Journal of Computers Vol. 34 No. 5, October 2023, pp. 59-71
doi: 10.53106/199115992023103405005

A New Comprehensive Algorithm for the Cutting and Packing Problem with Position Constraints

Hao Yuan[1], Kai Xu[1*], Xiao Meng[2], Qing Jia[3], Mingzhu Li[1]

[1] Jiangsu University, Zhenjiang, Jiangsu, China
285817681@qq.com, 1210598500@qq.com

[3] Changzhou Sinajet Science and Technology Co., Ltd, Changzhou, Jiangsu, China
sales@sinajet.cn

Received 14 August 2022; Revised 24 January 2023; Accepted 13 March 2023

Abstract. The cutting and packing (CP) is a widespread problem in the production processes of garment, shoes, board, and in other manufacturing industries. It is closely related to the production and management costs. This problem refers to the placement of pieces, without overlapping, in a limited space by maximizing the space utilization. In this work, a CP variant with position constraints, namely the horizontal striped fabric layout problem (HSFL), is investigated. We propose a solution approach for it, where a top-left fill (TLF) algorithm based on key points positioning strategy is used to deal with position constraints in the stripe alignment problem. This algorithm establishes a key points sequence for every piece, when determining the placement of pieces, compare the key points sequence of pieces. This algorithm also eliminates the error often occurred in the rectangle bounding box algorithm at the time of simplifying the layout process. The TLF algorithm based on key points positioning strategy and genetic algorithm (GA) are combined for obtaining the optimum layout. The datasets used in the computational experiment are collected from an apparel industry as well as from other publications. The comparison of our experimental results with the best known results shows that the proposed approach could improve the average computing speed and achieve higher utilization in 10 out of 16 instances. It can be concluded that the proposed approach is potential in balancing the computing speed and utilization rate. The algorithm has been successfully applied in an apparel company in China.

Keywords: cutting and packing, position constraints, key points positioning strategy, apparel industry

1 Introduction

The products of the apparel industry are becoming more and more diversified [1, 2]. The apparel industry still faces a lot of challenges, a major one of which is the cutting and packing (CP) problem [3-6]. The CP problem seeks to assign pieces in a finite space in a way to arrange them by maximizing the utilization of the space.

The horizontal striped fabric layout problem (HSFL) is studied here as a variant of the CP problem with position constraints. The objective of the study is to find the most appropriate layout of pieces that can be arranged into the striped fabric of a limited size. Besides dealing with the general difficulties of the CP problem, the HSFL problem must be capable of handling an involuted task, i.e., the stripe alignment issue. According to Wäscher et al. [7], the HSFL problem can be classified as a two-dimensional irregular open dimensional problem, which is a typical NP-hard problem [8, 9].

Tsao et al. [10] investigated the general CP problem in the apparel industry. They compared several approaches and found that the performance of the tabu-search algorithm (TS) was the best in minimizing production cost; while the simulated annealing-based genetic algorithm (SA-GA) and the genetic algorithm (GA) performed better in saving operating time. As per their findings, we propose a comprehensive approach, by combining the GA and the bottom-left filling (BLF) algorithm to optimize the HSFL problem. The proposed approach also includes the rectangle bounding box algorithm and the key points positioning strategy.

In order to test the feasibility of the proposed approach, we conduct simulation experiments with 16 benchmark instances. The proposed approach could attain utilization of 80% in 15.06s for the Stripedshirts instance provided by a Chinese apparel industry. The results show that the proposed approach is very competitive in utilization and computing time.

* Corresponding Author

The rest of the paper is structured as follows: the literature on position constraints and optimal layout approaches is surveyed in Section 2. In Section 3, we describe the HSFL problem, followed by the proposed methods for it in detail in Sections 4. Section 5 presents datasets and the comparison of the best results of the literature with the proposed approach. Finally, the present work is concluded in Section 6 with some suggestions for future work.

2 Literature Review

Very limited works on the CP problem with position constraints could be found in specialized literature. Toledo et al. [11] proposed a dotted-board model, where a board is discretized into a grid of dots. These points are used as the feasible positioning reference of the pieces. Leao et al. [12] established a coordinate system, where the -axis is continuous and the -axis is discretized for defining the board in terms of a set of stripes. No-fit polygons discretized along the -axis are used to check the overlapping of pieces. In a similar work, Saravanan et al. [13] and Akunuru and Babu [14] presented a semi-discrete geometric model for discretizing the board and pieces. Based on the dotted-board model, Domovic et al. [15] introduced three algorithms: Grid, Grid-BLP, and Grid-Shaking. However, their solution was affected by the degree of dispersion, which led to the unbalanced quality of the solution and increasing computing time.

GA is considered as a robust approach for solving large-scale CP problems [2, 16]. There are many works combining GA with other hybrid algorithms for optimizing the layout of different instances. Wang et al. [17] presented a hybrid method consisting of the ant colony optimization and GA, where genetic operators were applied to evolve new generation of ants. Lam et al. [18] used the gradient method to make up for the deficiency of GA in local search. Yuce et al. [19] first used the bees algorithm to obtain the best local results, and then ran a GA to find the final results, which improved the weakness of the bees algorithm in the global search.

