

A Heuristic Algorithm for the Fabric Spreading and Cutting Problem in Apparel Factories

Xiuqin Shang, Dayong Shen, Fei-Yue Wang, *Fellow, IEEE*, and Timo R. Nyberg

Abstract—We study the fabric spreading and cutting problem in apparel factories. For the sake of saving the material costs, the cutting requirement should be met exactly without producing additional garment components. For reducing the production costs, the number of lays that corresponds to the frequency of using the cutting beds should be minimized. We propose an iterated greedy algorithm for solving the fabric spreading and cutting problem. This algorithm contains a constructive procedure and an improving loop. Firstly the constructive procedure creates a set of lays in sequence, and then the improving loop tries to pick each lay from the lay set and rearrange the remaining lays into a smaller lay set. The improving loop will run until it cannot obtain any smaller lay set or the time limit is due. The experiment results on 500 cases show that the proposed algorithm is effective and efficient.

Index Terms—Construction and improvement, cutting and packing, fabric spreading and cutting, heuristic algorithm.

I. INTRODUCTION

CUTTING and packing problems [1], [2] are classic combinatorial optimization problems that address the optimal utilization of resources. Cutting problems regard the best use of materials such as cloth, paper, wood, and steel. The efficient use of material contributes to the economical utilization of natural resources. It is also of great economic importance in production processes. In this paper, we consider the fabric spreading and cutting problem (FSCP) in apparel factories which is a special case of cutting problems. Given the production plan of producing a certain number of garments of various styles and sizes, FSCP consists of spreading a set of avail-

Manuscript received October 9, 2018; revised November 15, 2018; accepted November 28, 2018. This work was supported in part by the National Key Research and Development Program of China (2018YFB1702701), the National Natural Science Foundation of China (61773381, 61773382, 61533019, 61702519), Dongguan's Innovation Talents Project (Gang Xiong), Guangdong's Science and Technology Project (2017B090912001), Beijing Natural Science Foundation (4182065), and Chinese Hunan's Science and Technology Project (20181040). Recommended by Associate Editor MengChu Zhou. (Corresponding author: Xiuqin Shang.)

Citation: X. Q. Shang, D. Y. Shen, F. -Y. Wang, and T. R. Nyberg, "A heuristic algorithm for the fabric spreading and cutting problem in apparel factories," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 4, pp. 961–968, Jul. 2019.

X. Q. Shang is with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing 100190, China (e-mail: xiuqin.shang@ia.ac.cn).

D. Y. Shen is with Qingdao Academy of Intelligent Industries, Qingdao 266109, China (e-mail: dayong.shen@qii.ac.cn).

F. -Y. Wang is with the Beijing Engineering Research Center of Intelligent Systems and Technology, Institute of Automation, CAS, Beijing 100190, China; and also with the Cloud Computing Center, CAS, Dongguan 523808, China (e-mail: feiyue.wang@ia.ac.cn).

T. R. Nyberg is with the School of Science at Aalto University, Aalto, Finland (e-mail: timo.nyberg@aalto.fi).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JAS.2019.1911573

able fabrics on a cutting bed and cutting them into garment components. The consumption of fabrics as well as the use of the cutting bed should be minimized [3]–[5]. FSCP is an NP-complete combinatorial problem for the solution space rises exponentially with the number of required garment styles and sizes, and fabric types and colors [3]. Existing algorithms mainly focus on middle and small instances [5], [6]. For large-scale apparel factories often receive large orders in a production cycle, we will focus on the middle and large instances in this paper. Section II brings out a brief review on FSCP. Section III describes the mathematical model for FSCP and proposes a heuristic algorithm for solving it. Section IV evaluates the proposed algorithm. Section V concludes the paper.

II. LITERATURE REVIEW

Cutting and packing problems form an old and very well-known family, called CP (cutting & packing) in Dyckhoff [1], Sweeney *et al.* [2], and Wascher *et al.* [7]. CP has received growing attention in numerous real-world applications such as computer science, industrial engineering, logistics, manufacturing, and so on in recent years. As a branch of CP, FSCP has not yet obtained sufficient attention in academic research although it plays an important role in industrial application. Some early publications focus on minimizing the time for fabric spreading and cutting in the garment industry. Wong *et al.* [8] present a spreading and cutting sequencing model using GA to solve the sequencing problem of the spreading and cutting system. Wong [6] proposes a generic optimized table-planning model to optimize apparel manufacturing resource allocation. To minimize the sum of the cutting cost and fabric cost, Jacobs-Blecha *et al.* [3] suggest a method to solve the cut order planning problem for apparel manufacturing. They consider the cutting order that only contains garments of the same fabric type and color, and can be completed in a lay with layers of different lengths. Degraeve and Vandebroek [9] formulate the problem as a mixed integer programming problem where the cutting setup costs and excess production are examined. They assume that all templates (referred to as stencils) have the same length independent of the garment size. All cutting pattern types should be enumerated in the proposed method. Degraeve *et al.* [10] extend this work by using a non-linear integer programming solver. Rose and Shier [4] suggest a tree search method for cut scheduling in the apparel industry. By this method, the number of lays is minimized while no excess garment components are produced. They assume that the requirement contains only garments with the same fabric type

