

Research Article

Hybridizing Beam Search with Tabu Search for the Irregular Packing Problem

Yunqing Rao , Peng Wang , and Qiang Luo 

The State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

Correspondence should be addressed to Peng Wang; huogewang@163.com and Qiang Luo; luoqiang0002@163.com

Received 13 July 2020; Revised 16 December 2020; Accepted 23 December 2020; Published 21 January 2021

Academic Editor: Neale R. Smith

Copyright © 2021 Yunqing Rao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The irregular packing problem involves arranging all the irregular pieces on a plate with the objective of maximizing the use of material. In this article, the layout is formed by the ordered sequence of the irregular pieces which is obtained by a hybrid search algorithm and where the order is decoded by a proposed placement principle. First, a novel no-fit-polygon (NFP) generator is introduced. Then, a placement principle is presented with the new NFP generator. Finally, a search algorithm hybridized with beam search (BS) and tabu search (TS) is proposed to search over the sequence. The numerical experiments with many benchmark problems show that the hybrid algorithm is an applicative and effective approach for solving the irregular packing problem. The hybrid algorithm can produce competitive solutions in less time than many other typical algorithms.

1. Introduction

The irregular packing problem arises in many manufacturing industries, such as metal cutting, textile, leather, and so on, with different industries having different objectives and constraints. In this article, the irregular packing problem is categorized as a 2D irregular open dimension problem (2DIP) by the typology of Wäscher et al. [1]. The problem is concerned with finding a good arrangement of all pieces, which may be arbitrary shape, on a plate. Usually, the objective is to maximize the material utilization or minimize the length of a plate with a fixed width.

The problem of finding an arrangement of pieces to cut from or pack inside larger objects is known as the cutting and packing problem, which is NP-hard by Garey and Johnson [2]. The Irregular Packing problem belongs to the more general class of combinatorial optimisation problems—the Cutting and Packing problems. The 2DIP problem is NP-complete, and as a result, the time consumption of the algorithm is very expensive when the scale of the problem is large. To obtain approximate solutions in a reasonable time, numerous methods for the irregular packing problem have been developed [3, 4].

There are two key strategies summarized in the published literature for handling this problem. The first one, called the legal placement method, represents the problem as a sequence of pieces and applying a placement rule to construct the solution. The second one, named the relaxed placement method, considers the problem as building a layout on the plate and moves pieces within the layout.

The legal placement method never allows the overlap to happen during the packing process. The two critical points of these methods are the placement sequence of pieces and the placement strategy. Art [5] implemented a placement strategy called BL (bottom-left) and using the multiple factors weighted to place the pieces. Dowsland and Dowsland [6] developed this strategy by permitting unpacked pieces to jump over the packed pieces to fill holes. Dowsland et al. [7] also proposed an algorithm where the edges of the plate width were alternated when searching for positions instead of packing pieces from the bottom to the top. Gomes and Oliveira [8] obtained the effective BL position by calculating intersections between the NFP of the already packed pieces and the next piece to

be packed. Oliveira et al. [9] put forward a famous placement strategy in a sophisticated constructive heuristic called TOPOS. The solution is produced from a floating position, not the BL position. What's more, both the next piece to be packed and its placement point are determined by three criteria: (a) minimum area of the rectangular enclosure; (b) minimum length of the rectangular enclosure; (c) maximum overlaps between the rectangular enclosures of the two pieces.

The other critical point of the legal placement method is the sequence of pieces. Dowsland et al. [10] proposed an algorithm that adopts a fixed sequence to place pieces. The sequence can be generated in a random way or simply by sorting the pieces by the area or length. Babu and Babu [11] implemented a genetic algorithm to search over the sequence of pieces. Oliveira et al. [9] selected the next piece dynamically by their TOPOS algorithm. Gomes and Oliveira [8] introduced a 2-swap heuristic for handling the sequence of pieces. Burke et al. [12] combined their new BLF(bottom-left-fill) heuristic algorithm with both hill climbing and tabu search, where the sequence was searched by randomly selecting one operator from four operators (an insert move, a pairwise swap, a three-way swap, and an n-way swap). The beam search algorithm was presented by Bennell and Song [13]; computational results showed that some best results published in the literature could be improved by their beam search algorithm. Sato et al. [14] proposed simulated annealing that searched for feasible layouts to be packed in a container with fixed dimensions. Burke et al. [15] extended the orbital sliding method of calculating no-fit-polygons to enable it to handle arcs and used hill climbing and tabu local search algorithm to search the sequence. Leung et al. [16] presented an extended local search algorithm swapping two given polygons in one placement and placing one polygon into a new position and used the tabu search algorithm to avoid local minima. Pinheiro et al. [17] proposed a random-key genetic algorithm, which prescribed the integration of the aforementioned metaheuristic and well-known placement rules. Mundim et al. [18] derived new concepts as the no-fit raster, which could be used to check overlapping between any two-dimensional generic-shaped pieces, and used a biased random-key genetic algorithm to determine the sequence in which pieces were packed. Martinez-Sykora et al. [19] proposed a constructive approach allowing both free orientation for the pieces, as in the case of the ceramic industry, or a finite set of orientations as in the case of the garment industry.

The legal placement methods have the advantage of generating feasible solutions and reducing an infinite solution space to a discrete set of solutions. However, reducing the number of solutions may generate the possibility of omitting the global optimum at the same time.

The obvious characteristic of the relaxed placement method is to allow overlaps during the packing process. It is very easy to construct an initial layout. The layout can be built randomly, and better results might be obtained by applying a good placement strategy. To minimize the

amount of overlap, the way of imposing penalties in objective functions was used widely.

Oliveira and Ferreira [20] applied a polygon-based representation where identifying and resolving overlap was handled by using D-functions. Jakobs [21] used minimum rectangular enclosures to represent the irregular pieces and then placed the pieces by the orthogonal method. Besides, the compaction algorithm was applied to move the piece towards the BL position in the orthogonal method, later, hybridizing linear programming compaction with a search algorithm to handle the process of moving pieces within the layout in Bennell and Dowsland [22]. Gomes and Oliveira [23] improved this approach by using a Simulated Annealing (SA) algorithm to search the solution space, where linear programming compaction was used to build the neighborhood structure. To avoid local optimum, meta-heuristics for resolving overlap have become quite popular in the relaxed placement methods [24–26].

The difference between the relaxed placement method and the legal placement method is that the former usually build the layout in a continuous region. This means that solution space can be infinite. The relaxed placement method has the advantage that no solutions are lost in the unchanged solution space. However, the time consumption of these methods may be unreasonable. In addition, such approaches even do not guarantee to produce a feasible solution. By contrast, the legal placement method produces a limited number of solutions, but the solution is effective and the run time of the algorithm is acceptable.