Kumar and Vidyarthi [20] presented an approach by combining particle swarm optimization algorithm (PSO) and GA, where the PSO algorithm is used first to reduce the solution space, and then the GA is applied to search the optimal result in the new solution space. Javidrad and Nazari developed a hybrid approach by mixing the PSO and simulated annealing (SA) algorithms, which first applies the PSO algorithm to find the optimal results, and then switches over to the SA algorithm the PSO fails. If the SA algorithm can find a better result, the search is turned to the PSO algorithm. That is, this approach uses two algorithms alternatively for finding the best result.

The GA, PSO, and SA are the probabilistic heuristics, the bottom left fill (BLF) and the large first (LF) are the deterministic heuristic. Jackobs [21] used the BLF algorithm to solve the CP problem for the first time. The LF was used in the work of Dowsland et al. [22]. Deterministic heuristic will converge always to the same result, every run. It was shown that deterministic heuristics fail at specific configurations to get the global optimum [23]. For this reason, some literature combines probabilistic heuristics and deterministic heuristics in the work. Burke et al. [24, 25] and Fırat and Alpaslan [26] used the BLF algorithm to process the placement of pieces and the SA algorithm to optimize the layout result.

3 Problem Description

3.1 HSFL Problem

The HSFL problem is defined as the job of finding the best layout combination of several sets of pieces on the horizontal striped fabric. Difficulties that need to be addressed on HSFL: how to solve the stripe alignment issues, how to ensure the highest fabric utilization, and how to obtain the layout with the minimal computing time.

3.2 Mathematical Modeling

It is supposed that we a layout of M sets of clothing of size i on a striped fabric of width W and length . The purpose is to minimize the total cost, which is composed of the cost of fabric C_f, setup cost C_s and time cost C_t. The notations used in this study are listed in Table 1.

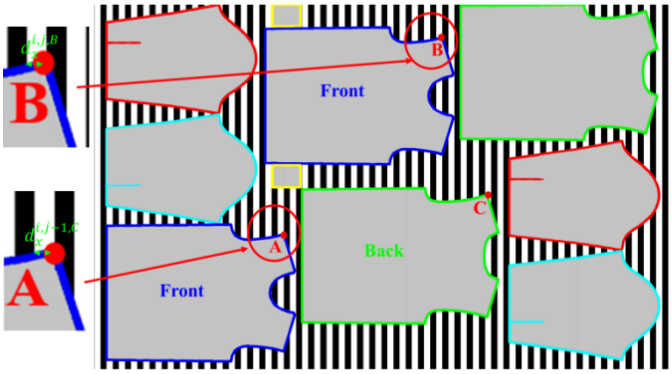


Fig. 1. How to make the stripe align

Table 1. The notations used in our study

Notations
Indices:
i : clothing size ($i = xs, s, m, l, xl, xxl$)
j : number of the pieces
k : number of key points on a piece
Parameters:
J : maximum number of pieces of a clothing
M : maximum sets of clothing used in the layout
L : length of fabric that can accommodate a complete set of pieces
W : width of fabric that can accommodate a complete set of pieces
Variables:
l_{ij} : maximum length of piece j of size i
W_{ij} : maximum width of piece j of size i
S_{ij} : area of piece j of size i
A_{ij} : area of the rectangle bounding box corresponding to piece j of size i
$d_x^{i,j,k_{\max}}$: x-coordinate of the upper left point $k_{\max}$ on piece j of size i
$d_x^{i,j,k_{\min}}$: x-coordinate of the lower right point $k_{\min}$ on piece j of size i
$d_y^{i,j,k_{\max}}$: y-coordinate of the upper left point $k_{\max}$ on piece j of size i
$d_y^{i,j,k_{\min}}$: y-coordinate of the lower right point $k_{\min}$ on piece j of size
z_j : $z_j = 1$, if piece j is used; $z_j = 0$, otherwise
$k_{\max}$: the upper left point
$k_{\min}$: the lower right point
$d_y^{i,j,k}$: x-coordinate of the upper left point k on piece j of size i
$d_y^{i,j-1,k'}$: x-coordinate of the upper left point k' on piece $j-1$ of size i
Constants:
F : total cost of a set of pieces layout
F_t : objective function (total cost of all sets of pieces layout)
C_S : setup cost related to the spreading of the fabric and fixing of the pieces
C_f : cost of the fabric per unit of area
C_t : time cost associated with the layout of all pieces

The HSFL focuses on minimizing the total cost as follow:
Minimize:

$$F = F_t \times M. \tag{1}$$

$$F_t = C_f (L \times W - M \sum_{i}^{J} S_{ij}) + (C_s + C_t) z_j. \tag{2}$$

Subject to:
Length and width constraints:

$$\max\{l_{ij}\} \le L, \forall j \in (1,2,3,...,J). \tag{3}$$

$$\max\{w_{ij}\} 7 \le W, \forall j \in (1,2,3,...,J). \tag{4}$$

Area constraint:

$$M \sum_{j}^{J} S_{ij} \le L \times W. \tag{5}$$

Overlap constraint:

$$d_x^{i,j,k} - d_x^{i,j-1,k'} > 0. \tag{6}$$

$$d_y^{i,j,k} - d_y^{i,j-1,k'} > 0. \tag{7}$$

Integrality constraints:

$$M = \frac{L \times W}{\sum_{j=1}^{5} A_{ij}}. \tag{8}$$

$$A_{ij} = (d_x^{i,j,k_{\max}} - d_x^{i,j,k_{\min}}) \times (d_y^{i,j,k_{\max}} - d_y^{i,j,k_{\min}}). \tag{9}$$

We use equation (1) to compute the total objective function which is the product of the single set total cost F and the total used sets M. In equation (2), the first term is the fabric scrap cost, which is obtained by multiplying the cost of fabric (C_f) with the fabric scrap ($L \times W - M \sum_j^J S_{ij}$). In the second term, C_s is the total setup cost and C_t is the time cost spent by a set of clothing. To minimize the total cost, our model must satisfy the following constraints: the length and width based constraints (3) and (4), respectively; the layout area based constraint (5); the overlap constraint (6) and (7); integrality constraints (8) and (9).