and color, and all templates have the same length independent of the garment size. Nascimento *et al.* [5] propose a state-space approach to solve the fabric spreading and cutting problem. By assuming a fixed number of lays, the objective is to minimize the comprehensive cost of the fabric spreading and cutting process. The number of cutting patterns will grow exponentially when the number of garment styles and sizes (or the quantity of fabric types and colors) increases. The above methods are effective and efficient for solving the small scale fabric spreading and cutting problem. But they will suffer the situation called state explosion and become powerless when solving middle and large scale problems that exist in many factories. Toscano *et al.* [11] discuss the two dimensional cutting stock problem that occurs in small-scale furniture factories. They bring out a heuristic approach to minimize both the number of stocks and the number of saw cycles. Hereby a saw cycle in a furniture factory is similar to a lay in a garment factory. The cutting stock problem with minimizing the number of stocks in a furniture factory can be considered as a three dimensional cutting stock problem with some constraints. If all the small items cannot rotate and have the same height as the one of the stock, this problem will be equivalent to the fabric spreading and cutting problem with single fabric type and color. Vanzela *et al.* [12] discuss the integrated lot sizing and cutting stock problem with saw cycle constraints in furniture factories. An effective method is brought out to reduce raw material waste as well as item (referred as pieces) production and inventory costs. The above two publications only consider stocks of the same type. In this paper the fabric spreading and cutting problem includes garments of various types and colors. We will study the fabric spreading and cutting problem in apparel factories and bring out a new algorithm. Wuttke and Heese [13] study the two-dimensional cutting stock problem with sequence dependent setup times. They formulate the sequencing problem as a mixed integer program (MIP) and derive a near-optimal algorithm. Foerster and Wascher [14] bring out a heuristic approach for solving the one-dimensional cutting stock problem with pattern reduction. The approach firstly creates a cutting plan then reduces the number of patterns used in this plan. Yanasse and Limeira [15] solve the cutting stock problem with pattern reduction in three steps. Patterns that will be used in the cutting plan are created in the first step. The complete cutting plan is obtained in the second step. The pattern combination method in Foerster and Wascher [14] is used in the third step to reduce the pattern number. Cui *et al.* [16] analyze the one-dimensional cutting stock problem with pattern reduction and solve it in two steps. A set of patterns are created in the first step, an integer linear programming (ILP) model is solved to minimize the sum of material and setup costs over the given pattern set in the second step. Cui *et al.* [17] present a heuristic algorithm for solving the two-dimensional arbitrary stock-size cutting stock problem. Ma *et al.* [18] formulate the combined cutting stock and lot-sizing problem with pattern setup as a mixed-integer linear programming model (MILP) and provide a dynamic programming-based heuristic (DPH) to solve it. Cui and Zhao [19] present a heuristic approach to solve the rectangular two-

dimensional single stock size cutting stock problem with two-staged patterns. In this approach the column-generation method is repeatedly applied to create patterns until all small items are produced. Alvarez-Valdés *et al.* [20] combine the greedy randomized adaptive search procedure (GRASP) and the path relink (PR) into an algorithm to solve the two-dimensional two-staged cutting stock problem. Some efficient approaches are proposed to solve the constrained two-staged two-dimensional cutting problem. These methods include the exact algorithm based on the bottom-up strategy by Hifi and M'Hallah [21], approximate algorithms based on strip generation by Hifi and M'Hallah [22], and the approach that combines strip-generation procedure with beam search by Hifi *et al.* [23]. The proposed algorithm performances continue to be improved. We will design an iterated greedy algorithm for solving FSCP. The above methods are beneficial references for some procedures such as dynamic programming and sequential heuristic are applied in our algorithm [24]–[29].