These approaches mentioned above can effectively solve the irregular packing problem, but the time consumption of the algorithm still can be reduced. More specifically, under the orbiting method of NFP, creating potential translation vectors and finding a feasible translation vector can be faster, and this accounts for a large portion of time consumption. Besides, the local evaluation of the placement position is extensively used, which may lead to the local optimal solution to some degree. Also, the search algorithms in the former literature have a random character. The algorithm needs to be run several times to obtain a better solution, even though it cannot always obtain a feasible solution. In this article, a hybrid search algorithm is proposed to search the sequence of pieces to be packed, which combines the local evaluation and global evaluation, and, an improved orbiting method of NFP, which reduces the steps of creating NFP in Burke et al. [27]. The proposed hybrid algorithm BSTS is somehow deterministic and can produce some good results with less time-consuming.

The rest of the article is as follows: Section 2 describes using the no-fit-polygon and the novel placement principle to construct the solution under the situation of representing the problem as a sequence of pieces. Section 3 depicts the hybrid search algorithm used to search over the sequence of pieces. When a new sequence is produced by the hybrid search algorithm, the algorithm described in Section 2 will be called to construct the layout. The computational results on many benchmark problems and conclusions are, respectively, presented in Sections 4 and 5.

2. Novel Placement Principle Based on Improved NFP

The main contribution of this article is to find out an applicative and effective algorithm for the irregular shape packing problem. The problem is presented as a sequence of pieces that be packed with a novel placement principle. To describe the hybrid algorithm clearly, some related concepts are defined and explained first. The NFP can efficiently avoid overlaps during the packing process. The placement principle is applied to rapidly find the placement positions on NFPs and place the pieces. It is also necessary to imply a search algorithm to search over the sequence of pieces.

2.1. The Improved NFP. The no-fit-polygon concept was first introduced by Art [5] and used to avoid overlaps when placing the pieces on the plate. Later it was developed and improved by several researchers. Although different approaches were proposed to generate NFP, they could be categorized into three general methods: (a) the orbiting algorithm [27, 28]; (b) Minkowski sums [29, 30]; (c) decomposition algorithm [31–33].

When polygon B slides around the external edge of polygon A (fixed), the locus of the reference point (on polygon B) form the NFP_{AB} . During the orbital process, the relative orientations of the two polygons remained unchanged (see Figure 1). There are three conditions about the position of the reference point.

If the reference point of B is positioned inside NFP_{AB} , A and B will overlap; if the reference point of B is positioned on the boundary of NFP_{AB} , B will touch A; if the reference point of B is positioned outside NFP_{AB} , A and B will neither overlap nor touch.

Hence, the problem of relative positions of two polygons can be simplified as finding the relative positions between one point and one polygon (see Figure 2).

In this article, the calculation of NFP is handled by the orbital algorithm. The improved approach is logically similar to the algorithm of Burke et al. [27], while a modified implementation is proposed.

In this section, the forming process of NFP is described. When polygon B slides around A, each step creates an edge of the NFP. Unlike the algorithm of Burke, the process of producing NFP is broken down into the following three subparts: finding potential vectors, eliminating infeasible vectors, and calculating minimum distance. This method can reduce the time of creating potential translation vectors and finding a feasible translation vector.

2.1.1. Subheadings Finding Potential Vectors. As previously described, the NFP is a polygon with each edge either derived from an edge of A or B, depending on the situation (see Figure 3).

The set of potential translation vectors is obtained by the touching edge pairs. There are three possible relative positions of two oriented edges (see Figure 4):

- (1) A vertex of the orbiting edge b touches the middle of the fixed edge a; the potential vector is $\overrightarrow{P_t P_e}$.
- (2) A vertex of the fixed edge a touches the middle of the orbiting edge b; the potential vector is $\overrightarrow{P_e P_t}$.
- (3) Both edges touch at a vertex. In this case, the potential vector can be detected by the following rules: if the angle from vector $\overrightarrow{B_j B_{j+1}}$ to $\overrightarrow{A_j A_{j+1}}$ is less than 180° , the translation vector is $\overrightarrow{A_j A_{j+1}}$; otherwise, it is $\overrightarrow{B_{j+1} B_j}$ (see Figure 5).

2.1.2. Eliminating Infeasible Vectors. The potential translation vector produced by the method mentioned above might be infeasible. This section is to eliminate those potential vectors which result in immediate intersections. For example, in Figure 6(a), two potential translation vectors, a4 and -b1, can be created according to the rules mentioned in case 3. Figure 6 demonstrates how to identify an infeasible translation vector (a4). The edge-pairs involving edge a4 have been omitted for brevity. Only the edge pairs of the other touch point need to be tested. If translating polygon B along vector a4, this would result in an immediate intersection between edges a1 and b1 (and also a2 and b1). Once one translation is infeasible, the vector should be eliminated, so a4 is not a feasible translation vector.

2.1.3. Calculating Minimum Distance. Before polygon B translates along the translation vector, the feasible translation vector must be trimmed, which means calculating the minimum distance of the vector. This is necessary because other edges may intersect during the translation of polygon B. For example, in Figure 7(a), polygon A and polygon B intersect because of applying the entire vector without trimming. In this case, the minimum distance from the original point to the intersection point defines the final feasible vector (see Figure 7(b)).

2.2. The Novel Placement Principle. The solution space of the irregular packing problem is infinite without limiting the placement positions and rotations. To reduce the solution space, some placement principles have been presented, such as BL and TOPOS. BL principle, in which a piece is placed to the bottom-left position, usually works for the orthogonal packing problem, not for the irregular packing problem. TOPOS algorithm is to place the piece by evaluating the placement positions with the evaluation criteria.

In this section, to determine placement position and rotation angle for each piece, a new placement principle based on BL principle and lowest-gravity-center principle [34] is applied. The modified and concise criteria are proposed to evaluate the best placement position. Two attributes (length L and area A of the rectangle enclosure) are selected to create the criteria Z as follows (equations (1)–(3)). Figure 8 displays the meaning of the symbol in equations.

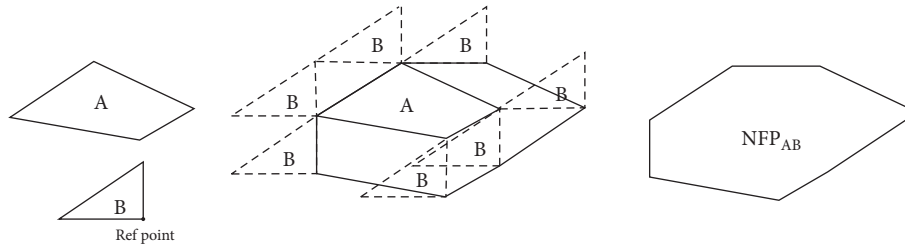


FIGURE 1: The no-fit-polygon of two polygons A and B.

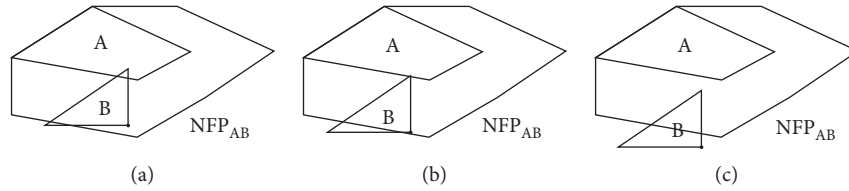


FIGURE 2: Three relative positions of the reference point and NFP_{AB} : (a) overlapping, (b) touching, and (c) neither overlapping nor touching.