4 Methods for the HSFL Problem

The proposed methods for the HSFL problem are discussed next. We combine the top-left fill (TLF) algorithm based on the key points positioning strategy with GA algorithm. The flowchart of the proposed methods is shown in Fig. 2.

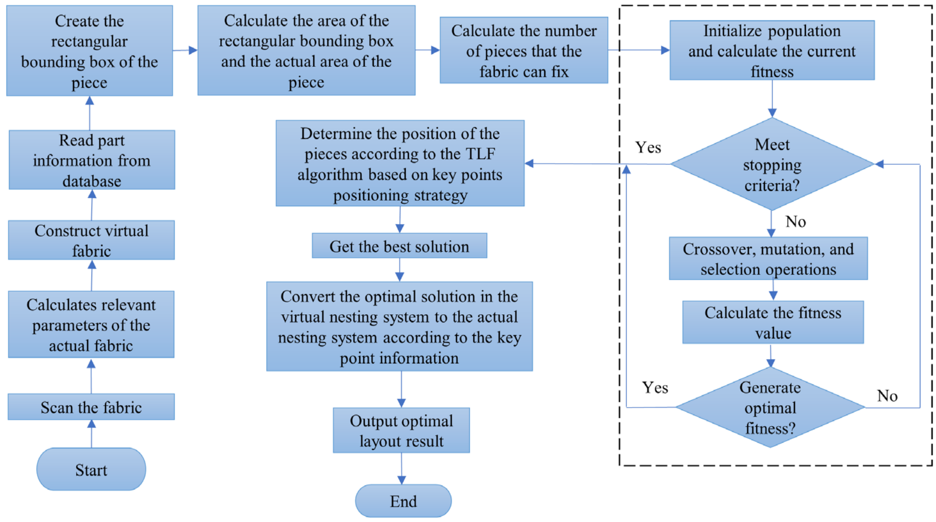


Fig. 2. The flowchart of the proposed methods

4.1 The rRectangle Bounding Box Algorithm

The rectangle bounding box algorithm defines the rectangular box bounding for the piece. This algorithm avoids the defining of threshold and deleting vertices, which accelerates the calculation and becomes more concise than the Ramer-Douglas-Peucker algorithm. It is to discretize irregular curves into straight lines and edges with a certain accuracy, and then convert irregular graphics containing curves into polygons. The Ramer-Douglas-Peucker algorithm use a few vertices to describe pieces will lose accuracy and produce errors to a certain extent.

We compare the coordinate values of the key points on the pieces to find the upper left and lower right points, and then use those two points to construct a rectangular bounding box. Fig. 3 shows the result of the rectangle bounding box algorithm.

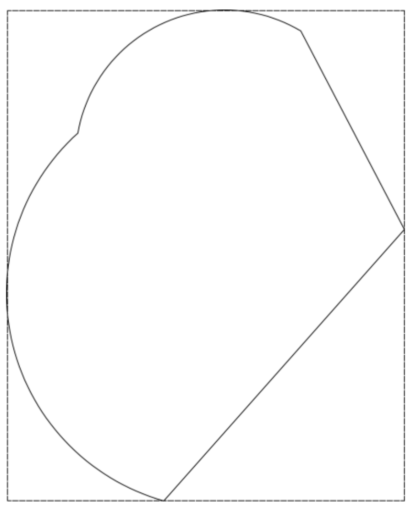


Fig. 3. The rectangle bounding box algorithm

4.2 The TLF Algorithm Based on the Key Points Positioning Strategy

The position constraint brought by the stripes is a problem that cannot be ignored in the layout process. According to the production requirements of apparel enterprises, the position constraint means that the proposed approach must meet the following: the alignment of the related pieces, the interchangeability of the same type of pieces, and the consistency of the stripes of the pieces in the same roll.

Considering the above requirements, we propose to take the key points of the first set of pieces as a reference and place other pieces into the subsequent layout. Therefore, we put forward the TLF algorithm based on key points positioning strategy.

The Key Points Positioning Strategy. Assuming that the left edge of the stripe is the starting edge, we establish a striped coordinate system that is discretized by the stripes with fixed-width. The top-left corner is the coordinate origin, the horizontal direction to the right is the positive direction of x-axis, and the vertical direction to the down is the positive direction of y-axis. We define vertexes of piece as key points and construct a key points sequence ($\{(d_x^{i,j,k}, d_y^{i,j,k})\}$) for every piece. For example, the key points sequence of piece 1 is $\{(d_x^{i,1,1}, d_y^{i,1,1}), (d_x^{i,1,2}, d_y^{i,1,2}), (d_x^{i,1,3}, d_y^{i,1,3}), (d_x^{i,1,4}, d_y^{i,1,4}), (d_x^{i,1,5}, d_y^{i,1,5})\}$. Fig. 4 shows the key points on the piece 1. $d_x^{i,j,k}$ is the x-coordinate, $d_y^{i,j,k}$ is the y-coordinate.