III. MATHEMATICAL MODEL AND HEURISTIC ALGORITHM

An apparel factory usually produces a number of different garment styles in a production cycle (e.g., week). Each of these styles consists of distinct couples of fabric types and garment sizes. Each of these couples is called a stock keeping unit (SKU) that corresponds to a certain garment style with the specific fabric type and garment size. The apparel manufacturing process involves three main steps, namely fabric spreading and cutting, garment sewing, and garment finishing. On the fabric spreading and cutting phase, the fabric rolls of different types and colors are spread over a cutting bed in many layers and a knife (or laser) follows a predetermined route to cut the fabrics into garment components. These components are then tied and put forward to the garment sewing step. The multiple layers of fabric on a cutting bed for a cut are called a lay. The detailed introduction to fabric spreading and cutting can be found in Nascimento *et al.* [5]. As shown in Fig. 1, each size of each garment style has a template that contains all of its garment components. All components of a template are positioned in a rectangular area in a way of ensuring minimum loss of fabric. The template of a specific garment style and size requires a fixed length of fabric. Templates are placed in a line to form a cutting pattern. Each lay is cut according to a cutting pattern. All SKUs that are cut simultaneously according to a template in a lay are called a column. Manufacturing costs in the fabric spreading and cutting stage account for a great proportion of the whole manufacturing costs. Manufacturing costs in this stage consist of the costs of using cutting machines, the costs of the waste cloth and etc. The costs of using cutting machines rely on the lay number while the costs of the waste cloth arise from the overcut garment components. For the sake of reducing the whole manufacturing costs, the scheduling of fabric spreading and cutting should be improved. Given the requirements of all SKUs in a production cycle, we try to generate a cutting plan with minimized lay number while all required garment components are cut without cutting any extra ones. Firstly, some necessary definitions are made in Table I.

With the above variables, FSCP can be illustrated as:

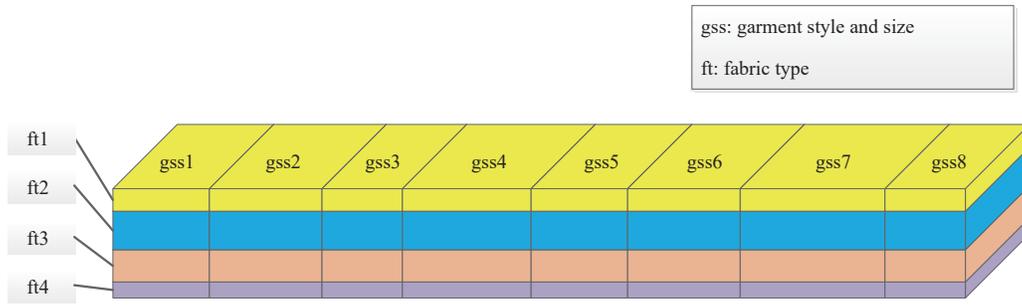


Fig. 1. A lay pattern for fabric spreading and cutting.

TABLE I
THE DEFINITIONS FOR FSCP

Parameter	Definition
l_{ub}	the maximum allowed fabric length of the cutting bed
h_{ub}	the maximum allowed fabric layer number of the cutting bed
g	the number of different garment figures, which is the product of the number of garment styles and the number of sizes of each garment style
f	the number of different fabric types
$s_{ij}(i = 1, 2, \dots, g; j = 1, 2, \dots, f)$	the requirement of the SKU with the i th garment figure and the j th fabric type
$l_i(i = 1, 2, \dots, g)$	the required fabric length of the i th garment figure
$s = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1f} \\ s_{21} & s_{22} & \dots & s_{2f} \\ \vdots & \vdots & \ddots & \vdots \\ s_{g1} & s_{g2} & \dots & s_{gf} \end{bmatrix}$	the matrix that stores the requirements of all the SKUs
$L = (l_1, l_2, \dots, l_g)$	the vector that stores the required fabric lengths of all the garment figures
a	the number of required lays
$q_{ki}(k = 1, 2, \dots, a; i = 1, 2, \dots, g)$	the number of the i th garment figure on the cutting pattern of the k th lay
$h_{kj}(k = 1, 2, \dots, a; j = 1, 2, \dots, f)$	the layer number of the j th fabric type in the k th lay
$V_k, k = 1, 2, \dots, a$	the volume of the k th lay
$UR_k, k = 1, 2, \dots, a$	the volume and utilization rate of the k th lay

$$\begin{aligned} \min a & \quad (1) \\ \text{s.t. } \sum_{k=1}^a q_{ki}h_{kj} &= s_{ij}; i = 1, 2, \dots, g; j = 1, 2, \dots, f & (2) \\ \sum_{i=1}^g l_i q_{ki} &\leq l_{ub}; k = 1, 2, \dots, a & (3) \\ \sum_{j=1}^f h_{kj} &\leq h_{ub}; k = 1, 2, \dots, a & (4) \\ V_k &= \sum_{i=1}^g l_i q_{ki} \times \sum_{j=1}^f h_{kj}; k = 1, 2, \dots, a & (5) \\ UR_k &= \frac{V_k}{l_{ub} \times h_{ub}}; k = 1, 2, \dots, a & (6) \end{aligned}$$

(1) is the objective function that minimizes the number of total used lays. (2) ensures that the production requirement is satisfied exactly. (3) and (4) guarantee that the length and height of each lay do not exceed the maximum allowed length and the maximum allowed height, respectively. (5) computes

