Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 8703782, 12 pages
http://dx.doi.org/10.1155/2016/8703782

Research Article

An Efficient Heuristic Approach for Irregular Cutting Stock Problem in Ship Building Industry

Yan-xin Xu

Department of Automation, Shanghai JiaoTong University and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China

Correspondence should be addressed to Yan-xin Xu; iamxuyanxin@sina.com

Received 20 June 2016; Revised 25 August 2016; Accepted 29 August 2016

Academic Editor: Mohammed Nouari

This paper presents an efficient approach for solving a real two-dimensional irregular cutting stock problem in ship building industry. Cutting stock problem is a common cutting and packing problem that arises in a variety of industrial applications. A modification of selection heuristic Exact Fit is applied in our research. In the case referring to irregular shapes, a placement heuristics is more important to construct a complete solution. A placement heuristic relating to bottom-left-fill is presented. We evaluate the proposed approach using generated instance only with convex shapes in literatures and some instances with nonconvex shapes based on real problem from ship building industry. The results demonstrate that the effectiveness and efficiency of the proposed approach are significantly better than some conventional heuristics.

1. Introduction

Cutting and packing problem reflects the large application scope, such as ship building, wood, shoe, garment, steel, and glass manufacturing, packing on shelves or truck beds in transportation and warehousing, and the paging of articles in newspapers [1, 2]. A wide variety of investigations in literatures exist into the two-dimensional cutting and packing problems. The basic idea is to find an arrangement of a finite number of items to cut from stocks or pack inside bins. The main goal is to minimize the number of stocks or bins and maximize the utilization of materials. Figure 1 provides an example of a layout from iron and steel industry.

Cutting and packing problems include knapsack problem, strip packing, bin packing, and cutting stock problem. In this paper, we focus on the two-dimensional irregular cutting stock problem. This problem is to cut a finite set of two-dimensional irregular items from a couple of stocks such that the number of stocks is minimized. Wäscher et al. (2007) proposed a useful complete problem typology of cutting and packing problems on the basis of Dychoff (1990) [3–5]. According to the typology of Wäscher, this is Two-Dimensional Single Stock-Size Cutting Stock Problems (2DCSP).

It can be described as follows: Given a set $L = (i_1, i_2, \ldots, i_n)$ of items, size of each one $s(i_j) \in (0, A_{W\times L}]$, to be cut from a set of rectangular cutting stock sheets (objects) of size $A_{W\times L}$ (length L and width W), the 2DCSP consist of finding cutting patterns that the solution minimizes the number of stock sheets used. Typical assumptions areas follows:

(1) All items must be within the stock sheet.

(2) Items must not overlap with each other.

This paper is organized as follows. A brief review of previous work in the field is presented in Section 2. The modification of Exact Fit selection heuristic and BLF placement heuristic is presented in detail and other common heuristics in literatures are introduced in Section 3. Section 4 gives experimental results on generated instances and real-world instances from ship building industry. In Section 5, the research is concluded and possible issues for future work are suggested.

2. Literature Review

A number of researches on rectangular items in 2DCSP have been done in literatures. The work on packing problems with

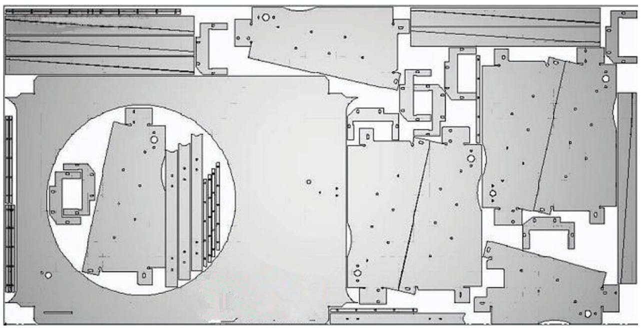

Figure 1: An example layout from iron and steel industry.

irregular items mainly focused on strip packing problem (SPP), which is a variant of 2D bin packing problem. In SPP, only one large strip object with fixed width and variable length is used to pack items and the target is to minimize the length of the strip object after all items are allocated. Hopper and Turton reviewed the approaches developed to solve 2D strip packing problem with metaheuristic algorithms [6]. Burke et al. presented a new bottom-left-fill heuristic to solve irregular strip packing problem. They packed shapes with tradition line representation and utilized hill climbing and Tabu search methods [7]. Júnior et al. also proposed an approach to solve irregular strip packing problem with nonconvex shapes. They integrate a greedy bottom-left as placement rule and a genetic algorithm as metaheuristic searching engine. They still utilized no-fit polygon to reduce geometric complexity [8]. However, the researches devoted to 2D irregular cutting stock problem have not received as much attention as SPP. Han et al. proposed a constructive heuristic to solve 2D irregular (convex) bin packing problem with guillotine constraints [9]. López-Camacho et al. presented an adapted heuristic to deal with irregular (convex) bin packing problem [10]. As far as we know, the 2D irregular (nonconvex) cutting stock problem is seldom mentioned in previous researches. In fact, many practical applications, such as ship building, shoe, and garment, widely concern irregular nonconvex items. Although some heuristics and strategies for convex shape are probably similar to nonconvex one, after all, they are not the same problem. We cannot directly compare the results with them. As a result, it is necessary to research approaches for nonconvex shapes.

A wide variety of approaches for irregular 2DCSP have been presented in literatures. They consist of optimization approaches, heuristic and metaheuristic, and the combination of the abovementioned approaches. The 2DCSP has been shown to be NP-hard. Due to the NP-complete nature of the problem, implementations of global optimization techniques, such as linear programming and column generation, have restrictions in applications involving an astronomical number of variables. Therefore, published methods mostly focused on heuristic and metaheuristic. They intended to find an acceptable approximate solution as soon as possible.

In 1DCSP, we only pay attention to the sequence of items to be packed, instead of the position in detail. However, according to 2DCSP, additional difficulty, where a particular item should be placed inside the object, is introduced. Thus, except for selection heuristic, placement heuristics must be considered in 2DCSP. Selection heuristic confirms the sequence of items to be placed. Then, placement heuristic is responsible for the specific location of items on objects according to some criteria. Many heuristics have been studied in literatures and they have respectively their own superiority in some instances. Among these solution approaches, one key strategy for representing and searching the solution space is to represent the solution as an ordered list of items and apply a placement rule to construct the solution, whereas it is proved that a good initial permutation of items is essential and beneficial for good packing performance. Therefore, many criteria for finding good permutation are presented in literatures. 81 different static ordering rules are studied in Abdelhafiez [11]. Among the group of rules, it is found that decreasing area and decreasing length rules give efficient results within all problem sets. Furthermore, some dynamic searching techniques are also used to select the next item [12, 13].