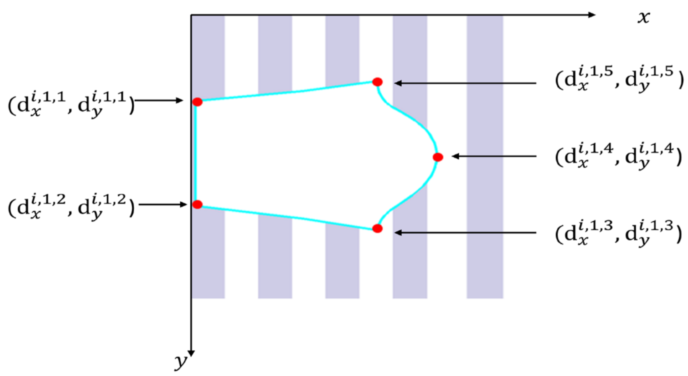


Fig. 4. Key points of the piece 1

The TLF Algorithm Based on the Key Points Positioning Strategy. Since the coordinate origin of the striped coordinate system is top-left corner, the traditional BLF algorithm is changed to TLF. The TLF algorithm based on the key points positioning strategy means that a key points sequence is constructed for each piece. In order to ensure the interchangeability of the same type of piece, we store the key points sequences of the first set of pieces as a reference for subsequent layouts. When placing into a new piece, compare the key points sequence of the new piece with the first set of pieces, adjust the position according to the comparison results and finally determine its positioning.

The steps of the TLF algorithm based on the key points positioning strategy are as follows:

Step1: Build the key points sequence $P_{i,j}$ for each piece $P_{i,j} = \{(d_x^{i,j,1}, d_y^{i,j,1}), (d_x^{i,j,2}, d_y^{i,j,2}), (d_x^{i,j,3}, d_y^{i,j,3}), (d_x^{i,j,4}, d_y^{i,j,4}), (d_x^{i,j,5}, d_y^{i,j,5})\}, \ldots, (d_x^{i,j,n}, d_y^{i,j,n})$, $i = xs, s, m, l, xl, xxl$; $j = 1, 2, 3, \ldots, J$. It should be noted that when the pieces are not placed, key points are initialized to (0,0);

Step 2: Place into piece from the top-left corner, update and record its key points sequence $P_{i,1}$;

Step 3: Place into a new piece. Compare the top-left point of the new piece with all the points of $P_{i,1}$, judge whether there are overlapping, if no, place it, record and update the key points sequence $P_{i,2}$; otherwise, place another new piece;

Step 4: repeat step 3 until all pieces are placed.

4.3 Overlapping Evaluation

Both no-fit polygon and inner-fit polygon can solve the overlapping problem in the layout problem [27, 28]. For two pieces, the no-fit polygon and inner-fit polygon describe all the overlapping placements. To obtain the no-fit polygon, one of the pieces is fixed and the other as the movable piece. The no-fit polygon is all the translations that place the movable piece to overlap the fixed piece [29]. The inner-fit polygon is derived from the no-fit polygon, the difference between them is the inner-fit polygon represents the translations by placing the movable piece completely inside a container [30].

The no-fit polygon and inner-fit polygon are mainly relying on the point contact. Based on the two concepts, we propose the key points to judge the overlapping. For two pieces, firstly find the most convex point of the new piece, calculate the distance between the most convex point with all the points of the placed piece. When all the distances are greater than zero, there will be no overlapping.

4.4 The GA Algorithm

After applying the key points positioning strategy to solve the stripe alignment problem, it is also necessary to choose a suitable optimization approach to improve the utilization rate of the fabric.

As per the observations of Tsao *et al.* [10], we choose the genetic algorithm (GA) to optimize the layout. The GA algorithm is a meta-heuristic algorithm that mimics the biological evolution, first proposed by Professor Holland in 1970s based on the Darwin's evolutionary theory [8, 15, 31]. The GA algorithm evolves a completely random individual population (the solution space) to obtain a better individual (the optimal solution). The GA algorithm flow chart as shown in the dotted box on the right of Fig. 3 and the vital elements in the GA algorithm as follows:

Encoding. Encoding converts a solution of the problem into a chromosome that can be understood by the GA algorithm. We use sort encoding according to the area of the pieces. Fig. 5 shows the coding rules in detail. We sort 5 pieces as per their actual area $S_{i,j}$. The number starts from 0, for example, the number of the largest piece is 0, that is $j = 0$.

Initial Population. The vital step of the GA algorithm is to generate the initial population, which is randomly composed of a certain number of individuals or artificially set according to the performance of the optimal solution [2, 15]. The size of the initial population depends on the scale of the layout problem. It generally cannot be less than 10, and should not be greater than 1000. In this paper, we set the initial population size to $5M$ and use the pre-layout obtained by the BLF algorithm as the initial population.

Fitness Function. The important foundation of population evolution is the fitness function. According to Gonçalves [32], the fitness function can be defined by the quality of a solution. In this work, we use the objective function F as the fitness function.

