the root node of the search tree
2. A new representation of the closed list data structure is used in order to speed up the construction of new configurations
3. A symmetric and dominate model strategies are introduced to reject some models and to curtail the search in the developed tree.

The experiment results indicate that the IBFBB algorithm performs well and show that the improved algorithm is able to solve efficiently small and medium problem instances within short execution time.

References

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

[2] G. Wäscher, H. Haußner, and H. Schumann, “An improved typology of cutting and packing problems,” *European Journal of Operational Research*, vol. 183, no. 3, pp. 1109–1130, 2007.

[3] M. Chlebík and J. Chlebíková, “Hardness of approximation for orthogonal rectangle packing and covering problems,” *Journal of Discrete Algorithms*, vol. 7, no. 3, pp. 291–305, 2009.

[4] P. Sweeney and E. Paternoster, “Cutting and packing problems: a categorized application-oriented research bibliography,” vol. 43, no. 7, pp. 691–706, 1992.

[5] M. Hifi and R. Ouafi, “A best-first branch and bound algorithm for orthogonal rectangular packing problem,” *International Transactions in Operational Research*, vol. 5, no. 5, pp. 345–356.

[6] D. Liu and H. Teng, “An improved bl-algorithm for genetic algorithm of the orthogonal packing of rectangles,” *European Journal of Operational Research*, vol. 112, no. 2, pp. 413–420.

[7] W. Huang, Y.-L. Wu, S. Lau, C.-K. Wong, and G. Young, “An effective quasi-human based heuristic for solving the rectangle packing problem,” *European Journal of Operational Research*, vol. 141, pp. 341–58.

[8] L. Wei, D. Zhang, and Q. Chen, “A least wasted first heuristic algorithm for the rectangular packing problem,” vol. 36, pp. 1608–1614, 2009.

[9] E. Hopper and B. Turton, “An empirical investigation of metaheuristic and heuristic algorithms for a 2d packing problem,” *European Journal of Operational Research*, vol. 128, pp. 34–57.

[10] V. D. Cung, M. Hifi, and B. L. Cun, “Constraind two-dimensional cutting stock problems a best first branch and bound algorithm,” *International Transactions in Operational Research*, vol. 7, pp. 185–210, 2000.

[11] Y. Cui, Y. Yang, X. Cheng, and P. Song, “A recursive branch-and-bound algorithm for the rectangular guillotine strip packing problem,” vol. 35, no. 1, pp. pp 1281–1291, 2008.

[12] R. Dechter and J. Pearl, *The optimality of A^**. Springer, 1998.

[13] M. Hifi and R. M’hallah, “An exact algorithm for constrained two dimensional two-staged cutting problems,” vol. 53, no. 1, pp. 140–150.

Rachid Ouafi is a Professor of Mathematics at the University of Science and Technology (USTHB), Algiers. He received a Ph.D. in Statistics from Paris6 University, France in 1988, and Ph.D. in Operation Research from the USTHB, Algiers in 2004. He worked as an instructor in the Department of Medical Sciences of the University of Algiers. He is currently director of research at the Laid3 laboratory in the department of Operational Research, Algiers University. His research interests are industrial systems optimization and transportation logistics.

Isma Dahmani is a Ph.D. candidate of Mathematics at Houari Boumedienne University of Science and Technology, Algiers. She received her MS in Operation Research in 2007 from the USTHB, Algiers. Her main area of research is in combinatorial optimization.