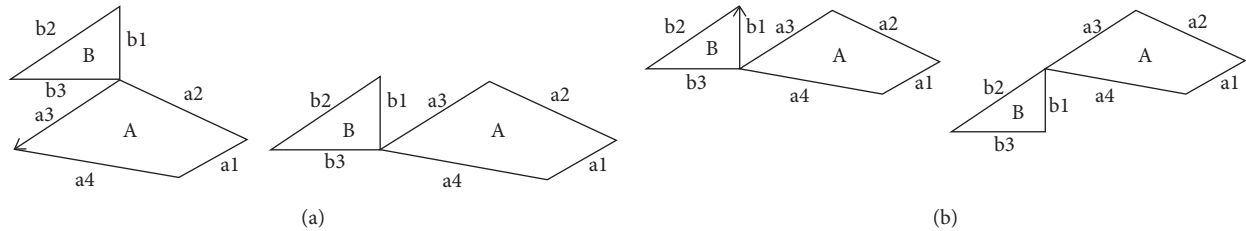


FIGURE 3: Translation vector: (a) derived from edge a_3 and (b) derived from edge b_1 .

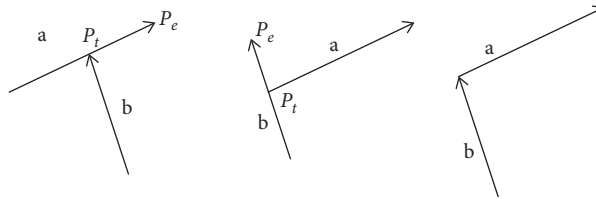


FIGURE 4: Three touching edge-pair types.

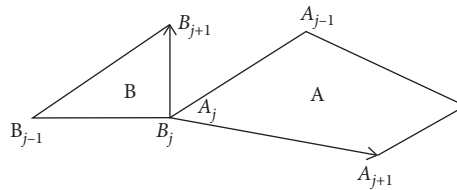


FIGURE 5: Both edges touch at a vertex.

$$\text{length: } L = \frac{L_{\text{new}}}{x_{\text{new}}}, \quad (1)$$

$$\text{area: } A = \frac{L_{\text{new}} \times W_{\text{new}}}{x_{\text{new}} \times y_{\text{new}}}, \quad (2)$$

$$\text{CostFuction(criteria): } Z = \min(L + A). \quad (3)$$

Now, given a piece, its gravity center is first calculated and then can generate two NFPs (gravity-center NFP and

bottom-left NFP) by viewing the gravity center and the bottom-left vertex as the reference point, respectively. Note that both gravity-center NFP and bottom-left NFP will be different when different rotations of the piece are performed. The lowest gravity center and bottommost-leftmost vertex are chosen to be evaluated by the criteria presented above. The position with the best evaluation is determined to be the placement position. Figure 9 shows an example of the presented placement principle. The two solid polygons (noted 1, 2) are the gravity-center NFPs with two different

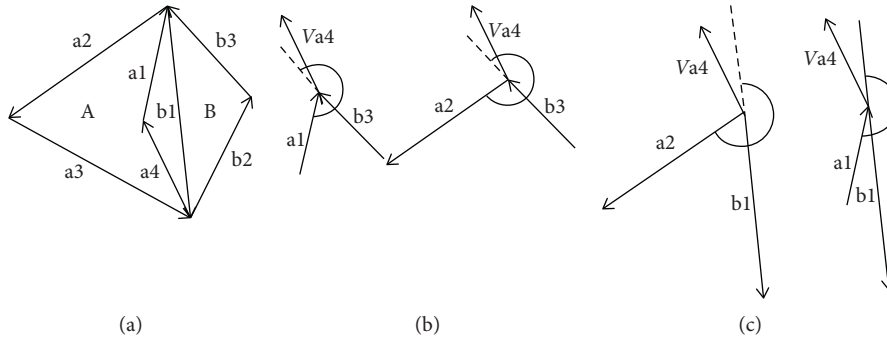


FIGURE 6: (a) Two vertex-vertex touch cases, (b) feasible edge translations of a4, and (c) infeasible edge translations of a4.

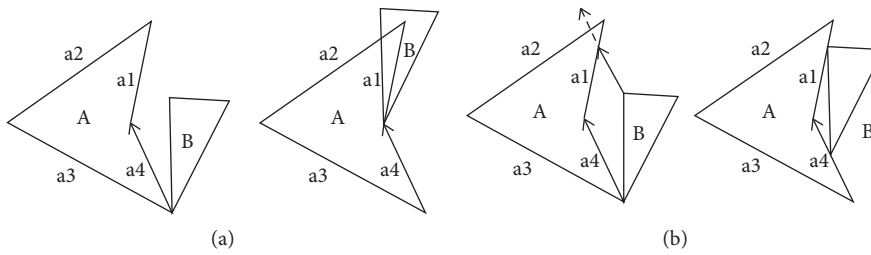


FIGURE 7: The final feasible translation vector: (a) untrimmed and (b) trimmed.

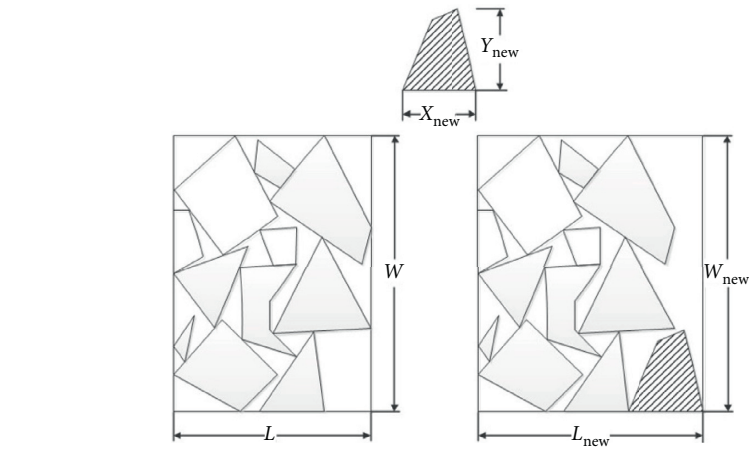


FIGURE 8: Illustration of the criteria for the new piece.