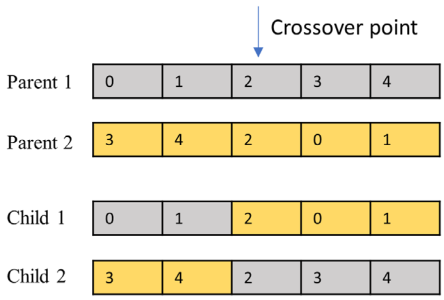
Genetic Operator. The genetic operators are mutation, crossover, and selection, which simulate chromosome evolution in three ways. These operators should be repeated before the stopping criteria are reached [15].

First of all, the selection operator determines the individuals that are suitable for an inheritance to descendants. This article uses the roulette wheel selection strategy. The selection probability P_m of the m-th individual is calculated by equation (10).

$$P_m = \frac{F_m}{\sum_m F_m} \times 100\%. \quad \textbf{(10)}$$

where, F_m is the fitness value of the m-th individual.

Secondly, the crossover operation refers to exchanging some genes between two homologous chromosomes to form two new individuals. We adopt the single-point crossover and two-point crossover operators. Their principles are shown in Fig. 6 and Fig. 7, respectively.



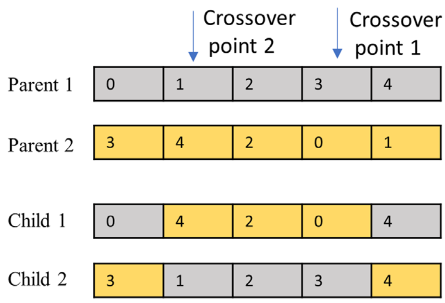


Fig. 6. Exchange of genes by single-point crossover operator **Fig. 7.** Exchange of genes by two-point crossover operator

Finally, the mutation is the operation that changes some genes of a chromosome to produce a new one. In the mutation operation, two homologous chromosomes and their mutation positions are selected randomly. Then, the two mutation positions are swapped to generate two new individuals.

Stopping Criteria. The termination condition of the GA algorithm can be that the best fitness of the individual has reached the threshold or the evolution has been performed for the specified maximum number of generations [15, 33]. Generally, to control the computational time, the maximum evolutionary generations should be limited to 100-600.

The GA algorithm has strong robustness in solving combinatorial optimization problems. Mutation operator is an approach that escapes the local optimal solutions. It is different from other algorithms in the sense that it finds the best solution among all the feasible solutions and avoids generating local optimal solutions [15].

5 Results

5.1 Experimental Environment and Parameters

The experimental result is an important index to evaluate the feasibility of the algorithm. All the algorithms are coded in C++ and compiled in Visual Studio 2010. The tests are performed in a PC equipped with a 2.11GHz Intel i5 CPU and 12GB RAM.

We define the objection function F as the fitness function of the GA algorithm. Fig. 8 shows that the fitness value fluctuates up and down when the number of iterations is small, and it is stabilized after 300 iterations. Therefore, the GA algorithm is allowed to evolve for 300 iterations. According to the theory of Soares and Vieira [33], the recommended parameters of the GA algorithm values are shown in Table 2. We adjust the parameters according to the instances of this study.

Table 2. The selected parameter of the GA algorithm

Parameters	Suggested value	Value used in the proposed approach
Population size	≥ 100	$5M$
Crossover rate	$0.6 \leq P_c \leq 0.99$	0.8
Mutation rate	$0.01 \leq P_m \leq 0.01$	0.01
Crossover operator	Two points	Two-point and single-point
Number of iterations	100-600	300

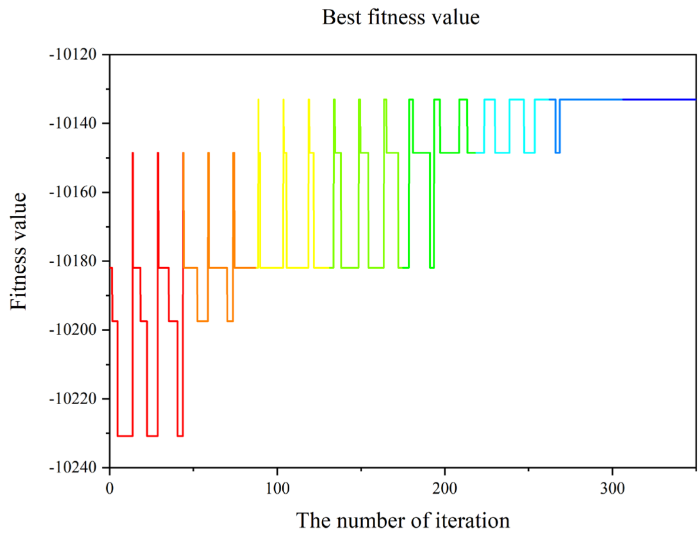


Fig. 8. Relationship between fitness value and generation

5.2 Datasets

Some classical benchmark instances were tested and verified by massive approaches in the literature. The shapes of most of those instances are used to simulate the shapes involved in the apparel industry [34]. We selected 15 instances to evaluate the proposed approach. In order to evaluate these instances by the proposed approach, we adjusted the width of the white and black stripe to 5 mm. These 15 instances were presented in [21], which were collected by ESICUP (EURO Special Interest Group on Cutting and Packing, https://www.euro-online.org/web-sites/esicup/data-sets). In the testing dataset, we also added a Stripedshirts instance taken from a Chinese apparel industry. The shapes of the Stripedshirts are shown in Fig. 2. The features of the 16 instances are clearly listed in Table 3.

