A BEST-LOCAL POSITION PROCEDURE-BASED HEURISTIC FOR TWO-DIMENSIONAL LAYOUT PROBLEMS

MHAND HIFI AND RYM M'HALLAH

Abstract. The two-dimensional layout (TDL) optimization problem consists of finding the minimal length layout of a set of shapes or pieces on a stock sheet of finite width and infinite length. In this paper, we solve the TDL problem using a new approximate solution method. First, we describe a constructive method based mainly upon the best-local position search (called BLP). Next, we enhance the BLP method by introducing a *dynamic re-ordering procedure* for re-starting the BLP approach. The resulting algorithm (called RBLP) yields satisfactory results within reasonable computational time. Extensive computational testing on problem instances taken from the literature shows the effectiveness of the proposed approach.

Keywords: best-local position, cutting and packing, dynamic re-ordering, heuristics, optimization.

1. Introduction

Unless $P = NP$, many interesting problems of operations research and artificial intelligence cannot be solved exactly within a reasonable amount of time. Consequently, heuristics must be used to solve large-scale real world problems. Heuristic algorithms may be divided into two main classes: general purpose algorithms designed independently from the optimization problem at hand, and tailored algorithms specifically designed for a given problem. The aim of this paper is to propose a tailored algorithm that approximately solves the two-dimensional layout (TDL) problem. The TDL problem consists of finding the minimal length layout of a set of pieces, with regular or irregular shapes, on a stock sheet of finite width and infinite length. The layout, known as a cutting pattern, does not contain any overlap among the pieces and/or between a piece and one of the edges of the sheet.

The TDL problem is one of the most interesting and complex Cutting and Packing problems. Cutting and Packing problems, referred to as CP in Dyckhoff *et al.* [D90, DF92], constitute a family of natural combinatorial optimization problems, admitted in numerous real-world applications such as computer science, industrial engineering, logistics, manufacturing and production processes. For instance, the TDL problem is encountered daily in the production of glass, metal sheets, photographic film, plastics, and in job scheduling.

The paper is organized as follows. First, in Section 2, we briefly present a review of some exact and approximate sequential algorithms for the regular or irregular TDL problems. In Section 3, we explain the motivation of the design of the proposed heuristics. In Section 4, we describe the first heuristic, called the *best-local position method*, which consists of sequentially positioning a set of ordered pieces. Each piece is tested locally for a set of potential positions defined with respect to already positioned pieces. The best position is one that minimizes the overall length while maximizing the packing of the already placed pieces. The best-local position method yields a satisfactory initial solution that is further enhanced by the *re-ordering procedure* discussed in Section 5. The performance of both of our heuristics is then tested, in Section 6, on a set of problem instances extracted from the literature, and representing a variety of shapes. Finally, in Section 7, we draw some conclusions and discuss future possible extensions.

2. Existing sequential TDL algorithms

Because of its importance and despite its NP hardness ([FPT81, MDL91]), the layout problem has been widely studied in the literature. A survey of the different approaches to this problem and its variations is available in [CFC94, DD95, DF92, HT01]. The design of the solution approaches depends generally on the particular framework of the application -giving rise to the particular TDL problem-, and the available computational resources.

Dori and Ben-Bassat ([DB83]) discussed an efficient nesting approach of congruent convex figures based on approximating a convex polygon by a polygon with fewer sides. Li and Milenkovic ([LM95]) considered the so-called compaction algorithm to solve the particular problem of planning motions of the non-convex polygonal shapes in an available layout in an attempt to enhance the layout in some fashion: increase efficiency, open up space for new shapes, or eliminate overlap among shapes.

Blazewicz *et al.* ([BHW93]) were the first to apply tabu search to irregular packing/cutting problems. The approach starts with a feasible solution, constructed by applying a simple placement procedure, and enhances it using a tabu search. For each selected item, different positions are tested and the best one is kept. The move describes a change of the allocation of an item from one position to the other, prohibiting overlapping configurations. Items that have changed their position during recent iterations are members of the tabu list. The best admissible move is determined by the objective function aiming at placing the rightmost elements into the void areas of the feasible solution. In comparison to Albano and Sapuppo's ([AS80]) method, the tabu search produced better results.

Poshyanonda and Dagli ([PD92]) proposed a neural network genetic algorithm (GA) hybrid approach. Their algorithm used the genetic algorithm to generate a sequence for the placement algorithm, which is based on a sliding method and combined with an artificial neural network.

In [DJ96], the authors used GA while employing task decomposition and contact detection. Hopper and Turton ([HT99]) investigated meta-heuristic and heuristic algorithms for the TDL rectangular cutting problem. In [FAH93], the authors detailed a hybrid approach for optimal nesting using GA and local minimization search. In [BM96], a modified genetic approach to a nesting problem has been proposed. Jakobs [J96] proposed a genetic algorithm. His approach represents a packing/layout by means of a permutation reflecting the order in which the shapes are packed. The packing positions are determined through the Bottom-Left strategy (BL). BL is very similar to the packing scheme initially proposed by Chazelle [C83]. The author proved that his representation turns out to be particularly effective for implementing the genetic algorithms crossover and mutation operators. Liu and Teng ([LT99]) improved Jakobs' BL procedure ([J96]) and implemented a GA for the orthogonal packing of rectangles.

Leung et al. ([LYC99]) developed a *Difference Process Algorithm*. This algorithm is another BL procedure that can access enclosed areas in the partial layout. Every insertion of a new current item in the layout/packing creates two empty rectangular spaces at its top and right side. The algorithm keeps track of the newly generated spaces selecting the one that is closest to the bottom-left corner of the object and sufficiently large for the allocation of the next rectangle.

Lai and Chan ([LC97]) used an evolutionary algorithm, which is hybridized with a local procedure. The algorithm does not use any cross-over operator and is only based on selection and mutation processes. The heuristic decoder is similar to the bottom-left algorithm used by Leung *et al.* ([LYC99]) and places the item in the position that is closest to the lower-left corner of the object.

Ribeiro *et al.* ([RCO99]) suggested the use of logical constraints to express acceptable patterns in an irregular TDL problem. Hifi [H99] proposed several approximation algorithms to solve the two-dimensional guillotine strip cutting/packing problem. These algorithms combine dynamic programming techniques and heuristics, such as Hill-Climbing.

For the rectangular and circular cases, Stoyan *et al.* [SY98, SN99] have developed a mathematical model for optimally solving both versions of the problem. Hifi [H98] developed exact algorithms for solving the special case of guillotine rectangular strip cutting/packing problems. To our knowledge, no exact solution procedure for this irregular TDL problem exists in the literature.