(1) the volume of the k th lay. (6) can obtain the volume and utilization rate of the k th lay. (2) The proposed algorithm in this paper is referred to as an HFSC (Heuristic algorithm for Fabric Spreading and Cutting). HFSC includes a constructive procedure to create an initial solution and an improving loop to refine the solution. The main procedure of HFSC is illustrated in Procedure 1. The lay set $BEST_LAY$ denotes the ultimate solution. Firstly an initial solution LAY is obtained by invoking $CreateLays$ (described in Procedure 2). Then each lay in LAY is examined whether it can be transferred to $BEST_LAY$. If the remaining lays in LAY can be rearranged into a less number of new lays after a lay is taken away from LAY , the lay will be transferred to $BEST_LAY$ and LAY is replaced with the new lays. If no such lay exists in LAY , all lays in LAY will be added to $BEST_LAY$. As shown in Procedure 2, Construction () creates and constructs a set of lays in series according to the input parameters. In order to accelerate the overall algorithm, the variables ref_v and ref_h are introduced with $l_{ub} \times h_{ub}$ and h_{ub} as their initial values, respectively. Construction (described in Procedure 3) is invoked repeatedly to create new

lays until all required SKUs are produced. Once a new lay is created, the values of ref_v and ref_h will be replaced with the mean volume and height of all created lays, respectively.

Procedure 1.

```

HFSC( $S, L, l_{ub}, h_{ub}$ )
BEST_Lay :=  $\Phi$ 
LAY := { $lay_1, lay_2, \dots$ } := Construction( $S, L, l_{ub}, h_{ub}$ )
do
  for each lay  $lay_k$  in Lay;
    Let  $S'$  store the numbers of SKUs included in LAY
  except
     $lay_k$ 
    LAY' := { $lay_1, lay'_2, \dots$ } := Construction( $S', L, l_{ub}, h_{ub}$ )
    if ( $|LAY'| + 1 < |LAY|$ )
      Add  $lay_k$  to BEST_LAY
      LAY := LAY'
      break;
    while ( $|BEST\_LAY|$  changes)
  return BEST_LAY

```

Procedure 2.

```

CreateLays( $S, L, l_{ub}, h_{ub}$ )
 $k := 1$ 
 $ref\_v := l_{ub} \times h_{ub}$ 
 $ref\_h := h_{ub}$ 
while (S contains non-zero elements)
   $lay_k :=$ Construction( $S, L, ref\_v, ref\_h, l_{ub}$ )
  Remove SKUs in  $lay_k$  from S
   $ref\_v :=$  the mean volume of  $lay_1, lay_2, \dots, lay_k$ 
   $ref\_h :=$  the mean height of  $lay_1, lay_2, \dots, lay_k$ 
   $k := k + 1$ 
return  $lay_1, lay_2, \dots, lay_k$ 

```

Construction () is illustrated in Procedure 3. Firstly, all possible heights for each fabric type and color are created by invoking CreatePossibleHeights (displayed in Procedure 4). For each composition of heights of different fabric types and colors that add up to ref_h , a set of columns is created by invoking CreateColumns (displayed in Procedure 5). With these columns, a lay is created by solving the one-dimensional knapsack model:

$$KS(l_{ub}, \{(l_1, qc_1), (l_2, qc_2), \dots, (l_n, qc_n)\}) = \max \left\{ \begin{array}{l} \sum_{i=1}^g l_i \times qu_i \\ \left. \begin{array}{l} \{(l_1, qu_1), (l_2, qu_2), \dots, (l_n, qu_n)\} \\ \text{is a feasible solution;} \\ 0 \leq qu_i \leq qc_i, (i = 1, 2, \dots, n); \\ \sum_{i=1}^g l_i \times qu_i \leq l_{ub}; \end{array} \right\} \end{array} \right. \quad (7)$$

For the sake of saving computation time, Construction () will return the best found lay once it find a lay whose volume exceeds ref_v or it cannot find better lays any more.

As illustrated in Procedure 4, CreatePossibleHeights () creates an integer set for each fabric type and color. Each number in the integer set for the j th ($1 \leq j \leq f$) fabric type corresponds to a possible layer number of the j th ($1 \leq j \leq f$) fabric type in the will-be-created lay. The possible layer number of the j th ($1 \leq j \leq f$) fabric type in the lay is among ref_h , 0 and the divisors of the required numbers of the SKUs with the j th ($1 \leq j \leq f$) fabric type. These divisors must be less than ref_h and guarantee the corresponding SKUs could be completely produced in the lay.