Metaheuristics are general frameworks for heuristics in solving combinatorial optimization problems. As packing tasks have very large search spaces, the size of the search space needs to be reduced in order to decrease the cost of computing time. Hence, recent literatures encourage the use of metaheuristic as searching engine. A number of metaheuristic approaches, such as *simulated annealing*, *Tabu search*, *neural networks*, and *genetic* and *particle swarm optimization algorithms*, have been applied in cutting and packing problem [14–16]. For example, Pinheiro et al. presented a random-key genetic algorithm to solve irregular nesting problem. They also used bottom-left placement rule and shrinking algorithm to improve partial solutions. Their approach integrated these algorithms to get good performance [17]. Sato et al. focused on irregular nesting problem. They utilized a pairwise placement strategy to allocate items in exact placements. Simulated annealing algorithm was used to control placement sequence and search solution space [18].

In addition to the approaches mentioned above, a combination of some techniques may produce a better solution to the problem, since the advantage of each technique can be utilized and the disadvantage may be overcome. Poshyanonda et al. combined a neural network in the pattern generation phase and a linear programming model to find the solution [19]. A hybrid algorithm is presented in Gomes and Oliveira [2] for solving irregular SPP. In their research, simulated annealing guided the search over the solution space, and linear programming models generated neighborhoods during the search process.

When studies focus on packing task with irregular shapes, a potential barrier to academic research is the lack of tools to assimilate the geometric complexity, compared with regular graphics. Recent literatures present some effective approaches to generate potential placement allocation without overlap, when the next item is to be packed on objects. A detailed tutorial about the geometry of nesting problems is given in Bennell and Oliveira [20]. These approaches include *no-fit polygon (NFP)*, *Phi Function*, *rectangle enclosure*, and *grid approximation* [21–25], wherein NFP is definitely the most successful one, which can be gained in two ways: *orbital* *sliding* and *Minkowski Sum* [26–28]. But high cost of computational time and robustness of generation for nonconvex irregular graphics restrict the utilization of NPF in some researches. The other approaches also get some attention in literatures.

3. Methodology

There are generally two modes of operation in 2DCSP: online and offline. Our research focuses on offline mode, in which the entire list of items' shapes to be packed is known in advance, and then places the items' shapes in a sequential manner by deciding which item should be placed next and in which orientation. In online mode, the dimensions of the next item's shape to be packed only become known once the current item has been packed. CSP has also a few variations depending on rotations of items: rotations of any angle are allowed; a finite number of angles are allowed; and rotations of any angle are not allowed. Our research deals with the second case and 4 orientations are allowed.

3.1. The Prelayout Phase. The operation of our approach is divided into two phases: the prelayout phase and the layout phase. In the prelayout phase, the initial sequence and orientation of items are confirmed on the basis of some criteria. According to the study of Abdelhafiez [11], the items are firstly sorted according to the nonincreasing order of their area. In the case of equal area of items, their sequence is decided in nonincreasing order by their length. The initial orientation of each item is determined by their MRE (Minimum Rectangular Enclosure, which is applied in Jakobs [23]). Each irregular-shaped item can be accommodated by its rectangular enclosure. The size of rectangular enclosure of each item is unequal with different orientation. The MRE defines a lower bound of the area of rectangular enclosure. It is worth mentioning that this approach does not replace the items by their MREs. Rather, it uses MREs as additional information for orienting the items.

Rectangular enclosures of 90 different orientations (0–89) of each item are obtained so as to find the minimum one. Obviously, the angle to produce MRE is selected as the initial orientation of the item. Each item is represented by a list of vertex coordinates. The rotated item is obtained as follows:

$$\begin{bmatrix} x_{k\theta} \\ y_{k\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix}, \tag{1}$$

where (x_k, y_k) denotes the coordinates of the kth vertex of the original item and $(x_{k\theta}, y_{k\theta})$ denotes the coordinates of the kth vertex of the rotated item. Accordingly, the area of rectangular enclosure of the angle Θ is computed as follows:

$$\begin{aligned} \mathrm{RC}_\theta = {} & \left[\max|_{k=1}^{M}\{x_{k\theta}\} - \min|_{k=1}^{M}\{x_{k\theta}\}\right] \\ & * \left[\max|_{k=1}^{M}\{y_{k\theta}\} - \min|_{k=1}^{M}\{y_{k\theta}\}\right]. \end{aligned} \tag{2}$$

Among the 90 different rectangular enclosures, the size of MRE is

$$\mathrm{RC}_{\min} = \min|_0^{89} \mathrm{RC}_\theta. \tag{3}$$

3.2. Layout Phase. In layout phase, a modification of Exact Fit heuristic is applied as selection heuristic. The placement heuristic employs a extension of bottom-left-fill heuristic. The placement of the items follows a single-pass placement strategy and takes place in a sequential manner. As for geometric computation, we adopt grid approximation to solve the question of overlap. These methods mentioned above are, respectively, introduced as follows.

3.2.1. The Proposed Selection Heuristic for Irregular 2DCSP. Originally, bin packing problem was defined as one-dimensional problem by Garey and Johnson [5]. Many heuristics for 1D problem are introduced in literatures. These heuristics are still able to be applied for 2D case. In this paper, the proposed selection heuristic is based on *Exact Fit* mentioned in Kos and Duhovnik [29], where Exact Fit (EF) is described as the following algorithm:

(1) Take a new bin and fill it with elements until the bin is over 1/3 full.

(2) Try to fill the rest of the bin exactly with single element or combination of the two or three elements.

(3) If there is no such combination that will fill bin exactly, relax the problem with decreasing the bin size for one unit and repeat procedure 2 until success.

(4) Repeat the whole procedure, until there are unassigned elements available [29].

In our research, the proposed EF heuristic is described as follows.

The items are firstly ordered in decreasing manner by area. Then, put them into a stock sheet until it is over 1/3 full. After that, try to find one single item or combinations of two or three items to fill the rest of the stock sheet exactly. If there is no such single item or combinations that meet the requirement, relax the problem with decreasing the size of stock sheet to p $(0 < p < 1, p \in \mathrm{Q})$ of its capacity. Then, repeat the abovementioned finding procedure. If it fails, decrease size to $2p$ of its capacity and repeat the finding procedure above, and so on. The outline of the Exact Fit heuristic for 2D case is shown with pseudocode in Algorithm 1. And pseudocodes in Algorithms 2, 3, and 4, respectively, show the process of trying to find one, or combination of two or three items.