In this paper, we present a heuristic that combines a *best-local position* procedure (BLP) with an *incremental subset items* procedure. The heuristic, which solves the TDL (ir)regular layout problem, uses the incremental subset items procedure to determine the (near)optimal ordering of the pieces while it relies on the best-local position method to search for the best layout of the ordered set of pieces. The BLP attempts to mimic the human placement strategy by adopting a dynamic ordering of the pieces. It prevents the problem of overlap computation by computing the maximum feasible translation step; it prohibits positions that can cause an overlap. It uses simple geometric computations based on projections; thus, avoiding the shortcomings of convex enclosure based methods (for more details the reader can refer to [MDL91]).

3. An overview of the Re-ordering BLP approach: RBLP

The TDL problem can be divided into two interrelated subproblems: finding the best ordering of n pieces, and identifying the best packing of the ordered set of the n pieces. To illustrate this interrelationship, we let P_i denote piece i, $i = 1, \ldots, n$. The orderings P_1, P_2, P_3 and P_1, P_3, P_2 yield, using a best local positioning strategy, the two different layouts of Figure 3.1. On the other hand, the layouts of Figure 3.1 can be obtained for the same ordering P_1, P_2, P_3 but using two different positioning strategies.

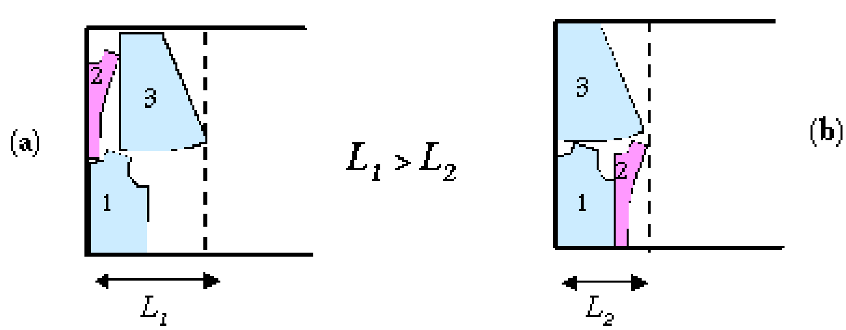

Figure 3.1: The relationship between the two facets of the problem..

Our heuristic addresses these two subproblems in a unique manner. The heuristic starts by ordering the pieces according to size; that is according to length, width or area. It packs the ordered set using the best-local position method discussed in Section 4. It then improves the near optimal packing using the dynamic reordering procedure discussed in Section 5. Indeed, it alters the near optimal packing in an attempt to get a "better"order of the pieces. This order, inspired from the current near optimal layout, moves any blocking piece from its current order to the last order in the set.

4. The Best-Local Position method: BLP

4.1. The principle of the BLP approach

The BLP approach begins by positioning the first piece on the left most bottom most position of the rectangular stock sheet. The following piece will be placed in one of five possible positions. Indeed, each placed piece generates a set of five possible positions for any piece to be packed: $(\bar{x}, 0)$, $(0, \bar{y})$,$(\underline{x}, \bar{y})$, $(\bar{x}, \bar{y})$, and $(\bar{x}, \underline{y})$,where $\bar{x}$, $\underline{x}$, $\bar{y}$, $\underline{y}$ are the maximum and minimum x and y coordinates of the already placed piece.

Only those positions that do not cause an overlap of the piece to be positioned with an already placed one are retained as valid candidate positions. Evidently, the candidate lists differ according to the piece to be placed. Since few of the candidate positions are duplicates, each position will appear only once in the candidate list. For each candidate position, the piece to be placed is packed horizontally and vertically with respect to the pieces on its left and bottom.

The positioning of a piece with respect to an already placed one is done such that the two pieces are in contact without superposition. In fact, the piece is translated in the vertical and horizontal directions until it comes in contact with a placed piece. The translation distance is computed using projections of the left L and bottom B parts of the piece to be placed on the right R and upper U parts of the pieces already positioned. For the horizontal translation, L is projected on R and R is projected on L. The translation step is the minimum of the two projection distances. (In case of non circular pieces, each breakpoint is projected on a segment on the other piece. The projection distance is the minimum among all projection distances of a breakpoint of L on R and vice versa.) The vertical translation is identical to the horizontal one. It is obtained by replacing L by B and R by U. Different orders of translations are considered in order to get the best packing.

This compaction step might be relatively expensive in terms of run time if the shapes are concave. It remains however definitely a lot cheaper than those based on the computation of minimal convex enclosures. Our previous experience has shown that the irregular example, referred to as $G8$ in the computational section requires more than 8 minutes to generate the same layout that the proposed method obtains in 17 seconds. The geometric projections are very simple; therefore are rapidly executed. This enhances the speed of layout generation; a particularly important aspect since the constructive method is to be combined with a re-ordering procedure. It is to be emphasized that any point that is detected by the no fit polygon and not directly by our method is eventually reached with our heuristic through the series of horizontal and vertical compactions.

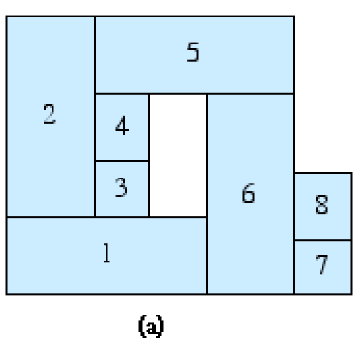

Figure 4.2: (a) Liu and Teng ([LT99]) improved BL algorithm and (b) the BLP approach on a small rectangular example.

If placing a piece on a candidate position violates the width constraint, the candidate position is immediately fathomed. Among all feasible candidate positions, the one that places the piece to the left most, bottom most position is preferred as it enhances the packing of the layout.

Our best local position method places pieces in holes generated by already placed ones without having to recognize them or to delimit them as holes; that is, without any geometric computation of the contour of the hole. By placing a piece in a position without causing an overlap with already placed pieces, we either position it to the right or top of the contour of the placed pieces or fill an existing hole (if the piece can fit in such a hole). In the latter case, which is herein of interest, we don’t worry about the hole itself or its shape. Indeed, at no time do we deal with the holes. Their filling is simply a consequence of our positioning strategy. For instance, in Figure 4.2.(b), when positioning

P_7, we didn't have any knowledge about the hole visualized in Figure 4.2.(a). For BLP, the position coinciding with the right lower corner of P_3 yields no overlap with pieces $P_1, \ldots, P_6$, so this position is an eligible one. Similarly, the position defined by the right lower corner of P_4 yileds no overlap with pieces $P_1, \ldots, P_6$, so offers an alternative position of P_7. However, the first position is better since the y-coordiante of its reference point is lower. Since the pieces are considered according to size, then the holes are being successively filled with the largest fitting piece yet available. This strategy guarantees a best left bottom packing.