Procedure 3.

```

CreateLay( $S, L, ref\_v, ref\_h, l_{ub}$ )
[ $\{ch_1, ch_2, \dots\}, \{ch_2, ch_2, \dots\}, \dots, \{ch_f, ch_f, \dots\}$ ]
:= CreatePossibleHeights( $S, L, l_{ub}, ref\_h$ )
Let best_lay be the best lay so far
while ( the volume of best_lay < ref_v and
   $ref\_h \times l_{ub} >$  the volume of best_lay )
  for each [ $h_1, h_2, \dots, h_g$ ] where  $h_j \in \{ch_{j1}, ch_{j2}, \dots\}$ 
    ( $j = 1, 2, \dots, f$ ) and  $\sum_{j=1}^f h_j = ref\_h$ 
    [ $c_1, c_2, \dots$ ] := CreateColumns ( $s, L, [h_1, h_2, \dots, h_f]$ )
    new_lay := KS ( $l_{ub}, \{c_1, c_2, \dots\}$ )
    if (the volume of new_lay  $\geq ref\_v$ )
      break;
   $ref\_h := ref\_h - 1$ ;
return best_lay

```

Procedure 4.

```

CreatePossibleHeights( $S, L, l_{ub}, ref\_h$ )
for each  $j \in \{1, 2, \dots, f\}$ 
   $CH_j := \{\}$ 
  if ( $\exists s_{ij} \geq ref\_h, i \in \{1, 2, \dots, g\}$ )
    add  $ref\_h$  to  $CH_j$ 
  for each  $ph \in \{ref\_h - 1, ref\_h - 2, \dots, 1\}$ 
    if ( $\exists s_{ij} \neq 0$ , and  $\left[ \frac{s_{ij}}{ph} \right] = \frac{s_{ij}}{ph}$ , and  $\frac{s_{ij}}{ph} \times l_i \leq l_{ub}$ ,
       $i \in \{1, 2, \dots, g\}$ )
      add  $ph$  to  $CH_j$ 
  add () to  $CH_j$ 
return  $\{CH_1, CH_2, \dots, CH_f\}$ 

```

The procedure CreateColumns () creates columns for the SKUs of each garment style and size according to the layer number of each fabric type and color in the will-be-created lay. $[h_1, h_2, \dots, h_f]$ denotes the layer number of each fabric type and color in this lay.

Procedure 5.

```

CreateColumns( $R, L, [h_1, h_2, \dots, h_f]$ )
for each  $i \in \{1, 2, \dots, g\}$ 
   $qc_i := \min \left\{ \frac{s_{ij}}{h_j}, j \in \{1, 2, \dots, f\} \text{ and } h_j \neq 0 \right\}$ 
   $column_i := (l_i, qc_i)$ 
return  $\{column_1, column_2, \dots, column_g\}$ 

```

IV. EXPERIMENTAL EVALUATION

The proposed HFSC algorithm is implemented and compared with SS-HR proposed by [12] in C++, and runs on a server with Intel Core2 Duo Q94002.66 GHz and Microsoft Windows 7 Ultimate. The compiling environment is Microsoft Visual Studio 2012.

We design ten groups (named G1, G2,..., G10) of test cases for evaluating HFSC. Each group contains 50 cases. Each of the 500 cases contains the required numbers of SKUs of 5 garment styles. Each garment style contains 6 garment sizes and 5 fabric types and colors. For each case, l_B , h_B and L are 720, 160 and (60, 63, 66, 69, 73, 76, 69, 72, 75, 78, 82, 86, 80, 83, 86, 90, 94, 98, 90, 94, 98, 102, 106, 110, 99, 103, 107, 111, 115, 120), respectively. The required numbers of SKUs in the 500 cases are generated randomly using the random seed 1000000 in sequence of G1–G10. Table II lists the lower bounds (LBs) and the upper bounds (UBs) of the required numbers of SKUs in the 500 cases, respectively. The data files of the 500 cases are in the attachments.

Table III lists the results comparison between SS-HR and HFSC on the ten groups of 500 test cases. In the experiment, K denotes the ultimate solution of the number of the cutting

bed. Accordingly, UR denotes the utilization rate of the ultimate solution. Besides, TC is the time (a second as a unit) that the computer program spends solving a case. To prevent HFSC from spending too much time on a case, the limit of the running time of HFSC for a case is 1200 s.

Table III shows that HFSC improves the mean K and the mean UR compared with SS-HR on G1–G10, while the mean

TABLE IV
THE DETAILED RESULTS OF HFSC ON G1

Case	K								
1	65	11	63	21	63	31	65	41	64
2	65	12	64	22	64	32	63	42	63
3	65	13	64	23	62	33	64	43	63
4	63	14	64	24	64	34	64	44	65
5	63	15	64	25	65	35	66	45	62
6	65	16	64	26	63	36	63	46	64
7	63	17	62	27	63	37	65	47	64
8	64	18	64	28	64	38	64	48	64
9	65	19	65	29	66	39	64	49	64
10	65	20	64	30	64	40	63	50	63