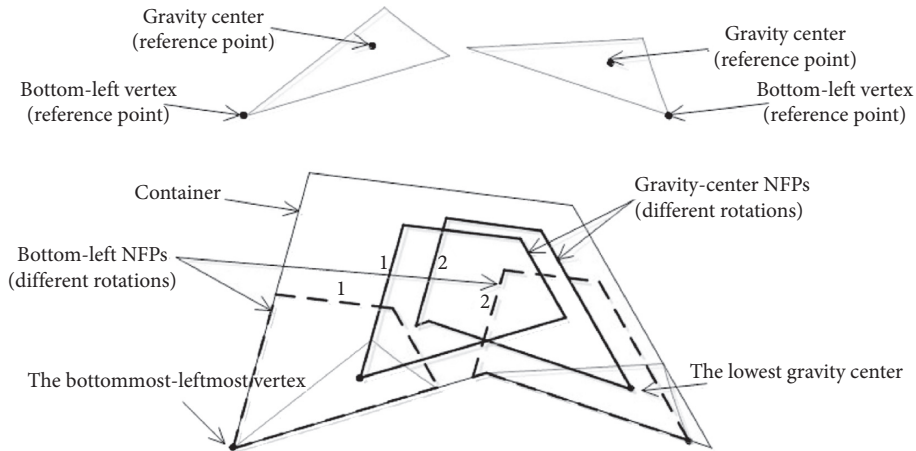


FIGURE 9: Finding the bottommost-leftmost vertex and lowest gravity center.

rotations. Meanwhile, the dotted polygons, noted 1 and 2 (some dotted edges coincide with the edges of the container) are the bottom-left NFPs correspondingly.

3. The Hybrid Algorithm for Searching Sequence

Usually, some sorting criteria, such as the area or the length, are applied to determine the packing sequence of pieces. Better solutions can be produced if a search algorithm is applied to searching over the sequence. BS and TS algorithms can successfully solve the 2DIP problem, but there are still some imperfections. The beam search is a heuristic with a strong global search and deterministic. However, this may also generate the possibility of removing other good solutions. To deal with the problem, tabu search is hybridized with a strong local search with beam search to implement an applicative and effective hybrid search algorithm BSTS. The focus of this section is on the hybridization of tabu search with beam search in the resolution of the packing sequence.

3.1. Tabu Search. Different from some greedy algorithms, tabu search is a search algorithm with a neighborhood set that accept worse solutions, not always better solutions. The advantage of this characteristic is that the tabu search can escape from local minima. Tabu list is a dynamic and flexible memory structure that prevents the algorithm from performing the recently executed moves in the next few iterations.

The tabu search mechanism in this article is similar to the algorithm of Glover and Taillard [35]. The neighborhood size in the tabu search is set to 5 and the tabu list is 100, which contains the recently searched solutions, not the moves. In this article, to extensively search over the packing sequence, tabu search is applied to hybridize with beam search. The focus of the tabu search is on escaping from local minima. So the neighbour structure is defined as a 4-way swap where four randomly selected pieces are swapped. The process can be represented in the following way (see Figure 10).

3.2. Beam Search. Beam search is an established heuristic derivative branch-and-bound. A common application is the algorithm that uses a tree search mechanism in which only the most promising nodes are performed to branch at each level according to the evaluations. Usually, a two-stage beam search is applied with two evaluations: local evaluation and global evaluation. A local evaluation (also called rough evaluation) can be faster but may result in discarding some good solutions, whereas a global evaluation (also called accurate evaluation) usually may be more time-consuming. So, the combination of the two evaluations is significant in most beam search algorithms.

First, a local evaluation is applied to evaluate all child nodes (branching from each beam node) and select the best γ nodes (filter width) at each level. Then, the global evaluation is used to accurately evaluate the γ nodes and retain the best w nodes (beam width) to branch (see Figure 11).

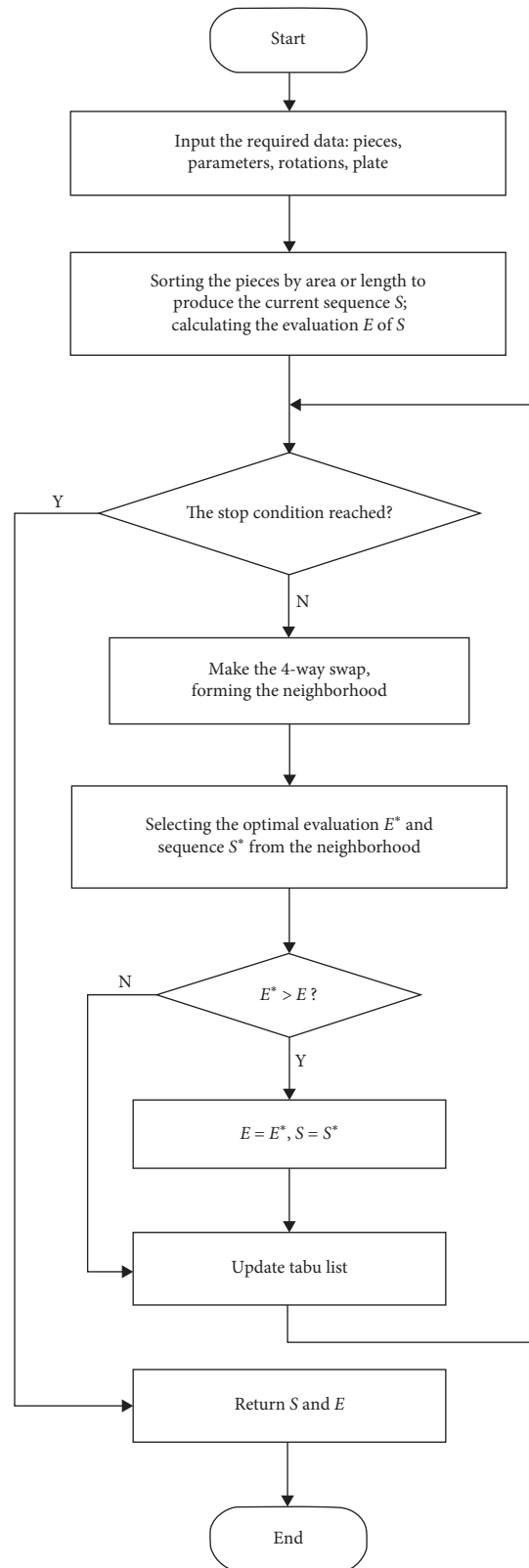


FIGURE 10: The algorithm of tabu search.

In this article, the search tree, from the root to the last level, represents the packing sequence of all pieces. At each level, each node with its parent nodes in the upper levels

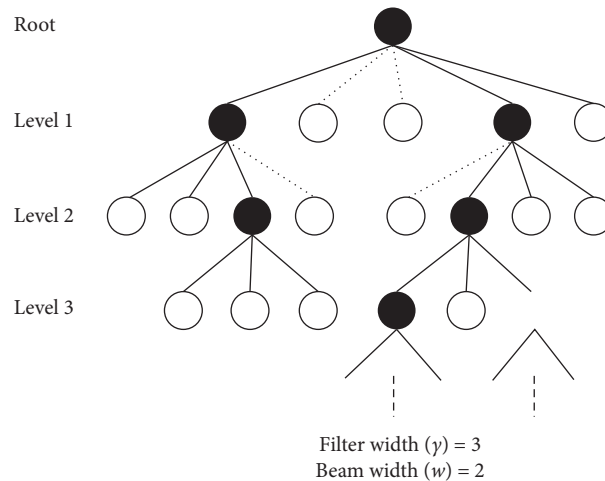


FIGURE 11: Illustration of a beam search.

represents a partial solution. Each branch from each beam node means adding an unpacked piece to the partial solution until the lowest level, which means the complete solution. Obviously, from the root to the lowest level, the number of levels is the number of all pieces. And at each level, the number of nodes branching from its parent node is the number of the remaining piece types. Figure 12 shows how beam search works for the problem.