It is to be emphasized that the proposed best local position method is neither the merely bottom-left procedure (BL) nor the improved bottom-left procedure proposed respectively in [J96] and [LT99]. The proposed procedure considers a set of possible positions for each piece to be placed. Only one of theses positions is that used in [J96] or [LT99]. For instance, our procedure successfully constructs the perfect puzzle of Figure 4.2.(b) while the method proposed in [LT99] can not position P_7 and P_8 (see Figure 4.2.(a)) in the perfect puzzle.

Note that using a set of possible positions does not get intractable as the number of pieces increases. Several of these positions are duplicates or infeasible, i.e., violate the width constraint or cause overlap after the translations of the piece.

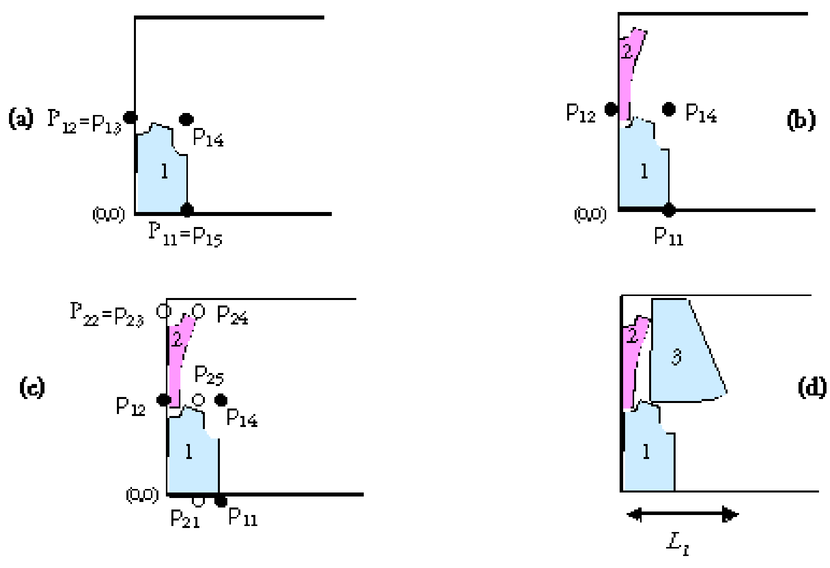

Figure 4.3: Applying some steps of the BLP on a small irregular example.

4.2. The BLP on small examples

To illustrate how the best local position method proceeds, we consider the three piece example of Figure 4.3, where the pieces are denoted $P_i,\ i = 1, 2, 3$.

1. P_1 is placed on the left most bottom most position of the stock sheet; i.e., it is placed in the $(0, 0)$ coordinates point. As it is illustrated by Figure 4.3.(a), placing P_1 automatically generates five possible positions for future pieces to be placed.
2. If p_{ij} is the j-th position generated by P_i, then $p_{11} = p_{15}$, and $p_{12} = p_{13}$; which reduces the candidate positions list to three positions. None of these positions is infeasible to P_2. If placed in p_{14}, P_2 is subject to a series of horizontal and vertical translations in different orders. Evidently, the best local position of P_2 is obtained by a vertical translation of P_2 from p_{12}, as shown in Figure 4.3.(b).
3. P_2 generates another set of five positions (see Figure 4.3.(c)) with $p_{22} = p_{23}$. The positions p_{12}, p_{21}, p_{22} and p_{25} are infeasible. The positions p_{12}, p_{21} and p_{25} cause an overlap while p_{22} violates the width constraint. Therefore, the candidate positions list reduces to three possible initial positions for P_3; i.e., $p_{11},\ p_{14}$ and p_{24}. The best final position for P_3 is that obtained by placing it in p_{14} and translating it horizontally then vertically, to produce the result of Figure 4.3.(d).

Experimental results suggest that BLP yields near optimal packings when the pieces are ordered according to size; that is according to length, width, or area. This is in conformance with the rules deducted from observing human markers.

Human markers start by diving the pieces into three sets: big, small, and other (those that are very long but narrow or vice versa.) He / she positions the large and other pieces, then fills the holes with the small pieces. That might require few translations of already positioned pieces (in particular those belonging to the other class.)

Quantifying the notion of big is not an easy task. Grinde [G96] developed a set of sixteen very complicated criteria to define size, but was not successful in proving the superiority of any of them. A comparison of the layouts he obtained to those provided by the BLP procedure using the length, width, and area criteria has shown that these criteria are satisfactory.

4.3. Comments on BLP

BLP yields near optimal packings. However, when not used in conjunction with a reordering procedure of the pieces, it is recommended that it be coupled with other rules that handle particular exceptions as explained herein.

Placing a piece in its best local position can sometimes be myopic, and should be remedied in certain cases with either a dynamic ordering of the pieces or the choice of a different position. For instance, if the four pieces of Figure 4.4.(a) were to be positioned in the order P_1, P_3, P_2 and P_4, then, placing P_3 in its best local position creates the artificial hole, shown as the shaded area, to the right of P_1 in Figure 4.4.(b). This hole can not be used to place P_2 or P_4. In this case, it is preferable to postpone positioning P_3 till P_2 is placed. Another alternative is to position P_3 to the right of P_1 since that represents a better potential of a global optimum. Not opting for the best local position depends not only on the number of pieces yet to be placed but also on their sizes.

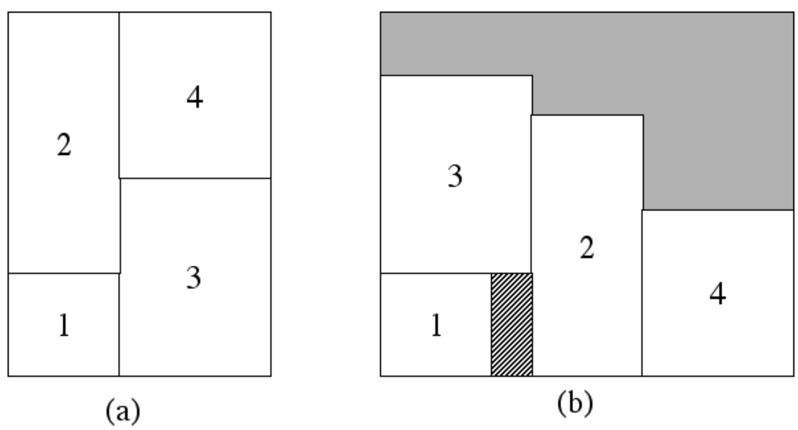

Figure 4.4: Illustration of a dynamic ordering of the pieces.