It can be seen that the proposed EF carries out some modification and extension on the original EF in order to optimize running time, because of geometric complexity of 2D case against 1D problem. Firstly, one unit is replaced with p of the total area in procedure (3) of original EF, since it is not feasible for 2D case with one unit. The introduction of p is described in Section 3.2.2 in detail. Secondly, the area of combination of items is compared with the available free area of stock sheet before they are placed. If the former is less than the latter, then we try to place them. Otherwise, the combination is abandoned. Thirdly, with regard to the combination of items, only after the first item is placed, the next one is considered, and so on. If all of the next items do not meet the requirement, the first placed one is removed and another group is taken into account.

Input: Set of items sorted by decreasing area I; Set of rotations θ for items of in I; Dimensions of Stock Sheet (L, W).
Output: Solution C (Cutting).
(1.1) *percentage* = 0; p [increment of allowed percentage]
(1.2) **while** $I \neq \{\}$ **do**
(1.3) fill the stock sheet with items until the stock sheet is over 1/3 full
(1.4) record every item that does not fit
(1.5) call Algorithm 2
(1.6) **if** the free area up to p after a item could be placed **then**
(1.7) reset $p = 0$ and go to step (1.5)
(1.8) call Algorithm 3
(1.9) **if** the free area up to p after a combination of 2 items could be placed **then**
(1.10) reset $p = 0$ and go to step (1.8)
(1.11) call Algorithm 4
(1.12) **if** the free area up to p after a combination of 3 items could be placed **then**
(1.13) reset $p = 0$ and go to step (1.11)
(1.14) **if** all the possible combinations of 1, 2 or 3 items could not be placed
AND *percentage* < free area of stock sheet **then**
(1.15) *percentage* = *percentage* + p
(1.16) **else** open a new stock sheet

Algorithm 1: The Exact Fit heuristic.

(2.1) **for each** $i \in I$ **do**
(2.2) **if** free area of stock sheet − area of the item i > percentage **then**
(2.3) **break**
(2.4) **if** area of the item i > free area of stock sheet **OR** the item i does not fit **then**
(2.5) **continue**
(2.6) **if** this item could be placed **then**
(2.7) **return**
(2.8) **else** record the item that could not be placed

Algorithm 2: One item.

(3.1) **for** each $i \in I$ **do**
(3.2) **if** free area of stock sheet − area of the item i − largest item's area > percentage **then**
(3.3) **break**
(3.4) **if** the item i does not fit **OR**
area of the item i + smallest item's area > free area of stock sheet **then**
(3.5) **continue**
(3.6) **if** this item could not be placed **then**
(3.7) record the item that could not be placed
(3.8) **else** {select a second item $j \in I \setminus i$}
(3.9) **for** each $j \in I \setminus i$ **do**
(11) **if** free area of stock sheet − area of the item i and j > percentage **then**
(12) **break**
(13) **if** item j or the combination of item i and j do not fit
OR area of item i and j > free area of stock sheet **then**
(14) **continue**
(15) **if** the item could be placed **then**
(17) **return**
(18) **else** remove the first item I **AND** record the combination of the 2 items that does not fit

Algorithm 3: Combination of 2 items.

(4.1) **for** each $i \in I$ **do**
(4.2) **if** free area of stock sheet − area of the item i − area of the two largest item's area > percentage **then**
(4.3) **break**
(4.4) **if** item i does not fit **OR** area of item i + 2 smallest items' area > free area of stock sheet **then**
(4.5) **continue**
(4.6) **if** this item could not be placed **then**
(4.7) record the item that could not be placed
(4.8) **else** {select a second item $j \in I \setminus i$}
(4.9) **for** $j \in I \setminus i$ **do**
(4.10) **if** free area of stock sheet − area of item i and j − area of largest item > percentage **then**
(4.11) **break**
(4.12) **if** item j or the combination of item i and j do not fit **OR** area of item i and j + area of smallest item > free area of stock sheet **then**
(4.13) **continue**
(4.14) **if** this item could not be placed **then**
(4.15) remove the first item I **AND** record the combination of item i and j that does not fit
(4.16) **else** {select a third item $k \in I \setminus \{i, j\}$}
(4.17) **for** each $k \in I \setminus \{i, j\}$ **do**
(4.18) **if** free area of stock sheet − area of item i, j and k > percentage **then**
(4.19) **break**
(4.20) **if** any item, or combination of two items or combination of the three items does not fit **OR** area of item i, j and k > free area of stock sheet **then**
(4.21) **continue**
(4.22) **if** the item could be placed **then**
(4.23) **return**
(4.24) **else** remove first 2 items **AND** record the combination of 3 items that does not fit

Algorithm 4: Combinations of 3 items.

In addition, Exact Fit places items only in one open stock sheet at a time. It is similar to Best Fit in placing procedure. It is worth noting that combination of the same items in different order can get quite different packing pattern. Some of them may get solution with good performance, while some probably get infeasible solution. As a result, for an item which cannot be placed in a stock sheet originally, a slim chance exists, that it might fit this object after the other items have been placed. In spite of increasing running time, the possibility in the abovementioned implementation is still required to be considered in our research, since material utilization gets more attention than running time in ship building industry. Hence, a record of items that conform to the abovementioned case is kept until they need to be cleaned up.

Finally, it is indispensable to keep some information about placement procedure so as to avoid unnecessary operations and reduce computation. For the case of one item, the information about all of the placed first items of combinations for each stock sheet is recorded to avoid them appearing in other different groups. For the case of two items, some ordered combinations of two items or the first two items of group with three items cannot fit some stock sheet. The information about the order of the two items needs to be kept, so do the stock sheets which cannot place these two items. For the case of three items, it is similar to the case of two ones that the information about the order of three items and the corresponding stock sheets is recorded.

3.2.2. The Tool for Geometrical Computation. In our research, grid approximation is used to test overlaps among items and deal with geometrical complexity of convex and nonconvex irregular shapes. It is a digitized representation technique and proposed by Poshyanonda and Dagli (2004) and Dagli and Hajakbari (1990) [24, 30].