3.3. Hybrid Algorithm. As mentioned above, the beam search is a heuristic with a strong global search and deterministic. However, this may also generate the possibility of removing other good solutions. To deal with the problem, hybridize tabu search with a strong local search with beam search to implement an applicative and effective hybrid search algorithm BSTS.

The hybrid search algorithm uses the same process of the beam search except that using global evaluation to evaluate the complete solution. In the hybrid search algorithm, the sequence of the already packed pieces has been determined by the BS mechanism and only the unpacked pieces need to be searched by tabu search, not simply ordered randomly or by decreasing area. The hybrid search algorithm is a two-stage strategy, where the local evaluation is the same criteria was presented in the placement principle, but the global evaluation is the length of the complete solution. The process of the algorithm can be represented as in Figure 13.

4. Results

In this section, the hybrid algorithm is tested by the benchmark problems previously published in the literature. To provide good solutions, NFP and the novel placement principle are discussed in Section 3 to place each piece by the sequence, which is given by the BSTS. Implementation and parameters are set by preliminary computational experiments, which are not discussed in this article.

4.1. Problem Instances. Hybrid algorithm (BSTS) is tested by the packing problem instances, also used as

benchmark problems in other literature. The data files can be downloaded from the ESICUP website (<http://www.fe.up.pt/esicup>). The details can be summarized in Table 1.

4.2. Results. The computational tests were conducted on a Core 2 D 2.0 GHz processor with 1024 Mb of RAM. The program was coded in Visual Studio C++.

As presented above, the times of applying global evaluation are rising with the increase of the filter width γ of the hybrid algorithm, which means more time-consuming. Furthermore, if the tabu search used to determine the sequence of the unpacked pieces is complicated, the hybrid algorithm may not be acceptable. According to the preliminary computational experiments, the parameters were set by the following rules: filter width = $\min \{M, O * 5\}$, Beam widths = $\{10, 100\}$ with tabu iterations = $(N - n - 1) * 5$, where N was the total number of all pieces, n was the number of packed pieces, M was the total number of the piece types, and O was the number of the allowed orientations. The neighborhood size in the tabu search is set to 5 and the tabu list is 100.

The computational tests consisted of 20 runs for each instance, both for the simulated annealing hybrid algorithm (SAHA) and the Random-Key Genetic Algorithm (μ -BRKGA). The SAHA and μ -BRKGA algorithms can obtain a good solution to the problem. There are some differences between these algorithms in the computational environment. The details about the hardware environment are shown in Table 2. The detailed parameters of the algorithms are represented in their published literature. Thus results are compared with the results of SAHA [23] in Table 3, and compared with the results of μ -BRKGA [36] in Table 4. The results of these algorithms are taken from their original articles.

The better results obtained by the hybrid algorithm BSTS have been highlighted. Less than 1% worse results are underlined. In comparison to the average results obtained by SAHA, the proposed BSTS algorithm produces three better

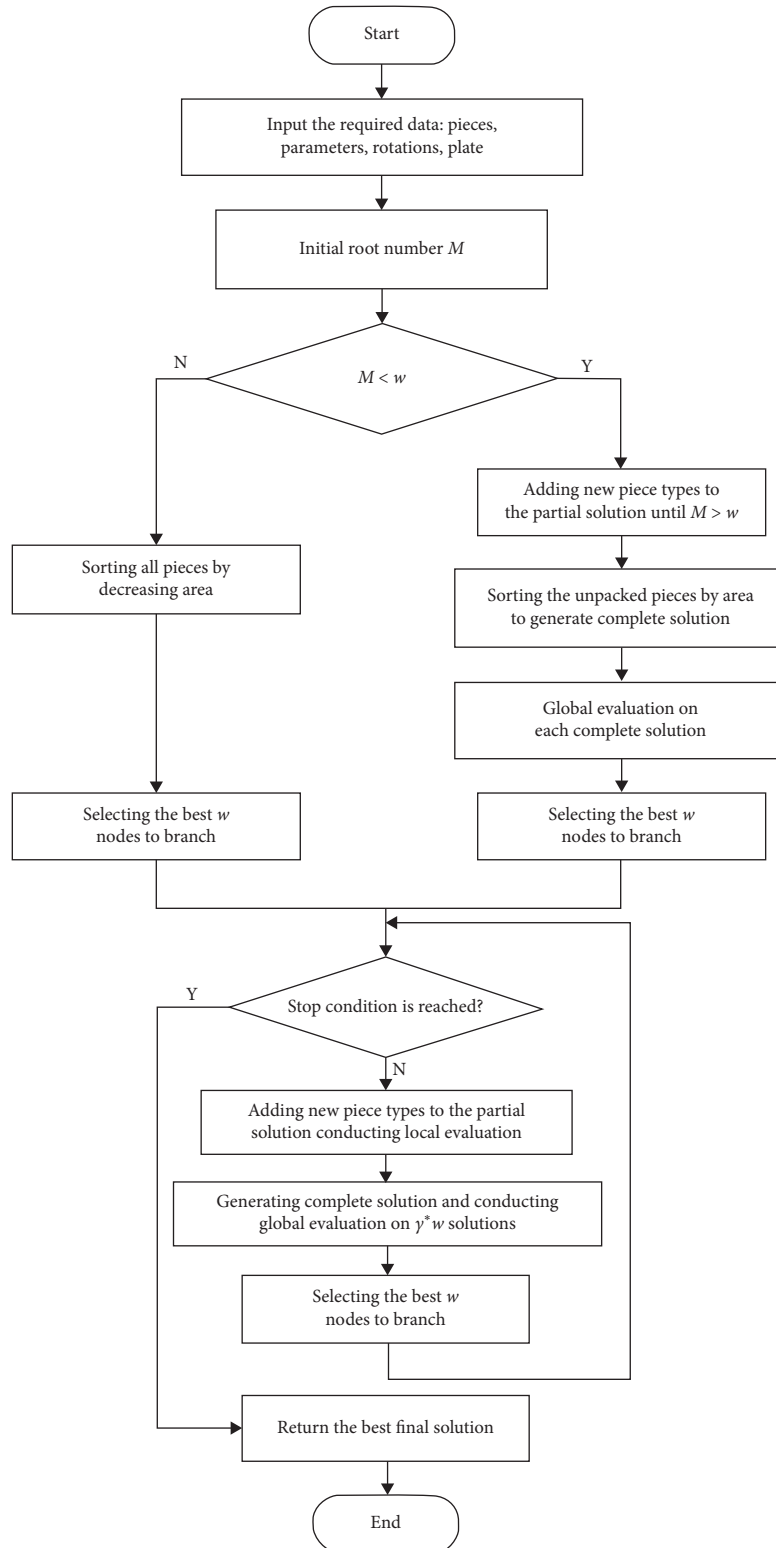


FIGURE 12: The algorithm of beam search.