Either of the aforementioned alternatives avoids creating artificial holes that have a high probability of not being exploited by forthcoming pieces. Thus, the method gets away from its simplistic and myopic placement strategy and further approaches the placement strategy of human markers. In a sense, it uses a dynamic positioning, where the positioning of a piece is temporarily postponed.

The additional rules are of interest only when BLP is not coupled with a reordering procedure. A reordering procedure is always capable of finding the best layout for any given positioning strategy. For instance, consider a perfect puzzle and a placement strategy. Then, you should be able by looking at the perfect puzzle to identify one of the "perfect" orderings that would reconsitute the puzzle. For example, the ordering $P_1, \ldots, P_4$ of Figure 4.4.(a) will be optimal when applied using the BLP (without any additional rules).

5. The Re-ordered Best-Local Position method: RBLP

The dynamic reordering procedure tries to improve the length of any layout by changing the order in which the pieces are being placed. The BLP procedure can sometimes create artificial holes that can't be used by the pieces that remain to be placed. This is particularly the case when a longer piece is positioned on top of a shorter piece. For instance, in Figure 4.4.(b), P_3, which is on top of P_1, has created an artificial hole to the right of P_1. In this sense, P_3 is a blocking piece. P_3 should be removed from the second position in the order and placed at the end of the order. It follows that the order should change from $P_1,\ P_3,\ P_2,\ P_4$ to $P_1,\ P_2,\ P_4,\ P_3$.

5.1. The RBLP method

In general, the *incremental subset items* procedure looks for those "blocking"pieces and removes them from their current position in the order to the end of the ordered set. The layout is then packed using the BLP procedure. This iterative procedure, called RBLP herein, is repeated until no further improvement is possible. **Box 1** describes the main steps of the RBLP procedure.

RBLP uses the initial layout obtained by the BLP method (Phase 1). It iteratively, for each piece, identifies the blocking pieces positioned on top of it. It handles the blocking pieces one at a time. Each blocking piece is removed from its current position in the current order and tagged to its end. BLP is applied to each current order. If the yielded layout is better than the incumbent solution, the optimal length and order is updated. The algorithm stops when all the blocking pieces for the last piece of the initial layout have been handled or when the optimal solution has been identified.

Box 1. ***The RBLP method***

Input: an instance of the TDL problem.
Output: a (near)optimal solution with length L^*;

<u>`The first phase.`</u>

1. Let N be the set of pieces ordered according to size; i.e.,
   (a). $N = \{P_{[1]}, P_{[2]}, \cdots, P_{[n]}\}$, where n denotes the number of pieces to pack/cut;
   (b). Apply the BLP procedure.
   (c). Set $i = 1,\ N^* = N$, and $L^* = L$, where L is the minimal length of the current layout.
   (d). Let $\underline{L}$ be the natural lower bound (see Section 6).

2. If $L^* = \underline{L}$ then set `Stop=true`; else set `Stop=false`.
   If $\underline{L}$ is not "evident" (for the irregular case) to compute, then set $\underline{L} = 0$ and `Stop=false`.

<u>`The second phase.`</u>

While not(`Stop`) **do**

A. Let S be the subset of N such that $\bar{x}_i \geq \underline{x}_j \geq \underline{x}_i$, and $\bar{x}_j > \bar{x}_i$,
   where $\bar{x}_k$, and $\underline{x}_k$ are the maximum and minimum x-coordinates of $P_{[k]},\ k = 1, \cdots, n$.
   Let n_s be the number of elements of S. Set $m = 1$, and $N' = N$.

B. Reorder N' such that the m-th element of S is moved to the end of N'.

C. Apply BLP to the new order N'.
   If $L < L^*$ then set $N^* = N$ and update L^* with L.

D. If $L = \underline{L}$ then set `Stop=true`.

   Else set $m = m + 1$. If $m \leq n_s$, goto B.

E. If $i = n$ then set `Stop=false`;
   Else Set $i = i + 1$.

Enddo

5.2. The RBLP on a small example

To illustrate the dynamic reordering procedure, we consider the example of Figure 5.5. It is to be emphasized that the example was only designed to illustrate how the algorithm works. The BLP would never yield the layout of Figure 5.5.a if the pieces are ordered according to size.

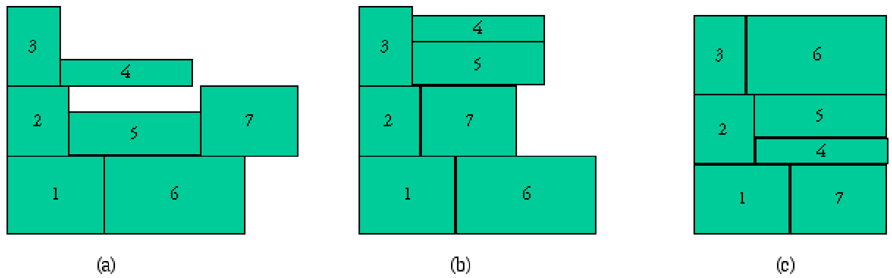

Figure 5.5: The RBLP procedure on a small example.

1. If the initial layout is that given by Figure 5.5.(a), then $N = \{P_1, \ldots, P_7\}$. Since the line $x = \bar{x}_1$ intersects both P_4 and P_5, then $S = \{P_4, P_5\}$.
2. We set $N^* = N$ and $N' = N$.
3. We remove P_4 from its current ordering, and position it at the end of N'; i.e., $N' = \{P_1,\ P_2,\ P_3,\ P_5,\ P_6,\ P_7,\ P_4\}$.
4. We apply BLP to the new ordering N'. The layout obtained is that illustrated in Figure 5.5.(b). Since, this layout has a shorter length than the original one, N^* and L^* are updated.
5. We remove P_5 from its current ordering, and position it at the end of N'; i.e., $N' = \{P_1,\ P_2,\ P_3,\ P_6,\ P_7,\ P_4,\ P_5\}$.
6. We apply BLP to the new ordering N'. The layout obtained is that illustrated in Figure 5.5.(c). Since, this layout has a shorter length than the current optimal one, N^* and L^* are updated. This layout turns out the be the optimal layout.

6. Computational results

In this section, we first discuss the complexity of the BLP and RBLP aproaches, then present their performance.

At worst, BLP is of complexity $O(2.5n(n-1))$. Indeed, once the first piece is positioned, any other piece can be placed in one of $5 * n_p$ positions, where n_p is the number of pieces already placed. So, the number of positions is $5(1+2+3+\cdots+(n-1))$. The complexity therefore follows.

Since in any given layout, the number of blocking pieces can not exceed $(n-1)$, RBLP runs BLP at most $(n-1)$ times. Subsequently, RBLP is at worst of complexity $O(2.5 * n(n-1)^2)$.