TABLE II
THE LBS AND THE UBs OF THE REQUIRED NUMBERS OF SKUS IN THE CASES OF G1–G10

	LB	UB
G1	300	400
G2	300	600
G3	400	500
G4	300	800
G5	400	700
G6	500	600
G7	300	1000
G8	400	900
G9	500	800
G10	600	700

TABLE III
THE RESULT COMPARISONS BETWEEN HFSC AND SS-HR ON G1–G10

	Mean K		Mean UR (%)	
	SS-HR	HFSC	SS-HR	HFSC
G1	66.04	63.92	61.20	63.19
G2	75.22	73.14	68.77	70.72
G3	75.16	73.06	69.08	71.06
G4	86.04	83.40	73.69	76.01
G5	85.70	82.78	73.97	76.57
G6	85.00	82.50	74.50	76.75
G7	96.16	93.36	77.53	79.85
G8	96.66	93.74	77.90	80.31
G9	95.98	93.30	78.04	80.27
G10	95.88	93.20	78.09	80.37

TABLE V
THE DETAILED RESULTS OF HFSC ON G2 AND G3

G2				G3			
Case	K	Case	K	Case	K	Case	K
1	72	26	75	1	73	26	73
2	73	27	73	2	73	27	73
3	74	28	73	3	73	28	73
4	72	29	73	4	73	29	73
5	75	30	74	5	74	30	72
6	74	31	74	6	73	31	73
7	73	32	72	7	74	32	72
8	74	33	73	8	72	33	73
9	73	34	73	9	73	34	73
10	73	35	72	10	73	35	74
11	73	36	73	11	73	36	74
12	72	37	74	12	73	37	73
13	73	38	74	13	74	38	73
14	74	39	74	14	73	39	73
15	72	40	73	15	72	40	73
16	72	41	72	16	73	41	73
17	75	42	73	17	74	42	74
18	71	43	72	18	73	43	73
19	72	44	74	19	72	44	73
20	73	45	73	20	73	45	73
21	73	46	73	21	73	46	73
22	75	47	74	22	73	47	73
23	73	48	74	23	73	48	73
24	73	49	74	24	73	49	73
25	73	50	71	25	74	50	73

TABLE VI
THE DETAILED RESULTS OF HFSC ON G4–G6

G4				G5				G6			
Case	K										
1	84	26	83	1	84	26	82	1	82	26	83
2	82	27	83	2	83	27	83	2	83	27	81
3	84	28	85	3	83	28	83	3	83	28	82
4	83	29	85	4	82	29	82	4	84	29	82
5	83	30	82	5	84	30	83	5	82	30	82
6	82	31	84	6	83	31	81	6	83	31	82
7	81	32	82	7	83	32	82	7	83	32	82
8	86	33	82	8	81	33	83	8	83	33	83
9	83	34	82	9	82	34	82	9	82	34	82
10	83	35	84	10	84	35	83	10	83	35	83
11	82	36	83	11	84	36	83	11	83	36	82
12	84	37	85	12	84	37	85	12	82	37	83
13	84	38	85	13	81	38	83	13	83	38	82
14	82	39	85	14	84	39	84	14	83	39	83
15	83	40	84	15	84	40	83	15	83	40	83
16	84	41	82	16	85	41	80	16	83	41	82
17	83	42	84	17	81	42	83	17	81	42	83
18	82	43	83	18	83	43	83	18	84	43	83
19	84	44	82	19	81	44	83	19	82	44	83
20	84	45	85	20	83	45	84	20	81	45	82
21	84	46	82	21	82	46	83	21	82	46	83
22	84	47	84	22	82	47	83	22	82	47	83
23	84	48	83	23	83	48	82	23	83	48	82
24	81	49	86	24	82	49	81	24	83	49	82
25	83	50	86	25	84	50	83	25	82	50	82

cutting bed number K of HFSC is less than that of SS-HR, and the utilization rate of cutting bed of HFSC increases compared with SS-HR. For the cases with the same LB and UB of the required numbers of SKUs, when the difference of the required numbers of SKUs reduces, the mean UR raises slightly. The detailed results of the G1–G10 are listed in Table IV–Table VII, respectively.

Table IV shows that HFSC may obtain a solution ranging from 62 to 66 for each of the 50 cases in G1 in a reasonable time. The difference between the maximum solution and the minimum one is about 6%.

The mean value of SKU numbers of the cases in G2 is approximately same as the one in G3. Table V shows that HFSC may obtain a solution ranging from 71 to 75 for each of the 100 cases in G2 and G3 in a reasonable time. The minimum of the solutions for the 100 cases is about 5% less than the maximum one.

Table VI shows that the cases in G4, G5 and G6 approximately have the same mean SKU number. HFSC may obtain a solution ranging from 81 to 86 for each of the 150 cases in G4, G5 and G6 in a reasonable time. The maximum solution is about 6% greater than the minimum one.