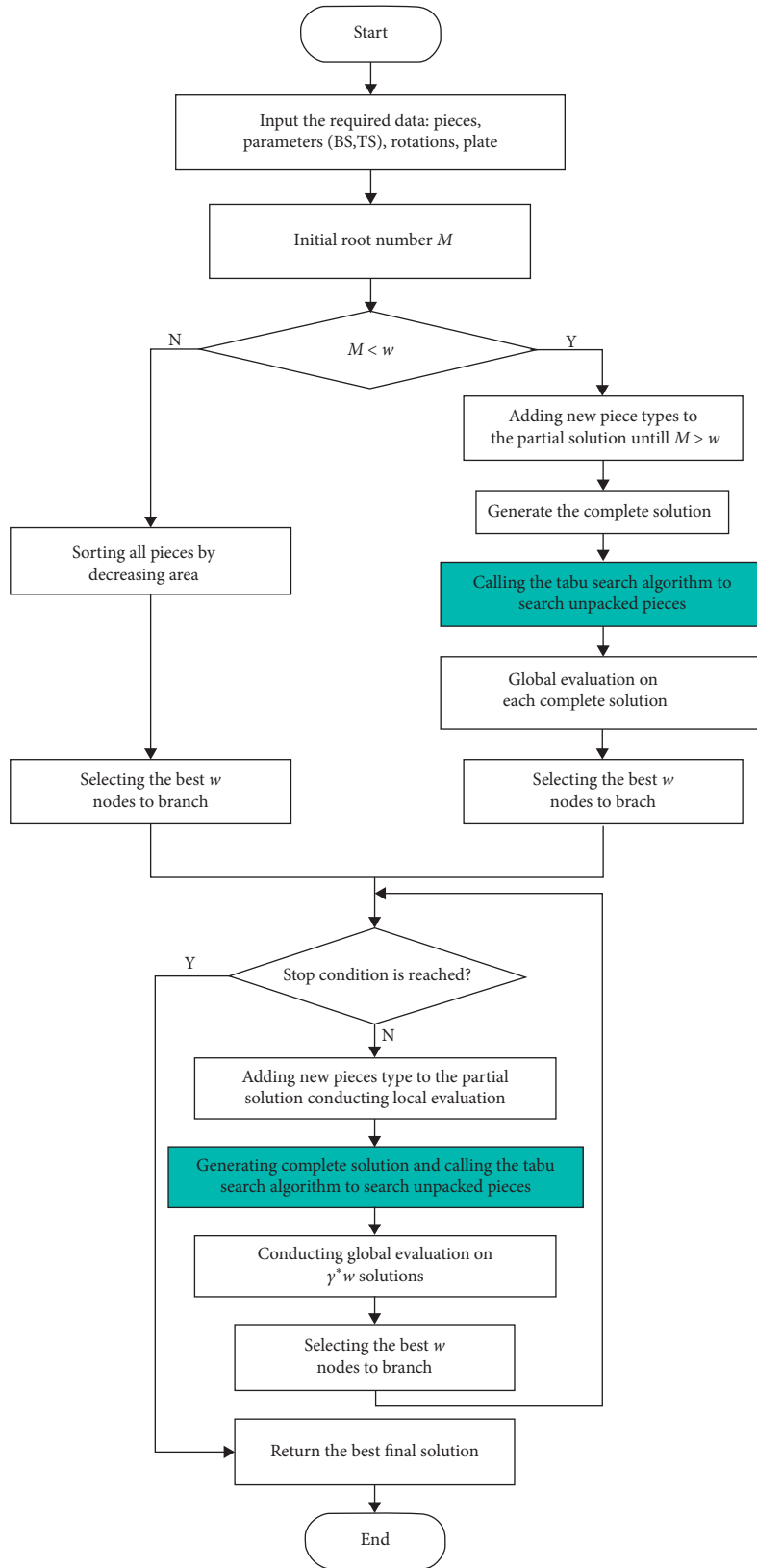


FIGURE 13: The hybrid search algorithm.

TABLE 1: Details of the benchmark problems.

Problem instance	Number of piece types	Total number of pieces	Rotational constraints	Sheet width
Fu	12	12	90 incremental	38
Jakobs1	25	25	90 incremental	40
Jakobs2	25	25	90 incremental	70
Shapes0	4	43	0 absolute	40
Shapes1	4	43	0, 180 absolute	40
Shapes2	7	28	0, 180 absolute	15
Dighe1	16	16	0 absolute	100
Dighe2	10	10	0 absolute	100
Albano	8	24	0, 180 absolute	4900
Dagli	10	30	0, 180 absolute	60
Mao	9	20	90 incremental	2550
Marques	8	24	90 incremental	104
Swim	10	48	0, 180 absolute	5752
Trousers	17	64	0, 180 absolute	79

TABLE 2: Computational environments.

Algorithm	#lan	CPU	RAM (G)	Times
BSTS	VS C++	Core 2 D CPU at 2.0 GHz	1	20
SAHA	C++	Pentium IV CPU at 2.13 GHz	0.5	20
μ -BRKGA	Java	Intel i7- 960 CPU at 3.2 GHz	8	20

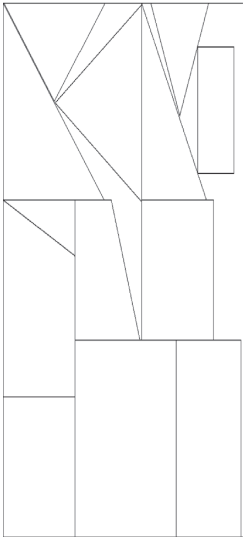
TABLE 3: BSTS algorithm results compared with the average results of SAHA.

Problem	Average results of SAHA				Average results of BSTS		
	Length	Avg. (%)	Time (s)	BW/FW	Length	Avg. (%)	Time (s)
Fu	32.70	87.15	296	<i>100/12</i>	32.99	86.38	282
Jakobs1	12.93	75.80	332	100/20	11.98	81.87	639
Jakobs2	25.86	74.62	454	100/20	25.84	74.66	777
Shapes0	63.15	63.18	3914	10/4	66.11	60.38	363
Shapes1	58.17	68.59	10314	10/4	60.14	65.26	557
Shapes2	26.53	81.41	2257	100/7	28.48	75.82	401
Dighe1	122	81.97	83	10/5	125.66	79.48	185
Dighe2	119.53	83.66	22	10/5	<i>121.12</i>	82.56	92
Albano	10280.1	84.68	2257	100/8	<i>10443.1</i>	83.37	926
Dagli	59.41	85.36	5110	100/10	62.37	80.93	413
Mao	1842.70	79.99	8245	100/9	1839.4	80.12	759
Marques	79.63	86.87	7507	100/8	79.94	86.40	591
Swim	6121.39	72.27	6937	100/10	6369.1	69.46	855
Trousers	244.68	89.01	8588	100/10	252.07	86.41	641

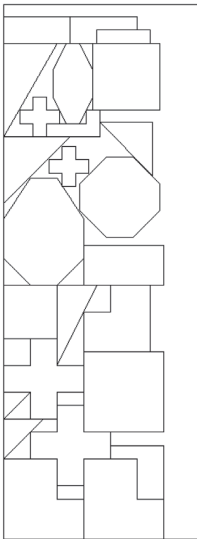
Compared with the average results of SAHA, the better results obtained by the hybrid algorithm BSTS are highlighted by bold. Less than 1% worse results are italicized, and this also shows that the BSTS algorithm can produce some comparable results. The bold and italicized values can display the advantage of the BSTS algorithm more clearly.