Table 3. The datasets information

No.	Dataset	Number of different pieces	Total number of pieces	Feasible orientations (°)	Average vertices by pieces
1	Albano	8	24	0, 180	7.43
2	Dagli	10	30	0, 180	5.60
3	Trousers	17	64	0, 180	5.06
4	Shirts	8	99	0, 180	6.63
5	Shapes0	4	43	0	8.75
6	Shapes1	4	43	0, 180	8.75
7	Shapes2	7	28	0, 180	6.29
8	Swim	10	48	0, 180	21.90
9	Dighe1	16	16	0	3.87
10	Dighe2	10	10	0	4.70
11	Fu	12	12	0, 90, 180, 270	3.58
12	Jakobs1	25	25	0, 90, 180, 270	5.60
13	Jakobs2	25	25	0, 90, 180, 270	5.36
14	Mao	9	20	0, 90, 180, 270	9.22
15	Marques	8	24	0, 90, 180, 270	7.37
16	Stripeshirts	4	5	0, 180	6.40

5.3 Experimental Result

Five sets of published results were considered for comparing the proposed approach. Hopper [35] collected and presented 18 benchmark instances for the CP problem. After a few years, Egeblad *et al.* [36], Bennell and Song [37], Shalaby and Kashkoush [34], and Sato *et al.* [30] proposed new heuristic solution methods for the CP problem and presented their experimental results.

In Table 4, we compare the best utilization obtained by the proposed approach with the five sets of published results taken from literature. The best utilization is marked by gray color. It can be seen that the proposed approach could perform better in 12 out of the 16 instances with an inferior difference of 1%, thus making the proposed approach a very competitive one. The proposed approach could obtain the best utilization for the Marque instance with an increase of 2.4%. The Swim, Shapes0, Shapes1, and Mao performed very poorly as four instances have more number of vertices which were challenging for the HSFL model.

The execution time is another evaluation criterion of the solutions. In the comparison of the execution time, the number of iterations and computer configuration need to be considered. Hence, for a fair comparison, we run those approaches for 30 iterations in the same computer configuration. In Table 5, we compare the execution time of the proposed approach with those of the considered five approaches. The least execution times are underlined. Hopper [35] and Egeblad et al. [36] took the highest execution time. The proposed approach took the least execution time on 10 out of the 16 instances. Fig. 9 focuses on the examples of Stripedshirts from China's apparel industry.

Table 4. Maximum utilization (%) for benchmark instances obtained by the proposed approach and other best solutions in the literature

No.	Datasets	Algorithms					
		Hopper (2000)	Egeblad *et al.* (2007)	Bennell & Song (2010)	Shalaby & Kashkoush (2013)	Sato *et al.* (2019)	The proposed approach
1	Albano	84.09	87.43	87.88	83.36	89.06	91.63
2	Dagli	77.1	87.15	87.99	83.97	88.73	90.79
3	Trousers	79.12	89.96	90.38	88.36	91.06	92.77
4	Shirts	79.65	86.79	89.69	86.26	88.52	90.78
5	Shapes0	57.83	66.50	64.35	65.41	68.79	68.32
6	Shapes1	62.25	71.25	72.55	71.25	76.73	74.45
7	Shapes2	63.97	83.60	81.29	74.74	83.61	85.47
8	Swim	67.42	74.37	75.04	68.25	75.66	73.24
9	Dighe1	71.01	100	100	100	100	100
10	Dighe2	70.23	100	100	100	100	100
11	Fu	83.82	90.96	90.28	89.06	92.31	93.37
12	Jakobs1	74.25	78.89	85.96	81.67	89.09	93.21
13	Jakobs2	68.4	77.28	80.40	77.20	87.73	90.57
14	Mao	68.65	82.54	84.07	78.4	86.05	85.27
15	Marques	82.73	88.14	88.92	86.47	91.02	93.42
16	Stripedshirts						87.52

Table 5. Average execution time (s) in 300 runs for benchmark instances obtained by the proposed approach and other best solutions in the literature

No.	Datasets	Algorithms					
		Hopper (2000)	Egeblad *et al.* (2007)	Bennell & Song (2010)	Shalaby & Kashkoush (2013)	Sato *et al.* (2019)	The proposed approach
1	Albano	25.32	23.27	24.32	24.14	24.05	<u>21.32</u>
2	Dagli	28.76	26.12	26.37	28.39	<u>25.15</u>	27.45
3	Trousers	20.12	23.69	22.01	20.13	21.29	<u>18.52</u>
4	Shirts	22.31	21.03	<u>16.12</u>	20.04	16.41	20.58
5	Shapes0	28.33	27.15	27.24	26.13	20.39	<u>19.46</u>
6	Shapes1	24.67	24.11	23.15	21.07	<u>18.31</u>	20.21
7	Shapes2	27.21	23.05	<u>17.84</u>	22.72	19.25	21.36

8	Swim	27.42	26.91	26.24	24.25	20.14	19.42
9	Dighe1	11.43	10.21	10.24	9.30	8.25	5.21
10	Dighe2	10.67	10.37	10.03	9.17	8.31	4.70
11	Fu	23.82	22.31	23.35	21.52	19.79	16.34
12	Jakobs1	28.23	25.23	25.11	24.15	20.41	22.46
13	Jakobs2	28.33	24.78	24.15	23.79	24.85	24.52
14	Mao	23.54	21.20	20.29	21.58	18.49	17.24
15	Marques	21.23	23.41	21.04	19.31	19.24	15.21
16	Stripedshirts	21.66	21.17	21.41	20.28	19.41	15.06