For the most massive 200 cases in G7–G10, HFSC may

obtain a solution ranging from 89 to 96 for each of them in a reasonable time, as shown in Table VII. The difference between the maximum solution and the minimum one is about 7%.

In the experiments, the results show that HFSC is superior to SS-HR through the detailed results mean K and mean UR in the 500 cases. Therefore, the conclusion can be obtained that the proposed method is efficient and has a good performance.

V. CONCLUSION

This paper focuses on the fabric spreading and cutting problem in apparel industries. A heuristic algorithm is brought out to minimize the frequency of using the cutting beds while all required SKUs are produced without producing extra garment components. This algorithm includes a constructive procedure and an improving loop. The constructive procedure creates all lays in sequence according to given requirements. In the algorithm, firstly a lay set is obtained by invoking the constructive procedure, and then the improving loop tries to withdraw each lay from the lay set and invoke the constructive procedure to rearrange the left lays into a smaller lay set. The proposed algorithm is implemented in C++. A computational experiment that includes 500 cases is designed to test the

TABLE VII
THE DETAILED RESULTS OF HFSC ON G7–G10

G7				G8				G9				G9			
Case	K														
1	94	26	93	1	94	26	91	1	95	26	93	1	94	26	93
2	93	27	91	2	93	27	92	2	95	27	95	2	92	27	92
3	95	28	92	3	93	28	93	3	94	28	92	3	93	28	93
4	91	29	96	4	95	29	95	4	92	29	93	4	92	29	94
5	95	30	89	5	93	30	93	5	95	30	95	5	94	30	93
6	92	31	94	6	96	31	96	6	92	31	94	6	94	31	92
7	95	32	94	7	92	32	93	7	92	32	93	7	93	32	93
8	91	33	94	8	96	33	95	8	92	33	91	8	93	33	94
9	95	34	96	9	94	34	93	9	91	34	93	9	93	34	93
10	96	35	92	10	94	35	95	10	94	35	94	10	92	35	93
11	92	36	92	11	93	36	94	11	95	36	94	11	94	36	93
12	93	37	95	12	95	37	94	12	94	37	93	12	93	37	94
13	95	38	96	13	91	38	93	13	94	38	94	13	93	38	94
14	94	39	92	14	94	39	94	14	93	39	92	14	94	39	93
15	93	40	92	15	93	40	93	15	92	40	92	15	93	40	92
16	95	41	92	16	93	41	95	16	94	41	94	16	93	41	93
17	91	42	94	17	93	42	93	17	93	42	93	17	92	42	94
18	96	43	89	18	94	43	92	18	95	43	94	18	94	43	93
19	94	44	96	19	92	44	95	19	93	44	93	19	94	44	93
20	92	45	92	20	92	45	92	20	92	45	93	20	93	45	94
21	91	46	94	21	95	46	96	21	93	46	92	21	93	46	92
22	95	47	93	22	94	47	94	22	93	47	93	22	94	47	94
23	94	48	93	23	93	48	94	23	93	48	91	23	94	48	94
24	95	49	95	24	95	49	96	24	95	49	93	24	94	49	93
25	93	50	92	25	94	50	95	25	94	50	96	25	94	50	92

computer program. For each case, the proposed algorithm gains an effective and efficient result.