The proposed algorithms, coded in `Fortran`, were run on a `Pentium III` (733 Mhz and 128 Mo of RAM), with CPU time limited to twelve minutes. Both algorithms were tested on different problem instances extracted from the literature. We have made these instances publicly available from `ftp://panoramix.univ-paris1.fr/pub/CERMSEM/hifi/TDL`, hoping to aid further development of exact and approximate algorithms for the TDL problem.

The different problem instances								
Circular			Rectangular			Irregular		
Inst	W	n	Inst	W	n	Inst	W	n
SY1	9.5	30	SCPL1	425	110	G8	59	8
SY2	8.5	20	SCPL2	127	120			
SY3	9	25	SCPL3	225	84			
SY4	11	35	SCPL4	365	102			
SY5	15	100	SCPL5	165	102			
SY6	19	100	SCPL6	657	56			
			SCPL7	357	139			
			SCPL8	475	156			
			SCPL9	175	117			
			Jak1	40	25			
			Jak2	40	50			

Table 1: Test problem details.

The problem instances we considered are summarized in Table 1. We tested a total of eighteen problems corresponding to circular, rectangular, and irregular pieces. For the circular TDL version of the problem, we used the six test problem instances, referred to as `SY1-SY6` in Table 1, optimally solved by Stoyan and Yaskov [SY98]. In this way, we compare our heuristic to an exact method.

For the rectangular TDL problem, we tested eleven problems. Since the rectangular problems available in the literature are limited, we used nine problems, noted as `SCPL1-SCPL9` in Table 1, that were solved optimally by Hifi [H99] for guillotine cuts. We then compared the performance of our heuristic to those proposed in [J96] and [LT99]. In fact, we used their two rectangular TDL problems as our last two instances, noted `Jak1` and `Jak2` in Table 1.

Finally, for the irregular TDL problem, we compare our heuristic to that proposed in [G96]. The instance is noted `G8` in Table 1.

6.1. The circular TDL version

First, we tested the six test problems of Stoyan and Yaskov [SY98], for which the optimal solutions are known. Column 2 labeled "*Best or Opt* "contains either the best known solution or the optimal one. The results yielded by BLP

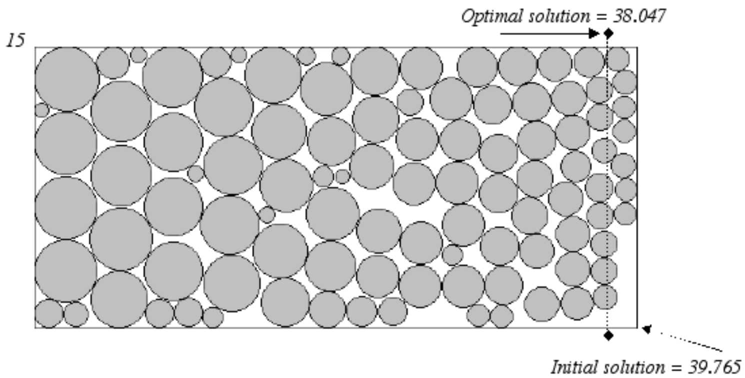

Figure 6.6: The Stoyan and Yaskov's example (SY5): the BLP initial solution, of length 39.765 corresponding to a 4.517% gap.

and RBLP for SY1-SY6 are reported respectively in Columns 3-5 and 6-8 of Table 2. Column 3 (respectively 6) contains the best length, of the considered strip, reached by the BLP (respectively RBLP). Column 4 shows the gap of the yielded solution from the optimal (or the best known) one. The gap $= 100 \times \frac{A(I)-Opt(I)}{Opt(I)}$, where $A(I)$ represents the length produced by the algorithm and $Opt(I)$ is the optimal length of problem instance I. Column 5 (respectively 8) displays the corresponding run time of the approach for the problem instance. For the irregular instance (G8), the negative *gap* values in columns 4 and 7 indicate that the considered algorithm (BLP or RBLP) improves the best known solution by $-gap\%$.

		BLP			*RBLP*		
Inst	L(opt)	L(BLP)	%Dev.	CPU	L(RBLP)	%Dev.	CPU
SY1	17.491	18.876	7.917	< 1	18.415	5.281	15
SY2	14.895	16.562	11.192	< 1	15.358	3.109	10
SY3	14.930	15.679	5.106	< 1	15.664	4.916	12
SY4	24.355	25.272	3.766	< 1	24.983	2.578	17
SY5	38.047	39.765	4.517	84	38.851	2.428	632
SY6	38.647	40.671	5.235	76	39.650	2.595	581
G8	156	142.5	-8.654	2	142.3	-8.782	17

Table 2: Performance of both BLP and RBLP on circular and irregular TDL problem instances.

As Table 2 shows, for the circular case, the BLP gives good quality results. On average, it is 6.288% from the optimum, with a worst-case of 11.192%. The BLP is however very fast. Its average run time is 27.333 seconds, with the largest observed value equal to 84 seconds, corresponding to a 100 piece problem. The BLP can therefore be considered a useful starting point for more complex procedures.

The RBLP, on the other hand, produces better results, but within a slightly larger computational time as observed in Table 2. Indeed, it produces an average deviation of 3.484%, varying in the interval $[2.428\%, 5.281\%]$. It is noteworthy that the average computational time is under four minutes, which is reasonable considering the good quality of the results.

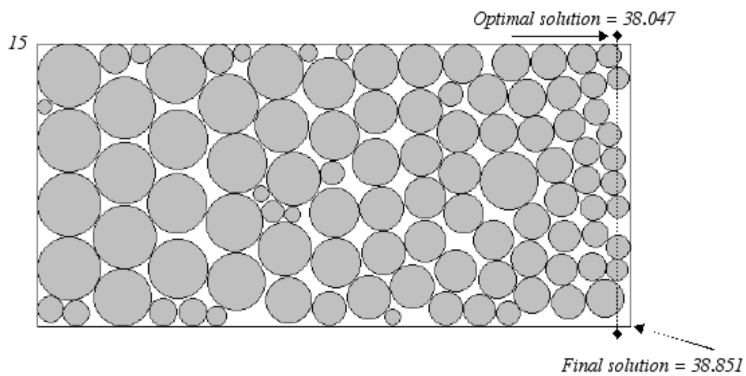

Figure 6.7: The Stoyan and Yaskov's example (SY5): the RBLP final solution, of length 38.851 with a 2.428% gap.

Figure 6.6 shows the structure of the initial solution of the instance SY5 (see Table 1) produced by the BLP. This solution corresponds to a 4.517% gap. Finding this solution requires 84 seconds. Figure 6.7 shows the structure of the final solution reached by the RBLP.

The seemingly high computational time for the examples SY5 and SY6 can be explained as follows.