In the beginning, each item is represented by a list of vertex coordinates ordered counterclockwise, $[(x_1, y_1), \dots, (x_N, y_N)]$, where N is the number of vertexes. For placement heuristic, items must be packed without overlap between each other and the objects. Consequently, degree of overlap on the stock sheet should be tested. During the packing process, MRE of each item is digitized as a matrix of size $K \times L$ to replace this item. The values of K and L are determined based on the desired precision for the computations. The stock sheet is also represented by a matrix $S(t)$, where $S_{ij}(t)$ is the value at location (i, j) on the stock sheet after the first t items have been allocated. $S_{ij}(t)$ is computed as follows:

$$S_{ij}(t) = S_{ij}(t-1) + P_{ij}(t), \quad S_{ij}(0) = 0, \tag{4}$$

where $S_{ij}(t-1)$ is the value of the matrix at the location (i, j) on the stock sheet after the first $t-1$ items have been allocated; $S_{ij}(0)$ is the value of the matrix at the location (i, j) on the stock sheet at the beginning of the packing process; $P_{ij}(t)$ is the value of the matrix representation at location (i, j) of the tth item to be allocated on the stock sheet.

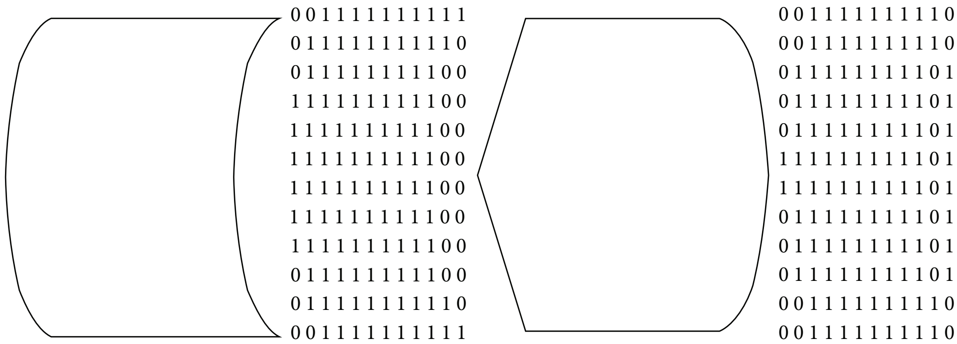

FIGURE 2: Binary representations.

In this paper, the grid approximation approach is that of Wong et al. (2009) [25]. This technique encloses each item with an imaginary rectangle to obtain the reference points in nesting procedure. Then, these particular rectangles are divided into a number of uniform grids, which is called pixel. Therefore, grid approximation is also named as pixel representation. According to these grids, the value of a pixel is "1" when the material of the sheet is occupied. Otherwise, the value of the pixel is "0." An example of matrix representation is given in Figure 2.

The item with a two-dimensional matrix of size $A_W^{(k)} \times A_L^{(k)}$ is represented as follows:

$$A^{(k)} = \begin{pmatrix} a_{11}^{(k)} & a_{12}^{(k)} & \cdots & a_{1A_L^{(k)}}^{(k)} \\ a_{21}^{(k)} & a_{22}^{(k)} & \cdots & a_{2A_L^{(k)}}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{A_W^{(k)}1}^{(k)} & a_{A_W^{(k)}1}^{(k)} & \cdots & a_{A_W^{(k)}A_L^{(k)}}^{(k)} \end{pmatrix}, \tag{5}$$

where

$$A_W^{(k)} = \frac{P_W^{(k)}}{R}, \tag{6}$$

$$A_L^{(i)} = \frac{P_L^{(k)}}{R}. \tag{7}$$

For each entry,

$$a_{i,j}^{(k)} = \begin{cases} 1, & \text{if pixel } (i, j) \text{ is occupied} \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where $P_L^{(k)}$ and $P_W^{(k)}$ denote the length and the width of an enclosing rectangle corresponding to the item p_k. R denotes the square side of a pixel (in this paper, $R = 0.085$ mm). Furthermore, the proposed heuristic allows each packing item to be rotated in several orientations, such as 0°, 90°, 180°, and 270°. The matrix representation of rotated item can be obtained by simply modifying the matrix of the original item shown above. Then, the matrix of the rotated item with 180° becomes

$$A_{\text{rotate}}^{(k)} = \begin{pmatrix} a_{A_W^{(k)}A_L^{(k)}}^{(k)} & \cdots & a_{A_W^{(k)}2}^{(k)} & a_{A_W^{(k)}1}^{(k)} \\ \vdots & \ddots & \vdots & \vdots \\ a_{2A_L^{(k)}}^{(k)} & \cdots & a_{22}^{(k)} & a_{21}^{(k)} \\ a_{1A_L^{(k)}}^{(k)} & \cdots & a_{12}^{(k)} & a_{11}^{(k)} \end{pmatrix}_{A_W^{(k)} \times A_L^{(k)}}. \tag{9}$$

The stock sheets are also able to be digitized as a finite number of equal-size pixels of size R^2 like items. For example, the stock sheet with the length L and the width W is characterized by a matrix U of size $U_W \times U_L$ as follows:

$$U = \left[u_{i,j}\right]_{U_W \times U_L}, \tag{10}$$

where

$$U_W = \frac{W}{R}, \quad U_L = \frac{L}{R}. \tag{11}$$

For each entry,

$$u_{i,j} = \begin{cases} 1, & \text{if pixel } (i, j) \text{ is occupied} \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

If the value of a pixel in the matrix of stock sheet is greater than "1," it means that there is an overlap. Despite the high speed of overlap test by grid approximation, compared with other tools of geometric computations, a large memory for representation scheme is still requisite. Consequently, only required matrices are generated in the procedure of overlap test.

3.2.3. The Proposed Placement Heuristic for Irregular 2DCSP. After the selections of item and stock sheet are completed, it is the turn of placement heuristic to allocate items onto stock sheet. Many placement strategies are presented in literatures. These different placement rules may generate quite different packing patterns even for the same problem. Some of them

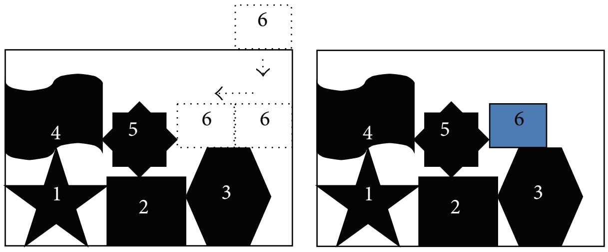

Figure 3: A layout example of BL routine.

outperform the other rules in some particular cases, while sometimes they get poor performance in other cases. A most widely known placement heuristic is bottom-left (BL), which is firstly proposed in Jakobs [23]. Then, BL is widely applied in many approaches for solving regular and irregular cutting and packing problem. Besides, several improvement and modification versions of BL are presented in literatures, such as improved BL [31] and bottom-left-fill (BLF) [32]. In this paper, the proposed placement heuristic is based on BLF. Then, the brief introduction about these heuristic is as follows.