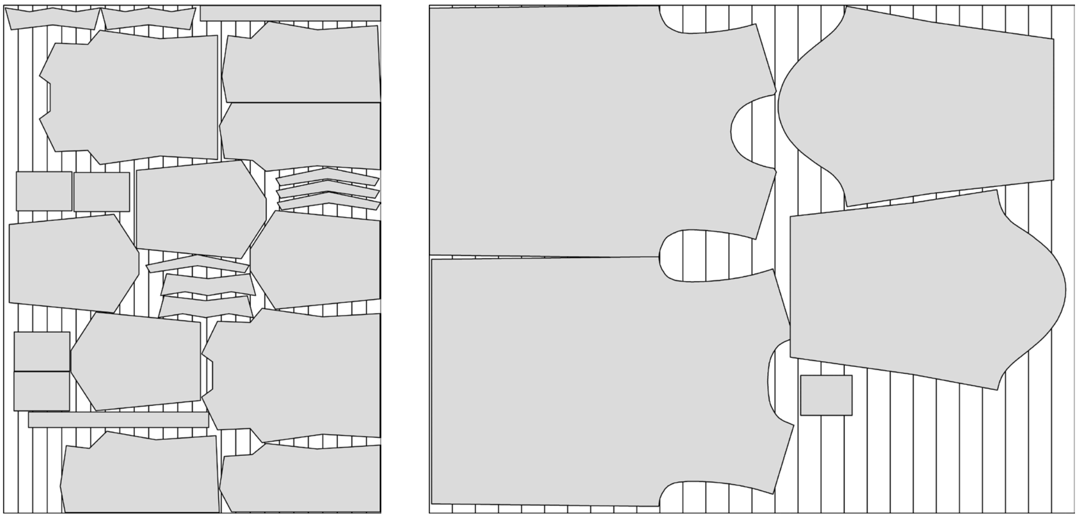


Fig. 9. The layouts for the Stripedshirts datasets

6 Conclusion

In this paper, we study a CP variant with position constraints, the HSFL problem. For this practical problem, we establish an integer linear mathematical model and propose an approach to solve it. We process the parameters of fabric and pieces based on the machine vision system.

Different from the existing layout algorithms, we propose the TLF algorithm based on key points positioning strategy, and combine it with GA algorithm to solve the layout problem with position constraints.

The proposed approach has been tested on 15 datasets available in literature on the irregular strip packing problem and a research instance of this study. Comparing the computational results for these datasets, we can conclude that the proposed approach has the potential to minimize the wastes and computing time. Also, it can solve the CP problems with position constraints. However, this algorithm is not suitable for layouts of multi-size pieces. In future research, the development of algorithms can be strengthened in these fields.

Acknowledgement

The authors would like to express their gratitude to EditSprings (https://www.editsprings.com/) for the expert linguistic services provided.

Data Availability

The experimental data used to support the findings of this study are included in the article. The Stripedshirts instance will be supplied by the file named “Supporting Information”.

References

[1] S.W. Sung, Y.J. Jang, Heuristic for the assort-packing and distribution problem in the fashion apparel industry, International Journal of Production Research 56(9)(2018) 3116-3133.

[2] Y. Xu, S. Thomassey, X. Zeng, Optimization of garment sizing and cutting order planning in the context of mass customization, The International Journal of Advanced Manufacturing Technology 106(7-8)(2020) 3485-3503.

[3] X. Shang, D. Shen, F.-Y. Wang, T.R. Nyberg, A Heuristic Algorithm for the Fabric Spreading and Cutting Problem in Apparel Factories, IEEE/CAA Journal of Automatica Sinica 6(4)(2019) 961-968.

[4] E. Enes, S. Kipöz, The role of fabric usage for minimization of cut-and-sew waste within the apparel production line: Case of a summer dress, Journal of Cleaner Production 248(2020) 119221.1-119221.12.

[5] T.-J. Wang, J.-Y. Peng, Y.-F. Hung, Modeling Fabric Cutting Scheduling as Mixed Integer Programming, in: Proc. IEEE International Conference on Industrial Engineering and Engineering Management, 2016.

[6] K.-T. Tung, C.-Y. Chen, Y.-F. Hung, Solving Cutting Scheduling Problem by Simulated Annealing Search Method, in: Proc. 2016 IEEE International Conference on Industrial Engineering and Engineering Management, 2016.

[7] G. Wäscher, H. Haußner, H. Schumann, An improved typology of cutting and packing problems, European Journal of Operational Research 183(3)(2007) 1109-1130.

[8] R. M’Hallah, A. Bouziri, Heuristics for the combined cut order planning two-dimensional layout problem in the apparel industry, International Transactions in Operational Research 23(1-2)(2016) 321-353.

[9] A.K. Sato, T.C. Martins, M.S.G. Tsuzuki, Massive parallelization accelerated solution for the 2D irregular nesting problem, IFAC-PapersOnLine 52(10)(2019) 119-124.

[10] Y.C. Tsao, T.L. Vu, L.W. Liao, Hybrid Heuristics for the Cut Ordering Planning Problem in Apparel Industry, Computers & Industrial Engineering 144(2020) 106478.