First, these two examples were run using a Real representation of the variables instead of the Integer(2) representation adopted for all the other instances. When an Integer(2) representation is used, all data are initially multiplied by 1000 prior to running the placement procedure, then divided by 1000. This guarantees a precision of 0.001 of the layout length and positions. Any higher precision is of no interest in a practical setting. Obviously, the Integer(2) representation is computationally cheaper. However, for those two particular cases, the optimal length of the layout is greater than the largest allowable Integer(2). Subsequently, it was not possible to use an Integer(2) representation. Limited experimentation shows that the real representation is on average 30% more expensive than its Integer(2) counterpart.

Second, the number of possible positions gets largers as the number of pieces increases. The number of immediately fathomed positions is relatively small since these positions are not inside already placed pieces. Consequently, we have to preprocess each position to check if it would yield an overlap with already placed pieces.

Third, the sizes of the pieces are very diverse. This implies, in terms of layout, that smaller pieces have a large number of feasible positions. Choosing the best among these feasible positions is also time consuming.

Finally, as far as RBLP is concerned, the larger the number of pieces and the more diverse their sizes, the larger the number of blocking pieces. Since the original BLP is time consuming and the number of reruns of BLP is large, the RBLP winds up with a relatively large computational time. Usually, however, we do not need to investigate all of the blocking pieces. The first ones on the list are the ones with the highest impact on the solution (principle of diminishing return.)

6.2. The irregular TDL version

Second, the BLP and RBLP methods are tested on an irregular TDL problem. This example places non-convex irregular pieces on a rectangular strip of fixed width and infinite length. This example, instance `G8`, is extracted from Grinde [G96] (where the author uses a heuristic based on the minimal-area convex enclosure described in Grinde and Cavalier [GC95]). It corresponds to the pattern of a single size woman's dress with a total of eight non convex pieces to be displayed on a 59 in wide cloth.

The results obtained by both BLP and RBLP for this instance are given in the last line of Table 2.

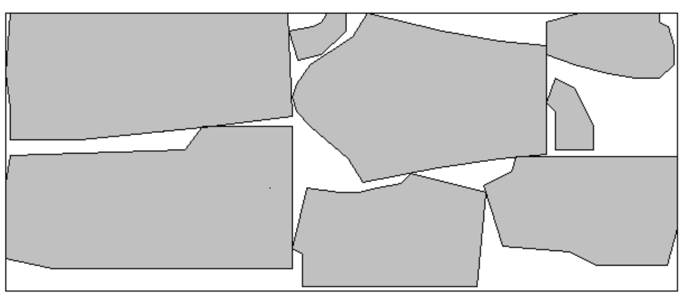

Figure 6.8: Grinde's example: the initial solution produced by the BLP approach.

It is noteworthy that BLP succeeds in improving the known solution. The RBLP further improves the solution obtained by BLP. BLP took approximately two seconds to find the solution shown in Figure 6.8 with a length of 142.5, and about few seconds to produce a slight enhancement as shown in Figure 6.9 (with a length of 142.3) when the RBLP approach is applied.

It is noteworthy that both BLP and RBLP approaches yield better solutions than that of Grinde [G96] (the best layout obtained by the author is 156). Indeed, it is 13.5 inches shorter than the layout published in [G96] when CP is applied, and 13.7 inches when RBLP is used. In this case, the RBLP approach produces an equivalent solution because of the small number of pieces considered. It turns out that this solution is the optimal one. A human marker will opt for this solution, and total enumeration of possible orders and positions of the eight pieces gives an equivalent solution with the same length.

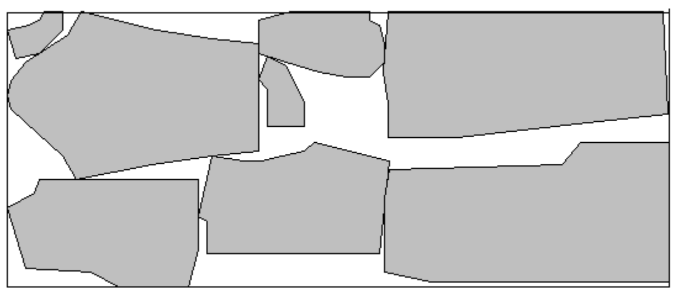

Figure 6.9: Grinde's example: the final solution given by the CP_GA approach.

Inst	L(Opt) or LB	L(BLP)	*BLP* %Dev.	CPU	L(RBLP)	*RBLP* %Dev.	CPU
SCPL1	49*	57	16.327	14	54	10.204	28
SCPL2	151*	155	2.649	10	155	2.649	30
SCPL3	156*	168	7.692	6	166	6.410	18
SCPL4	59*	61	3.390	14	61	3.390	30
SCPL5	129*	135	4.651	10	134	3.875	30
SCPL6	19*	22	15.789	4	21	10.526	16
SCPL7	121	121	0	18	121	0	18
SCPL8	64*	68	5.882	34	66	3.125	60
SCPL9	90*	98	8.889	11	94	4.444	30
Jak1	15	17.0	13.333	1	16	6.667	5
Jak2	15	16.5	10.000	5	16	6.667	25
Average			8.054	11.545		5.268	26.363

Table 3: Performance of both the BLP and the RBLP on rectangular TDL test problems. The symbol $*$ means that the reported value denotes the lower bound of the treated instance.

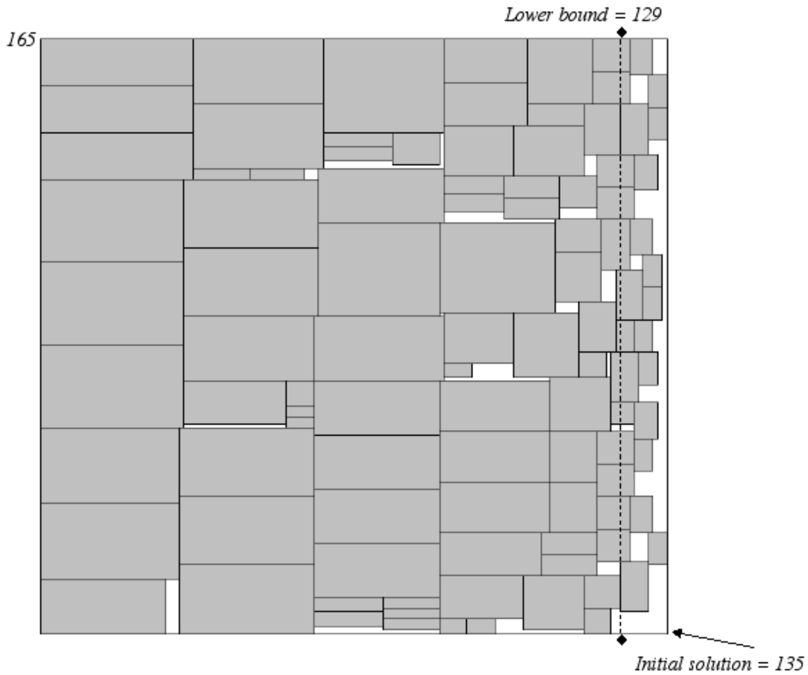

Figure 6.10: The BLP initial solution pattern of instance `SCPL5`. It has a 135 length and a 4.651% gap.