(1) Bottom-Left (BL). Each item starts from the top-right corner and it slides as far as possible to the bottom and then as far as possible to the left of the stock sheet. In the process of moving, the item cannot overlap with any other placed item and the boundaries of stock sheet. These successive vertical and horizontal movement operations are repeated until no other movement is possible. Then, a valid position is found and locked as the final position of the item on the stock sheet. A layout example of BL heuristic is shown in Figure 3. The permutation of items is (1, 2, 3, 4, 5, 6). The performance of this heuristic greatly depends on the initial ordering of the items.

(2) Bottom-Left-Fill (BLF). The strategy of BLF is choosing bottom-left position among a set of positions or nodes. The nodes are obtained by projecting the left and bottom sides of each positioned item onto the right and top sides of already-positioned ones. Since the generation of the layout is based on the allocation of the lowest sufficiently large region in the partial layout rather than on a series of BL moves, it is capable of filling existing gaps in packing pattern [6]. BLF is originally applied in rectangular bin packing problem. Since the items are irregular shapes in this paper, these bottom-left nodes are those BL positions of the item's MRE. Then, the item still needs to move in the manner of BL until the final position is found. According to the same items and permutation in Figure 3, the layout of BLF is shown in Figure 4. These blue dots are the BL positions above. The placing blue triangle will try all of them from bottom to top. If there are several feasible locations as candidates, the lowest one is the chosen position. As shown in Figure 4, the locations of two red dots are the candidates and the right one is lower than the left one. Hence, we choose the right one. After allocating the item, we try to move it until it finds final position. It is worth noting that the purple dot is not a feasible position and the item finally is not placed on it.

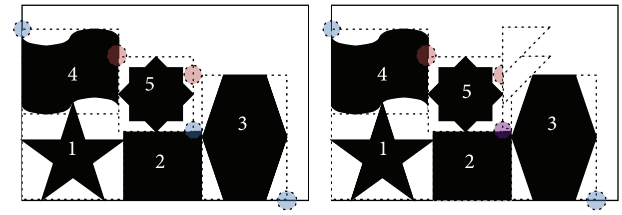

Figure 4: A layout example of BLF heuristic.

(3) Bottom-Left-Fill with Maximal Connected Region (BLFM). This is a modification of the BLF heuristic. After items are placed, the free connected area is called the available connected region in this research. Consequently, among these candidate nodes, the BLFM selects the best node, on which maximal connected region is obtained. In other words, the area of the rectangular enclosure of the placed items is minimal. A layout of BLFM is shown in Figure 5. There are three candidate BL positions marked with blue dots. After the item is placed on the rule of BLF, the area of MRE is calculated and the position with the smallest MRE is the final one for the item, as shown in Figure 5(a).

As we know, different angles of irregular items may generate quite different packing pattern on account of geometrical complexity of irregular shapes. As a result, for BLF and BLFM, the items need be rotated in several orientations, such as 0°, 90°, 180°, and 270°, when they are placed on candidate nodes. The rotation with best performance is the final orientation of item. Of course, the premise is that this layout must be a feasible solution. Otherwise, this rotation is discarded. If no positions can be chosen as BL positions, rotation is unnecessary. Because the overlap tests in placement procedure are time-consuming, we avoid unnecessary operation as far as possible.

3.2.4. The Other Selection Heuristics and Outline of Our Algorithm. Except Exact Fit heuristic, there are several well-known selection heuristics, such as First Fit (FF), First Fit Decreasing (FFD), First Fit Increasing (FFI), Best Fit (BF), and Best Fit Decreasing (BFD). These heuristics are also used for comparison against the proposed heuristic in the paper. The brief introduction about these heuristic is as follows.

(1) First Fit (FF). The packed stock sheets are called open and ordered from left to right. They are considered as candidates for the next item to be packed. When the next item is to be placed, the open stock sheets are checked in sequence so as to find the first one where this item fits. If no open stock sheet can fit it, then a new stock sheet is opened to allocate this item.

(2) First Fit Decreasing (FFD). The items are ordered in decreasing manner according to area. Then, the item is placed according to FF.

(3) First Fit Increasing (FFI). The items are ordered in increasing manner according to area. Then, the item is placed according to FF.

(4) Best Fit (BF). All the open stock sheets are checked in sequence for the sake of finding the one with minimum

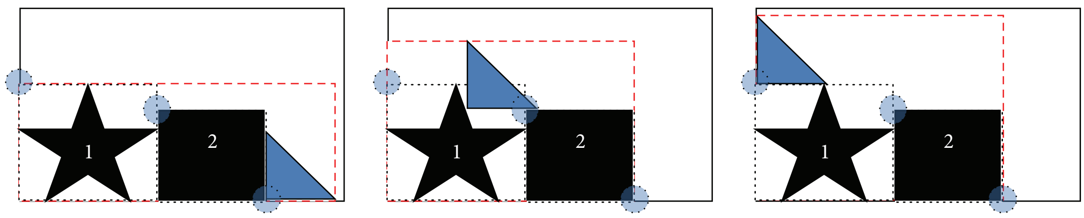

Figure 5: A layout example of BLFM heuristic.

Table 1: Information about the instances under consideration.

Type	Size of stock sheet	Num. of items	Num. of instances	Rotations
1	1500 × 1500	84	40	(0, 90, 180, 270)
2	1500 × 1500	54	40	(0, 90, 180, 270)
3	1500 × 1500	90	40	(0, 90, 180, 270)
4	1500 × 1500	81	40	(0, 90, 180, 270)
5	1500 × 1500	60	40	(0, 90, 180, 270)
6	1500 × 1500	90	40	(0, 90)
7	1500 × 1500	90	40	(0, 90, 180, 270)
8	1500 × 1500	42	40	(0, 90, 180, 270)
9	1500 × 1500	45	40	(0, 90, 180, 270)
10	1500 × 1500	54	40	(0, 90, 180, 270)

available free area or minimum waste after this item is placed. If no open stock sheet can fit it, then a new stock sheet is opened to allocate this item.