TABLE 4: BSTS algorithm results compared with the results of μ -BRKGA.

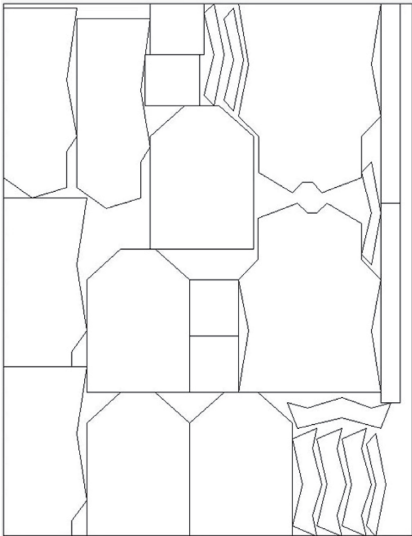
Problem	The results of μ -BRKGA				The results of BSTS		
	Best (%)	Avg. (%)	Time (s)	BW/FW	Best (%)	Avg. (%)	Time (s)
Fu	91.96	88.34	1815	100/12	88.79	86.38	282
Jakobs1	89.09	78.79	6300	100/20	83.55	81.87	639
Jakobs2	79.75	77.90	7150	100/20	75.72	74.66	777
Shapes0	66.50	63.50	1200	10/4	62.19	60.38	363
Shapes1	72.55	65.89	1860	10/4	67.55	65.26	557
Dighe1	100.00	100.00	1920	10/5	82.21	79.48	185
Dighe2	100.00	100.00	2050	10/5	85.12	82.56	92
Albano	89.11	84.69	4200	100/8	84.46	83.37	926
Marques	89.14	84.31	3240	100/8	87.31	86.40	591
Swim	73.59	70.78	7400	100/10	71.74	69.46	855
Trousers	90.52	88.67	7250	100/10	88.84	86.41	641



(a)



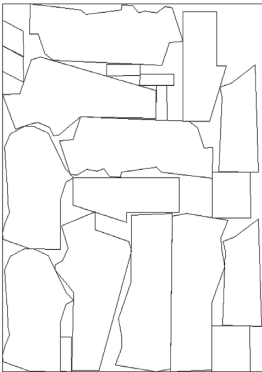
(b)



(c)



(d)



(e)

FIGURE 14: Continued.



FIGURE 14: (a) Fu: 88.36% (282 s), (b) Jakobs1: 83.55% (639 s), (c) Marques: 86.40% (591 s), (d) Swim: 71.74% (855 s), (e) Mao: 81% (759 s), (f) Albano: 84.46% (926 s), and (g) Trousers: 88.84% (641 s).

TABLE 5: The detailed results of 20 runs for 14 instances.

Fu	Jakobs1	Jakobs2	Shapes0	Shapes1	Shapes2	Dighe1	Dighe2	Albano	Dagli	Mao	Marques	Swim	Trousers
86.36	79.67	74.23	61.01	67.31	75.26	78.5	85.12	84.46	80.37	81	86.76	68.83	86.77
88.79	80.99	73.67	60.82	67.55	80.23	77.2	81.67	84.46	83.01	78.24	86.4	69.58	88.84
85.84	81.67	74.59	60.92	64.04	73.72	76.62	83.37	82.74	79.37	79.15	85.93	69.51	86.6
86.1	81.67	75.54	62.19	66.17	76.06	80.14	80.15	84.06	80.5	80.11	86.76	69.38	86.6
86.36	81.67	74.59	59.55	66	76.06	80.84	82.12	83.4	83.01	79.89	85.93	69.11	85.55
86.36	83.55	74.23	60.36	63.43	75.26	80.26	80.64	83.4	81.27	80.87	87.31	68.47	85.92
86.36	81.67	74.23	60.36	64.35	76.06	78.18	82.75	84.27	79.49	79.3	86.76	68.74	86.64
86.36	83.55	74.66	61.29	66.17	76.33	80.35	82.25	84.06	80.37	79.38	85.4	68.22	86.09
87.66	81.67	74.23	59.55	63.45	76.06	78.39	84.85	82.7	80.89	80.01	86.68	69.82	86.4
85.86	81.67	75.72	60.36	65.2	79.33	82.21	81.03	83.05	82.06	80.66	87.31	68.71	86.88
86.1	83.33	74.23	60.72	65.41	76.06	80.32	81.89	82.61	80.76	80.87	85.93	69.92	86.05
85.84	82.28	74.23	60.27	63.33	76.33	78.61	81.76	82.54	80.76	79.38	86.25	70.41	86.4
86.36	80.33	74.23	59.91	64.04	76.33	78.67	83.31	83.05	80.12	80.11	86.25	68.92	86.36
86.36	81.67	75.54	59.55	63.74	75.26	79.68	82.56	83.81	81.54	79.97	84.98	70.6	87.26
85.84	81.67	75.09	59.55	63.84	75.52	79.41	82.75	83.05	81.54	81	86.25	71.74	85.75
86.36	81.67	74.34	60.45	66.17	76.87	80.38	84.46	82.61	79.74	80.68	86.76	69.08	86.12
85.84	82.49	74.23	59.55	63.94	75.26	78.39	81.82	83.27	81.67	80.87	85.4	68.89	85.45
86.1	82.86	74.59	59.82	67.55	73.97	78.45	84.46	83.05	80.37	79.89	86.68	69.33	86.19
86.36	81.67	75.72	61.2	66.17	75	81.31	82.18	83.36	79.62	80.87	86.76	70.41	86.36
86.36	81.67	75.39	60.18	67.31	74.48	81.73	82	83.36	82.06	80.11	87.31	69.04	86.05

results (Jakobs1, Jakobs2, Mao) and four comparable results (Dighe2, Fu, Albano, Marques) which are within 1% of the average results. In comparison to the average results obtained by μ -BRKGA, the proposed BSTS algorithm produces two better average results (highlighted in Table 4). Although the best results of μ -BRKGA surpass the BSTS, the time of μ -BRKGA much overtakes the BSTS. In ten benchmark problems, there are all solutions (highlighted in Table 4) to be produced much faster than μ -BRKGA. The layouts for the best results of the seven cases can be found in Figure 14 in Appendix. The details of 20 runs for the 14 benchmark problems are also shown in Table 5 in Appendix.