[11] F.M.B. Toledo, M.A. Carravilla, C. Ribeiro, J.F. Oliveira, A.M. Gomes, The Dotted-Board Model: A new MIP model for nesting irregular shapes, International Journal of Production Economics 145(2)(2013) 478-487.

[12] A.A.S. Leao, F.M.B. Toledo, J.F. Oliveira, M.A. Carravilla, A semi-continuous MIP model for the irregular strip packing problem, International Journal of Production Research 54(3)(2016) 712-721.

[13] K.S. Saravanan, A. Rajasekhar, N.R. Babu, An efficient collision avoidance and neighborhood search algorithm for nesting of two-dimensional parts in two-dimensional sheets, Journal of Advanced Manufacturing Systems 11(1)(2012) 51-65.

[14] R. Akunuru, N.R. Babu, A semi-discrete geometric representation for nesting problems, International Journal of Production Research 51(14)(2013) 4155-4174.

[15] D. Domović, T. Rolich, M. Golub, Evolutionary hyper-heuristic for solving the strip-packing problem, The Journal of the Textile Institute 110(8)(2019) 1141-1151.

[16] J. Martens, Two genetic algorithms to solve a layout problem in the fashion industry, European Journal of Operational Research 154(1)(2004) 304-322.

[17] J. Liu, S. Xu, F. Zhang, L. Wang, A Hybrid Genetic-Ant Colony Optimization Algorithm for the Optimal Path Selection, Intelligent Automation & Soft Computing 23(2)(2017) 235-242.

[18] Y.C. Lam, Y.M. Deng, C.K. Au, A GA/gradient hybrid approach for injection moulding conditions optimisation, Engineering with Computers 21(3)(2006) 193-202.

[19] B. Yuce, F. Fruggiero, M.S. Packianather, D.T. Pham, E. Mastrocinque, A. Lambiase, M. Fera, Hybrid Genetic Bees Algorithm applied to single machine scheduling with earliness and tardiness penalties, Computers & Industrial Engineering 113(2017) 842-858.

[20] N. Kumar, D.P. Vidyarthi, A novel hybrid PSO–GA meta-heuristic for scheduling of DAG with communication on multiprocessor systems, Engineering with Computers 32(1)(2016) 35-47.

[21] S. Jakobs, On genetic algorithms for the packing of polygons, European Journal of Operational Research 88(1)(1996) 165-181.

[22] K.A. Dowsland, S. Vaid, W.B. Dowsland, An algorithm for polygon placement using a bottom-left strategy, European Journal of Operational Research 141(2)(2002) 371-381.

[23] T.C. Martins, M.S.G. Tsuzuki, Simulated annealing applied to the irregular rotational placement of shapes over containers with fixed dimensions, Expert Systems with Applications 37(3)(2010) 1955-1972.

[24] E.K. Burke, G. Kendall, G. Whitwell, A New Placement Heuristic for the Orthogonal Stock-Cutting Problem, Operations Research 52(4)(2004) 655-671.

[25] E.K. Burke, G. Kendall, G. Whitwell, A Simulated Annealing Enhancement of the Best-Fit Heuristic for the Orthogonal Stock-Cutting Problem, INFORMS Journal on Computing 21(3)(2009) 505-516.

[26] H. Fırat, N. Alpaslan, An effective approach to the two-dimensional rectangular packing problem in the manufacturing industry, Computers & Industrial Engineering 148(2020) 106687.
[27] S.C.H. Leung, Y. Lin, D. Zhang, Extended local search algorithm based on nonlinear programming for two-dimensional irregular strip packing problem, Computers and Operations Research 39(3)(2012) 678-686.
[28] T. Imamichi, M. Yagiura, H. Nagamochi, An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem, Discrete Optimization 6(4)(2009) 345-361.
[29] L. Queiroz, M. Andretta, Two effective methods for the irregular Knapsack problem, Applied Soft Computing 95(2020) 106485.
[30] A.K. Sato, T.C. Martins, A.M. Gomes, M.S.G. Tsuzuki, Raster penetration map applied to the irregular packing problem, European Journal of Operational Research 279(2)(2019) 657-671.
[31] S. Vitayasak, P. Pongcharoen, C. Hicks, A tool for solving stochastic dynamic facility layout problems with stochastic demand using either a Genetic Algorithm or modified Backtracking Search Algorithm, International Journal of Production Economics 190(2016) 146-157.
[32] J.F. Gonçalves, A Hybrid Biased Random Key Genetic Algorithm for a Production and Cutting Problem, IFAC PapersOnLine 48(3)(2015) 496-500.
[33] M.M. Soares, G.E. Vieira, A new multi-objective optimization method for master production scheduling problems based on genetic algorithm, The International Journal of Advanced Manufacturing Technology 41(5-6)(2009) 549-567.
[34] M.A. Shalaby, M. Kashkoush, A Particle Swarm Optimization Algorithm for a 2-D Irregular Strip Packing Problem, American Journal of Operations Research 3(2)(2013) 268-278.
[35] E. Hopper, B. Turton, An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem, European Journal of Operational Research 128(1)(2007) 34-57.
[36] J. Egeblad, B.K. Nielsen, A. Odgaard, Fast neighborhood search for two- and three-dimensional nesting problems, European Journal of Operational Research 183(3)(2007) 1249-1266.
[37] J.A. Bennell, X. Song, A beam search implementation for the irregular shape packing problem, Journal of Heuristics 16(2)(2010) 167-188.