(5) Best Fit Decreasing (BFD). The items are ordered in decreasing manner according to area. Then, the item is placed according to BF.

4. Performance Evaluation

In order to illustrate the performance of our approach, two sorts of instances are proposed in this research. The first sort of instances is randomly generated in accordance with some rules. The items in these instances are irregular convex polygons. The number of sides of these polygons is between 3 and 10. We generated 400 instances from 10 different types. Table 1 presents the information about these types of instances. The second sort of instances is based on real-life data from ship building industry. All algorithms are implemented and performed on a computer with processor Intel Core i5 3.0 GHz. The resolution ratios of the items and stock sheets are 300 PPI. Furthermore, for evaluating the solutions, we consider the ratio of sum of percentage of usage for each stock sheet and the total number of sheets used. The ratio is called packing density (PD):

$$\mathrm{PD} = \frac{\sum_{i=1}^{N} U_i}{N}. \tag{13}$$

As described in Section 3.2.1, the decreasing percentage p is an important parameter for EF heuristic. Regardless of whether the value of p is close to 1 or 0, computational cost will increase, while the placement efficiency is not high. Hence, it is significant to find a balance point to set p a value. Experimental results show that $p = 0.05$ is a reasonable balance. No matter whether $p < 0.05$ or $p > 0.05$, acceptable computational cost and good performance cannot be achieved at the same time. According to the introduction of EF in Section 3.2.1, a new bin needs to be filled until its fullness is over 1/3. We tend to give a variety of parameters to facilitate comparison, in order to obtain the best results. Thereupon, two different EF heuristics are proposed and, respectively, named $\mathrm{EF}_{1/4}$ and $\mathrm{EF}_{1/2}$, since their fullness demands are 1/4 and 1/2. For the sake of consistency, EF is called $\mathrm{EF}_{1/3}$ in our study. Thus, we obtain 8 different selection heuristics, coupled with the 5 heuristics introduced in Section 3.2.4. The 8 heuristics will be combined with 3 placement heuristics shown in Section 3.2.3.

A total of 10 types of instances were considered for computational tests, as shown in Table 1. Each row of this table has the following information: instance type, size of stock sheet, number of items, number of instances in this type, and set of allowable rotations for the items. It should be stressed in particular that the rotations of type 6 are only 0° and 90°, as the rectangularity of the items in these instances is close to 1. Rectangularity is a parameter index to measure the shape of polygon and its value is the ratio of its area and the area of its minimum rectangle enclosure. If it equals 1, it means that the polygon is just a rectangle. Thus, two rotations is enough for the instances of type 6 without redundant operation.

In the paper, all combinations of selection and placement heuristics (8 × 3 = 24) are employed to solve these instances. The average fitness for every combination is shown in Table 2. The best result for each type is in bold and italic font. The results indicate that the strategy of EF in 1D bin packing problem can also be applied to 2D irregular cutting stock problem. By comparing the results, we can find that better performance is got by $\mathrm{EF}_{1/3}$ and $\mathrm{EF}_{1/4}$ among these selection heuristics and the placement heuristics and BLFM

Table 2: Average PD of all the combinations.

Placement heuristic	Selection heuristic FF	FFD	FFI	BF	BFD	$EF_{1/4}$	$EF_{1/3}$	$EF_{1/2}$	Average
BL	0.336	0.411	0.291	0.337	0.412	***0.470***	0.461	0.418	0.392
BLF	0.428	0.552	0.341	0.426	0.553	0.571	***0.581***	0.554	0.501
BLFM	0.490	0.637	0.372	0.490	0.639	0.670	***0.672***	0.642	0.577
Average	0.418	0.533	0.335	0.418	0.535	0.570	***0/571***	0.538	0.490

Table 3: Average computational time of all the combinations (seconds).

Placement heuristic	Selection heuristic FF	FFD	FFI	BF	BFD	$EF_{1/4}$	$EF_{1/3}$	$EF_{1/2}$	Average
BL	15.3	15.2	15.2	15.4	15.3	32.9	32.9	32.8	39.0
BLF	3.6	3.5	3.5	3.7	3.7	21.8	21.8	21.7	14.4
BLFM	3.8	3.7	3.7	3.9	3.8	22.7	22.7	22.6	15.1

Table 4: Average PD of BLFM combined with all the selection heuristics.

Type	PD of selection heuristic FF	FFD	FFI	BF	BFD	$EF_{1/4}$	$EF_{1/3}$	$EF_{1/2}$
1	0.470	0.650	0.347	0.472	0.651	0.665	***0.667***	0.652
2	0.504	0.693	0.370	0.495	0.698	0.740	***0.752***	0.712
3	0.401	0.567	0.327	0.399	***0.568***	0.563	0.554	0.565
4	0.530	0.698	0.374	0.515	0.699	0.695	***0.707***	0.689
5	0.406	0.568	0.296	0.395	0.571	0.562	***0.578***	0.573
6	0.587	0.608	0.562	0.587	0.608	0.610	0.608	***0.615***
7	0.290	***0.401***	0.219	0.287	0.399	0.382	0.388	0.392
8	0.526	0.780	0.398	0.534	0.776	***0.812***	0.801	0.780
9	0.595	0.742	0.449	0.601	0.743	***0.918***	***0.918***	0.745
10	0.581	0.696	0.437	0.590	0.697	0.812	***0.813***	0.697
Average	0.489	0.640	0.378	0.488	0.641	0.676	***0.679***	0.642

outperforms the others no matter the selection heuristic. The combination of $EF_{1/3}$ and BLFM is undoubtedly the best among all the combinations.

With respect to the computational time, Table 3 shows a time comparison between the set of heuristics. It can be seen that heuristics in EF families consume more time than the other heuristics. The results are inevitable due to complex packing principles of EF families. Moreover, we can find that BL heuristic takes more time than the other two heuristics, because of its sliding mode in packing process. And the heuristics in FF families take similar time to the ones in BF families. The same cases happen in BLF families. Because production period is relatively long in ship building industry, while cost of materials is very high, utilization of material is paid more attention than computational time. However, we still give the results of computational time as reference.