5. Conclusions

The article describes a hybrid search algorithm for the irregular packing problem. The packing problem is presented as a sequence of pieces. The sequence is searched by the proposed hybrid search algorithm BSTS and decoded by an improved placement principle. The improved orbiting method of NFP is presented. It can create potential translation vectors and find a feasible translation vector in a significantly faster time. The placement position of each piece is determined by an evaluation, which is also the local evaluation of the hybrid search algorithm. To generate

complete solutions, tabu search is applied to search over the unpacked pieces and decide Beam nodes by using global evaluation. The hybrid algorithm BSTS is somehow deterministic and can produce some good results with less time consumed. Furthermore, hybridizing BS with TS can decrease the possibility of removing other good solutions caused by the determinacy of the single BS approach.

The experiments show that the hybrid algorithm is an applicative and effective algorithm for the irregular packing problem. To demonstrate its efficiency and determinacy, the proposed algorithm can produce three better results and four comparable results that are within 1% of the average results of SAHA in the 14 benchmark problems. Also, the algorithm can produce available solutions much faster than μ -BRKGA approaches.

In future work, the performance of the BSTS will be examined in more instances available on the ESICUP website, extending the ideas developed here to other bin packing problems with irregular pieces, such as the problems arising from cutting leather, in which bins are neither identical nor rectangular, and can even have some defects, defining nonusable zones.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant no. 51975231) and Foundational Research Funds for the Central Universities (Grant no. 2019kfyXKJC043).

References

- [1] 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.
- [2] M. R. Garey and D. S. Johnson, "Computers and intractability. A guide to the theory of NP-completeness," *Journal of Symbolic Logic*, vol. 48, no. 2, pp. 498–500, 1983.
- [3] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82–117, 2013.
- [4] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Computers & Industrial Engineering*, vol. 137, 2019.
- [5] R. C. Art Jr., *An Approach to the Two Dimensional Irregular Cutting Stock Problem*, Massachusetts Institute of Technology, Cambridge, MA, USA, 1966.
- [6] K. A. Dowsland and W. B. Dowsland, "Packing problems," *European Journal of Operational Research*, vol. 56, no. 1, pp. 2–14, 1992.
- [7] K. A. Dowsland, W. B. Dowsland, and J. A. Bennell, "Jostling for position: local improvement for irregular cutting patterns," *Journal of the Operational Research Society*, vol. 49, no. 6, pp. 647–658, 1998.
- [8] 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.
- [9] J. F. Oliveira, A. M. Gomes, and J. S. Ferreira, "TOPOS—a new constructive algorithm for nesting problems," *OR Spektrum*, vol. 22, no. 2, pp. 263–284, 2000.
- [10] K. A. Dowsland, S. Vaid, and W. B. Dowsland, "An algorithm for polygon placement using a bottom-left strategy," *European Journal of Operational Research*, vol. 141, no. 2, pp. 371–381, 2002.
- [11] A. R. Babu and N. R. Babu, "A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms," *Computer-Aided Design*, vol. 33, no. 12, pp. 879–891, 2001.
- [12] E. Burke, 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.
- [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, 2008.
- [14] A. K. Sato, T. de Castro Martins, and M. de Sales Guerra Tsuzuki, "A simulated annealing based algorithm with collision free region for the irregular shape packing problem," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 3968–3973, 2011.
- [15] 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.
- [16] S. C. H. Leung, Y. Lin, and D. Zhang, "Extended local search algorithm based on nonlinear programming for two-dimensional irregular strip packing problem," *Computers & Operations Research*, vol. 39, no. 3, pp. 678–686, 2012.
- [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, 2015.
- [18] L. R. Mundim, M. Andretta, and T. A. de Queiroz, "A biased random key genetic algorithm for open dimension nesting problems using no-fit raster," *Expert Systems with Applications*, vol. 81, pp. 358–371, 2017.
- [19] A. Martinez-Sykora, R. Alvarez-Valdes, J. A. Bennell, R. Ruiz, and J. M. Tamarit, "Matheuristics for the irregular bin packing problem with free rotations," *European Journal of Operational Research*, vol. 258, no. 2, pp. 440–455, 2017.
- [20] J. F. C. Oliveira and J. A. S. Ferreira, "Algorithms for nesting problems," in *Applied Simulated Annealing*, pp. 255–273, Springer, Berlin, Germany, 1993.
- [21] S. Jakobs, "On genetic algorithms for the packing of polygons," *European Journal of Operational Research*, vol. 88, no. 1, pp. 165–181, 1996.
- [22] J. A. Bennell and K. A. Dowsland, "Hybridising tabu search with optimisation techniques for irregular stock cutting," *Management Science*, vol. 47, no. 8, pp. 1160–1172, 2001.
- [23] A. M. Gomes and J. F. Oliveira, "Solving irregular strip packing problems by hybridising simulated annealing and linear programming," *European Journal of Operational Research*, vol. 171, no. 3, pp. 811–829, 2006.
- [24] R. Heckmann and T. Lengauer, "A simulated annealing approach to the nesting problem in the textile manufacturing industry," *Annals of Operations Research*, vol. 57, no. 1, pp. 103–133, 1995.

- [25] J. A. Bennell and K. A. Dowsland, "A tabu thresholding implementation for the irregular stock cutting problem," *International Journal of Production Research*, vol. 37, no. 18, pp. 4259–4275, 1999.
- [26] J. Egeblad, B. K. Nielsen, and A. Odgaard, "Fast neighborhood search for two- and three-dimensional nesting problems," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1249–1266, 2007.
- [27] 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.
- [28] A. Mahadevan, *Optimization in computer-aided pattern packing (marking, envelopes)*, Ph.D. thesis, North Carolina State University, Raleigh, NC, USA, 1984.
- [29] P. K. Ghosh, "An algebra of polygons through the notion of negative shapes," *CVGIP: Image Understanding*, vol. 54, no. 1, pp. 119–144, 1991.
- [30] 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.
- [31] Z. Li and V. Milenkovic, "Compaction and separation algorithms for non-convex polygons and their applications," *European Journal of Operational Research*, vol. 84, no. 3, pp. 539–561, 1995.
- [32] P. D. Watson and A. M. Tobias, "An efficient algorithm for the regular W1 packing of polygons in the infinite plane," *Journal of the Operational Research Society*, vol. 50, no. 10, pp. 1054–1062, 1999.
- [33] P. K. Agarwal, E. Flato, and D. Halperin, "Polygon decomposition for efficient construction of Minkowski sums," *Computational Geometry*, vol. 21, no. 1-2, pp. 39–61, 2002.
- [34] H.-y. Liu and Y.-j. He, "Algorithm for 2D irregular-shaped nesting problem based on the NFP algorithm and lowest-gravity-center principle," *Journal of Zhejiang University—SCIENCE A*, vol. 7, no. 4, pp. 570–576, 2006.
- [35] F. Glover and E. Taillard, "A user's guide to tabu search," *Annals of Operations Research*, vol. 41, no. 1, pp. 1–28, 1993.
- [36] B. Amaro Júnior, P. R. Pinheiro, and P. V. Coelho, "A parallel biased random-key genetic algorithm with multiple populations applied to irregular strip packing problems," *Mathematical Problems in Engineering*, vol. 2017, Article ID 1670709, 11 pages, 2017.