6.3. The rectangular TDL version

Third and last, we tested our approaches on 11 rectangular TDL problems: `SCPL1-SCPL9` taken from Hifi [H99], and `Jak1-Jak2` from Jakobs [J96]. Given that the optimal solution for eight of the first nine problems are not available, we compare the solutions we generated to a lower bound. The most natural lower bound is the sum of the surfaces of all pieces divided by the width of the strip.

The computational results appear in Table 3. Column 2, labeled “*L(Opt) or LB* ”contains either the optimal solution, if it is known, or a lower bound (that has not been proven optimal). Whenever a lower bound is used, the problem instance is marked with a $*$ sign. The lower bound $LB = \left(\sum_{i=1}^{n} l_i w_i \beta_i\right)/W$, where β_i denotes the demand value of the i-th rectangular piece of length l_i and width w_i. (Evidently, none of these lower bounds can equal the optimal solution unless the pieces form a perfect rectangular puzzle of width W). Only

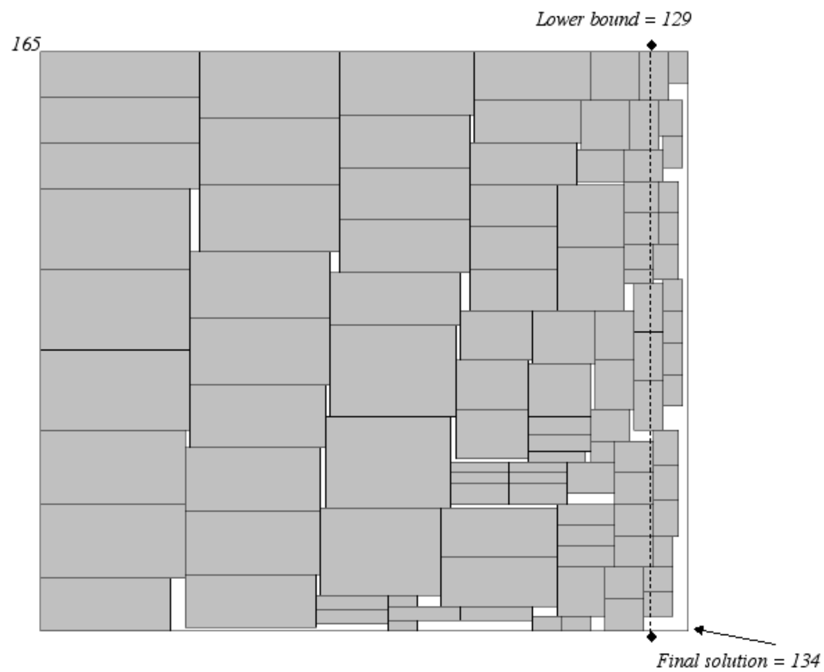

Figure 6.11: The RBLP final packing pattern (of instance `SCPL5`). It has a 134 length and a 3.875% gap.

the optimal solutions for `SCPL7` and `Jak1-Jak2` are known.
`SCPL7` has two pieces of length 121 each; thus, any optimal layout has to be as long as any of the two pieces. Jakobs [J96], on the other hand, artificially created his two problems by cutting a rectangle into a number of smaller rectangles.

From Table 2, we observe that:

1. The BLP yields good quality results. It is on average 8.054% from the optimum. Occasionally, it obtains seemingly poor results, with a worst-case of 16.327%. It is suspected in these cases that the lower bound is not very tight. Even though the problem sizes are relatively large, BLP remains very fast. The average runtime is 11.545 seconds with the largest observed value being 34 seconds (instance `SCPL8`, which is a very large-scale problem instance).

2. The results of RBLP, globally, improved the results of all test problems. The percentage deviation varies in the interval $[0, 10.526]$, with an average percentage deviation equal to 5.268%. Despite its relatively longer (than BLP) runtime, RBLP appears to be a good choice when a better quality solution is needed.

Figures 6.10-6.11 show the solution of the instance SCPL5 produced by both the BLP and the RBLP. The pre-processing procedure creates the packing shown in Figure 6.10, whose length is 135. Generating this packing takes approximately 10 seconds. This solution, corresponding to a deviation of 4.651%, is a good starting solution.
Figure 6.11 illustrates the structure of the (final) packing pattern created by the RBLP after 30 seconds. The solution has obviously improved, being only 3.875% from the optimum.

7. Conclusion

We have developed a new best local position heuristic for solving the regular and irregular two-dimensional layout problem. This heuristic is used in conjunction with a dynamic reordering strategy that is inspired from the current layout of the pieces. The heuristic avoids the lengthy geometric computations required for the detection of overlaps and/or definition of the minimal enclosure area. The reordering strategy determines the "best"ordering of the pieces while the layout heuristic searches for the best layout of the ordered set of pieces. Extensive testing of different problem instances taken from the literature shows that the heuristics yield good solutions within a very short computing time.

The proposed heuristics might further be improved using either of the following two directions of research.

The first direction of research is to design a (sequential) hybrid approach combining genetic algorithm (see Jakobs [J96]) and tabu search (see Blazewicz *et al.* [BHW93]). Each of these two meta heuristics explores and exploits the search space in its own way. Since none of them consistently dominates the other, it is better to hybridize both approaches instead of trying to improve the performance of either the diversifying or the intensifying meta heuristic. Hybridization makes the two meta heuristics compensate each other and federate their behaviors.

The second direction of research is to design parallel algorithms able to solve approximately (or exactly) large-scale TDLs and similar problems of the same family. The use of parallel approaches will increase the size of the problems that can be solved efficiently. It is stipulated that a parallel hybrid algorithm, combining the BLP and the incremental subitems procedure (or a good representation of crossover operators in genetic algorithms), will increase the size of the problems that can be solved efficiently and should converge very fast.

Acknowledgments. The authors thank Professor V. Zissimopoulos for kindly providing some of the data sets. Thanks are also due to Van-Dat Cung for his helpful comments and suggestions.

References

[AS80] A. Albano and G. Sappupo, *Optimal Allocation of two-dimensional irregular shapes using heuristic search methods,* IEEE Transactions on Systems, Man and Cybernetics, SMC-10, pp. 242-248, 1980.