In order to report the good performance of BLFM, the average PD of BLFM combined with all the selection heuristics is indicated in Table 4. We try our best to diversify the features of these polygons in these instances without loss of generality. Consequently, rectangularity of polygons in type 6 is close to 1. The average area of polygons in types 3 and 7 is much less than the area of stock sheet, and so on. Because of the diversity of these instances, we can compare the advantages and disadvantages of different heuristics for different types. It can be seen from the results that $EF_{1/2}$ outperforms the others for type 6, since they are too rectangle. And modifications of EF are defeated by the other five heuristics for types 3 and 7, as too many small items exist in these instances.

Although the calculation time has been partly reduced by recording some important information, the computational cost of EF + BLFM is still higher than the others. However, since the performance of EF + BLFM is better than the other combinations for most of all types of instances, the specificity of the shipbuilding industry determines the desirability of this combination.

Apart from the generated instances, another sort of instances is from real-life data in ship building industry. Ten types of instances from the industrial practice were adopted to demonstrate the effectiveness and efficiency of the proposed algorithm. Thirty-one shapes of items from these instances are shown in Figure 6.

The performance of the proposed approach is compared with the real-world manual nesting result and our proposed Grouping Particle Swarm Optimization (GPSO) algorithm

Table 5: Results obtained from real-life data.

Ins.	Num. of items	Size of stock sheet		PD of the approaches			Average computational time		
		L (mm)	W (mm)	GPSO	Real	EF + BLFM	GPSO	Real	EF + BLFM
1	400	4520	2170	0.661	0.631	***0.712***	71.3	102.3	82.4
2	420	5238	2170	0.766	0.755	***0.817***	71.6	101.7	82.7
3	300	4340	2170	0.601	0.550	***0.651***	50.7	80.6	61.8
4	480	4488	2170	0.554	0.499	***0.606***	72.5	103.4	83.6
5	500	5236	2170	0.640	0.598	***0.701***	72.8	103.5	83.9
6	900	11880	2170	0.426	0.376	***0.477***	174.1	204.2	185.4
7	1000	11880	2340	0.236	0.206	***0.287***	195.4	225.1	206.1
8	1000	11880	2430	0.402	0.352	***0.453***	195.2	224.9	206.3
9	880	11880	2470	0.515	0.465	***0.566***	173.9	204.8	184.8
10	850	11880	2170	0.456	0.398	***0.507***	172.5	202.6	183.2

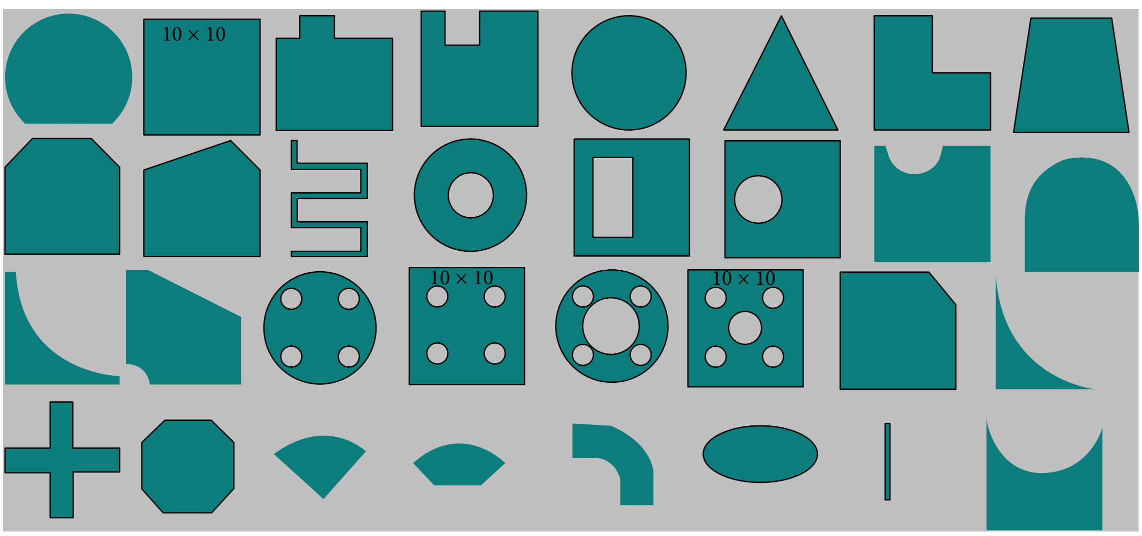

Figure 6: Examples of items from real-life data.

previously (this will be published on IJCA). Table 5 summarizes the best results of the instances from industrial practice, and the results obtained by the proposed approach are marked in bold and italic font. The PD of the packing pattern for the proposed approach, the real data, and the GPSO algorithm are, respectively, shown in the fifth, sixth, and seventh columns of Table 5. And the computational time of our approach, GPSO algorithm, and the real data are shown in the last three columns in Table 5.

The results indicate that the proposed methodology clearly outperforms the other approaches in all types of instances and the GPSO algorithm is better than manual nesting result in most cases. For some instances, in which the shapes with higher rectangularity and lower irregularity (a parameter corresponding to rectangularity) are more than others, the performances of GPSO and artificial operation are similar, but the speed of the former is faster. It is worth noting that manual nesting is not entirely artificial operation, which is aided by computer software. Manual operations are mainly minor readjustments and optimization on the basis of computer software. Thus, it is not as slow as in imagination. In terms of computational time, our proposed approach takes less time than real data but more than GPSO algorithm. Among them, GPSO is the fastest approach, while our approach gets the best performance. As mentioned above, shipyards focus more on material utilization than computational cost. Accordingly, such results demonstrate that our algorithm is able to effectively deal with irregular cutting stock problem of real world in ship building industry.

5. Conclusion

In this paper, we present the modification of Exact Fit and bottom-left-fill heuristic and, respectively, employ them as selection and placement heuristic to solve irregular 2DCSP. Moreover, grid approximation method is applied as a tool of geometric computation. It provides an easier way to detect whether overlap occurs, compared with other geometrical tools such as NFP. We adopt couple of combinations with different selection heuristic and placement heuristic, which are common heuristic in literatures. These combinations are presented in order to compare their solutions with ours in the research. We generate some representative instances to assess the performance of our approach. Some instances from real problem in ship building industry are also adopted. The results demonstrate that the presented approach performs well in most of instances and can deal with practical applications with complex shapes and industrial constraints.

Competing Interests

The author declares that they have no competing interests.