[BHW93] J. Blazewicz, P. Hawryluk and R. Walkowiak, *Using a tabu search approach for solving the two-dimensional irregular cutting problem,* Annals of Operations Research, vol. 41, pp. 313- 327, 1993.

[BM96] C. Bounsaythip and S. Maouche, *A genetic approach to a nesting problem,* Proceedings of the 2NWGA, Vaasa, Finland, pp. 89-104, 19-23 August 1996.

[C83] B. Chazelle, *The bottom-left bin packing heuristic: an efficient implementation,* IEEE Transactions on Computers, vol. 38, pp. 423-429, 1987.

[CFC94] C.H. Cheng, B.R. Feiring and T.C.E. Cheng, *The cutting stock problem - A survey,* International Journal of Production Economics, vol. 36, pp. 291-305, 1994.

[DJ96] R. Dighe and M.J. Jakiela, *Solving pattern nesting problems with GA employing task decomposition and contact detection,* Evolutionary Computation, vol. 3, pp. 239-266, 1996.

[DB83] D. Dori and M. Ben-Bassat, *Circumscribing a convex polygon by a polygon of fewer sides with minimal area addition,* Computer Vision, Graphics and Image Processing, vol. 24, pp. 131-159, 1983.

[DD95] K.A. Dowsland and W. Dowsland, *Solution approaches to irregular nesting problems,* European Journal of Operational Research, vol. 84, pp. 506-521, 1995.

[DDB98] K. Downsland, W. Downsland and J. Bennell, *Jostling for position-local improvement for irregular cutting patterns,* Journal of Operational Research Society, vol. 49, pp. 1-12, 1998.

[D90] H. Dyckhoff, *A typology of cutting and packing problems,* European Journal of Operational Research, vol. 44, pp. 145-159, 1990.

[DF92] H. Dyckhoff and U. Finke, Cutting and packing in production and distribution, a typology and bibliography, *Physica Verlag,* Wurzkurg, 1992.

[FPT81] R.J. Fowler, M.S. Paterson and S.L. Tatimoto, *Optimal packing and covering in the plane are NP-complete,* Information Processing Letters, vol. 12, pp. 133-137, 1981.

[FAH93] K. Fujita, S. Akagi and N. Hirokawa, *Hybrid approach for optimal nesting using a genetic algorithm and a local minimization algorithm,* Proceedings of the ASME Conferences on Advances in Design and Automation, vol. 65, pp. 477-484, 1993.

[GC95] R. B. Grinde and T. M. Cavalier, *A new algorithm for the minimal-area convex enclosure problem,* European Journal of Operations Research, vol. 84, pp. 522-538, 1995.

[G96] R.B. Grinde, *Applying minimal enclosure tools to the pattern layout problem, Working paper*, University of New Hampshire, 1996.

[H99] M. Hifi, *The strip cutting/packing problem: incremental substrip algorithms-based heuristics,* Pesquisa Operacional, Special Issue on Cutting and Packing Problems, vol. 19, No 2, pp. 169-188, 1999.

[H98] M. Hifi, *Exact algorithms for the guillotine strip cutting/packing problem,* Computers and Operations Research, vol. 25, pp. 925-940, 1998.

[HT01] R. Hopper and B.C.H. Turton, *A review of the application of meta-heuristic algorithms to 2D strip packing problems,* to appear in Artificial Intelligence Review, 2001.

[HT99] R. Hopper and B.C.H. Turton, *An empirical investigation of meta-heuristics and heuristic algorithms for a 2D packing problem,* European Journal of Operational Research, vol. 128, pp. 34-57, 2000.

[J96] S. Jakobs, *On the genetic algorithms for the packing of polygons,* European Journal of Operational Research, vol. 88, pp. 165-181, 1996.

[LC97] K.K. Lai and W.M., *An evolutionary algorithm for the rectangular cutting stock problem,* International Journal of Industrial Engineering, vol. 4, pp. 130-139, 1997.

[LYC99] T.W. Leung, C.H. Yung and C.K. Chan, *Applications of genetic algorithm and simulated annealing to the 2-dimensional non-guillotine cutting stock problem*, IFORS '99, Beijing China.

[LM95] Z. Li and V. Milenkovic, *Compaction and separation algorithms for non-convex polygons and their applications,* European Journal of Operational Research, vol. 84, pp. 539-561, 1995.

[LT99] 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, pp. 413-420, 1999.

[MDL91] V. Milenkovic, K. Daniels and Z. Li, *Automatic marker making*, Proceedings of the Third Canadian Conference on Computational Geometry, Simon Fraser University, Vancouver, B.C., August 6-10, T. Shermer, Ed, pp. 243-246, 1991.

[PD92] P. Poshyanonda and C.H. Dagli, *A hybrid approach to composite stock cutting: Neural networks and genetic algorithms*, in: M. Jamshidi, R. Lumia, J. Mullins, M. Shahinpoor, (Eds.), Robotics and Manufacturing: Recent Trends in Research, Education, and Applications, vol. 4, 1992.

[RCO99] C. Ribeiro, M.A. Carravilla and J.F. Oliveira, *Applying constraint logic programming to the resolution of nesting problems*, Pesquisa Operacional, Special Issue on Cutting and Packing Problems, vol. 19, No 2, pp. 239-247, 1999.

[SY98] Y.G. Stoyan and G.N. Yaskov, *Mathematical model and solution method of optimization problem of placement of rectangles and circles taking into account special constraints,* International Transactions in Operational Research, vol. 5, No 1, pp. 45-57, 1998.

[SN99] Y.G. Stoyan and M.V. Novozhilova, *Non-guillotine placement of rectangles into a strip of given width,* Pesquisa Operacional, Special Issue on Cutting and Packing Problems, vol. 19, No 2, pp. 189-210, 1999.

Authors addresses:
Mhand Hifi
+ CERMSEM-CNRS UMR 8095, Maison des Sciences Economiques
Université Paris 1 Panthéon-Sorbonne
106-112, boulevard de l'Hôpital, 75647 Paris cedex 13, France.
+ Membre extérieur du PR*i*SM-CNRS UMR 8636,

Université de Versailles-Saint Quentin en Yvelines,
45 avenue des Etats-Unis, 78035 Versailles cedex, France.
mailto:Mhand.Hifi@univ-paris1.fr

Rym M'Hallah
+ Department of Quantitative Methods,
Institut Supérieur de Gestion de Sousse, B.P. 763, Sousse 4000, Tunisia.
+ Research Institution for Computer Science and Telecommunications,
Parc technologique des Communications, Route de Raoued km 3.5, 2083 Ariana, Tunisia.
mailto:Rym.Mhallah@univ-reims.fr

.