Acknowledgments

The work of this paper was supported by National Natural Science Foundation of China, nos. 61290323, 61203178, and 61304214.

References

[1] J. A. Bennell and J. F. Oliveira, "A tutorial in irregular shape packing problems," *Journal of the Operational Research Society*, vol. 60, no. 1, pp. S93–S105, 2009.

[2] A. M. Gomes and J. F. Oliveira, "A 2-exchange heuristic for nesting problems," *European Journal of Operational Research*, vol. 141, no. 2, pp. 359–370, 2002.

[3] 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.

[4] H. Dyckhoff, "A typology of cutting and packing problems," *European Journal of Operational Research*, vol. 44, no. 2, pp. 145–159, 1990.

[5] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, NY, USA, 1979.

[6] E. Hopper and B. C. H. Turton, "A review of the application of meta-heuristic algorithms to 2D strip packing problems," *Artificial Intelligence Review*, vol. 16, no. 4, pp. 257–300, 2001.

[7] E. K. Burke, R. S. R. Hellier, G. Kendall, and G. Whitwell, "A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem," *Operations Research*, vol. 54, no. 3, pp. 587–601, 2006.

[8] B. A. Júnior, P. R. Pinheiro, R. D. Saraiva, and P. G. C. D. Pinheiro, "Dealing with nonregular shapes packing," *Mathematical Problems in Engineering*, vol. 2014, Article ID 548957, 10 pages, 2014.

[9] W. Han, J. A. Bennell, X. Zhao, and X. Song, "Construction heuristics for two-dimensional irregular shape bin packing with guillotine constraints," *European Journal of Operational Research*, vol. 230, no. 3, pp. 495–504, 2013.

[10] E. López-Camacho, G. Ochoa, H. Terashima-Marín, and E. K. Burke, "An effective heuristic for the two-dimensional irregular bin packing problem," *Annals of Operations Research*, vol. 206, pp. 241–264, 2013.

[11] E. Abdelhafiez, "Efficient items-ordering rules in cutting stock problem," in *Proceedings of the 9th International Conference on Mechanical Design and Production Engineering MDP9*, Cairo, Egypt, January 2008.

[12] J. A. Bennell and J. F. Oliveira, "A tutorial in irregular shape packing problems," *Journal of the Operational Research Society*, vol. 60, no. 1, pp. S93–S105, 2009.

[13] J. A. Bennell and X. Song, "A beam search implementation for the irregular shape packing problem," *Journal of Heuristics*, vol. 16, no. 2, pp. 167–188, 2010.

[14] R. Alvarez-Valdes, F. Parreño, and J. M. Tamarit, "A tabu search algorithm for a two-dimensional non-guillotine cutting problem," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1167–1182, 2007.

[15] D. S. Liu, K. C. Tan, S. Y. Huang, C. K. Goh, and W. K. Ho, "On solving multiobjective bin packing problems using evolutionary particle swarm optimization," *European Journal of Operational Research*, vol. 190, no. 2, pp. 357–382, 2008.

[16] E. K. Burke, G. Kendall, and G. Whitwell, "A simulated annealing enhancement of the best-fit heuristic for the orthogonal stock-cutting problem," *INFORMS Journal on Computing*, vol. 21, no. 3, pp. 505–516, 2009.

[17] P. R. Pinheiro, B. Amaro Júnior, and R. D. Saraiva, "A random-key genetic algorithm for solving the nesting problem," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 11, pp. 1159–1165, 2016.

[18] A. K. Sato, T. D. C. Martins, and M. D. S. G. Tsuzuki, "A pairwise exact placement algorithm for the irregular nesting problem," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 11, pp. 1177–1189, 2016.

[19] P. Poshyanonda, A. Bahrami, and C. Dagli, "Two dimensional nesting problem: artificial neural network and optimization approach," in *Proceedings of the International Joint Conference of Neural Networks*, pp. IV-572–IV-577, Baltimore, Md, USA, June 1992.

[20] J. A. Bennell and J. F. Oliveira, "The geometry of nesting problems: a tutorial," *European Journal of Operational Research*, vol. 184, no. 2, pp. 397–415, 2008.

[21] J. Bennell, G. Scheithauer, Y. Stoyan, and T. Romanova, "Tools of mathematical modeling of arbitrary object packing problems," *Annals of Operations Research*, vol. 179, no. 1, pp. 343–368, 2010.

[22] N. Chernov, Y. Stoyan, and T. Romanova, "Mathematical model and efficient algorithms for object packing problem," *Computational Geometry*, vol. 43, no. 5, pp. 535–553, 2010.

[23] S. Jakobs, "On genetic algorithms for the packing of polygons," *European Journal of Operational Research*, vol. 88, no. 1, pp. 165–181, 1996.

[24] P. Poshyanonda and C. H. Dagli, "Genetic neuro-nester," *Journal of Intelligent Manufacturing*, vol. 15, no. 2, pp. 201–218, 2004.

[25] W. K. Wong, X. X. Wang, P. Y. Mok, S. Y. S. Leung, and C. K. Kwong, "Solving the two-dimensional irregular objects allocation problems by using a two-stage packing approach," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3489–3496, 2009.

[26] E. K. Burke, R. S. R. Hellier, G. Kendall, and G. Whitwell, "Complete and robust no-fit polygon generation for the irregular stock cutting problem," *European Journal of Operational Research*, vol. 179, no. 1, pp. 27–49, 2007.

[27] E. K. Burke, R. S. R. Hellier, G. Kendall, and G. Whitwell, "Irregular packing using the line and arc no-fit polygon," *Operations Research*, vol. 58, no. 4, pp. 948–970, 2010.

[28] J. A. Bennell and X. Song, "A comprehensive and robust procedure for obtaining the nofit polygon using Minkowski sums," *Computers & Operations Research*, vol. 35, no. 1, pp. 267–281, 2008.

[29] L. Kos and J. Duhovnik, "Rod cutting optimization with store utilization," in *Proceedings of the International Design Conference*, pp. 313–318, Dubrovnik, Croatia, 2000.

[30] C. H. Dagli and A. Hajakbari, "Simulated annealing approach for solving stock cutting problem," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 221–223, Los Angeles, Calif, USA, November 1990.

[31] 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, 1999.

[32] B. Chazelle, "Bottom-left bin-packing heuristic: an efficient implementation," *IEEE Transactions on Computers*, vol. C32, no. 8, pp. 697–707, 1983.