REFERENCES

- [1] H. Dyckhoff, "A typology of cutting and packing problems," *European J. Operational Research*, vol. 44, no. 2, pp. 145–159, 1990.
- [2] P. E. Sweeney and E. R. Paternoster, "Cutting and packing problems: a categorized, application-orientated research bibliography," *J. Operational Research Society*, vol. 43, no. 7, pp. 691–706, 1992.
- [3] C. Jacobs-Blecha, J. C. Ammons, A. Schutte, and T. Smith, "Cut order planning for apparel manufacturing," *IIE Trans.*, vol. 30, no. 1, pp. 79–90, 1998.
- [4] D. M. Rose and D. R. Shier, "Cut scheduling in the apparel industry," *Computers & Operations Research*, vol. 34, no. 11, pp. 3209–3228, 2007.
- [5] D. B. Nascimento, J. N. D. Figueiredo, S. F. Mayerle, P. R. Nascimento, and R. M. Casali, "A state-space solution search method for apparel industry spreading and cutting," *Int. J. Production Economics*, vol. 128, no. 1, pp. 379–392, 2010.
- [6] W. K. Wong, "Optimisation of apparel manufacturing resource allocation using a generic optimised table-planning model," *Int. J. Advanced Manufacturing Technology*, vol. 21, no. 12, pp. 935–944, 2003.
- [7] G. Wascher, H. Haubner, and H. Schumann, "An improved typology of cutting and packing problems," *European J. Operational Research*, vol. 183, no. 3, pp. 1109–1130, 2007.
- [8] W. K. Wong, C. K. Chan, and W. H. Ip, "Optimization of spreading and cutting sequencing model in garment manufacturing," *Computers in Industry*, vol. 43, no. 1, pp. 1–10, 2000.
- [9] Z. Degraeve and M. Vandebroek, "A mixed integer programming model for solving a layout problem in the fashion industry," *Management Science*, vol. 44, no. 3, pp. 301–310, 1998.
- [10] Z. Degraeve, W. chet, and R. Jansbc, "Alternative formulations for a layout problem in the fashion industry," *European J. Operational Research*, vol. 143, no. 1, pp. 80–93, 2002.
- [11] A. Toscano, S. Rangel, and H. H. Yanasse, "A heuristic approach to minimize the number of saw cycles in small-scale furniture factories," *Annals of Operations Research*, vol. 258, no. 2, pp. 719–746, 2017.
- [12] M. Vanzela, G. M. Melega, S. Rangel, and S. A. D. Araujo, "The integrated lot sizing and cutting stock problem with saw cycle constraints applied to furniture production," *Computers & Operations Research*, vol. 79, pp. 148–160, 2017.
- [13] D. A. Wuttke and H. S. Heese, "Two-dimensional cutting stock problem with sequence dependent setup times," *European J. Operational Research*, vol. 265, no. 1, pp. 303–315, 2018.
- [14] H. Foerster and G. Wascher, "Pattern reduction in one-dimensional cutting stock problems," *Int. J. Production Research*, vol. 38, pp. 1657–1676, 2000.
- [15] H. H. Yanasse and M. S. Limeira, "A hybrid heuristic to reduce the number of different patterns in cutting stock problems," *Computers & Operations Research*, vol. 33, pp. 2744–2756, 2006.
- [16] Y. Cui, C. Zhong, and Y. Yao, "Pattern-set generation algorithm for the onedimensional cutting stock problem with setup cost," *European J. Operational Research*, vol. 243, no. 2, pp. 540–546, 2015.
- [17] Y. Cui, Y. P. Cui, and L. Yang, "Heuristic for the two-dimensional arbitrary stock-size cutting stock problem," *Computers & Industrial Engineering*, vol. 78, pp. 195–204, 2014.

- [18] N. Ma, Y. Liu, Z. L. Zhou, and C. B. Chu, "Combined cutting stock and lot-sizing problem with pattern setup," *Computers & Operations Research*, vol. 95, pp. 44–55, 2018.
- [19] Y. Cui and Z. Zhao, "Heuristic for the rectangular two-dimensional single stock size cutting stock problem with two-staged patterns," *European J. Operational Research*, vol. 231, no. 2, pp. 288–298, 2013.
- [20] R. Alvarez-Valdés, R. Mart, J. M. Tamarit, and A. Parajin, "GRASP and path relinking for the two-dimensional two-staged cutting stock problem," *INFORMS J. Computing*, vol. 19, no. 2, pp. 261–272, 2007.
- [21] M. Hifi and R. M'Hallah, "An exact algorithm for constrained two dimensional two-staged cutting problems," *Operations Research*, vol. 53, no. 1, pp. 140–150, 2005.
- [22] M. Hifi and R. M'Hallah, "Strip generation algorithms for constrained twodimensional two-staged cutting problems," *European J. Operational Research*, vol. 172, no. 2, pp. 515–527, 2006.
- [23] M. Hifi, R. M'Hallah, and T. Saadi, "Algorithms for the constrained two staged two-dimensional cutting problem," *INFORMS J. Computing*, vol. 20, pp. 212–221, 2008.
- [24] J. Li, X. H. Meng, and X. Dai, "Collision-free scheduling of multi-bridge machining systems: a colored traveling salesman problem-based approach," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 1, pp. 139–147, 2018.
- [25] X. Meng, J. Li, M. C. Zhou, X. Dai, and J. Dou, "Population-based incremental learning algorithm for a serial colored traveling salesman problem," *IEEE Trans. Systems, Man, and Cybernetics: Systems*, vol. 48, no. 2, pp. 277–288, 2018.
- [26] X. W. Guo, S. X. Liu, M. C. Zhou, and G. D. Tian, "Dual-objective program and scatter search for the optimization of disassembly sequences subject to multi-resource constraints," *IEEE Trans. Automation Science and Engineering*, vol. 15, no. 3, pp. 1091–1013, 2018.
- [27] J. Zhao, S. X. Liu, M. C. Zhou, and X. W. Guo, L. Qi, "Modified cuckoo search algorithm to solve economic power dispatch optimization problems," *IEEE/CAA J. Automatic Sinica*, vol. 5, no. 4, pp. 794–806, 2018.
- [28] J. Zhao, S. X. Liu, M. C. Zhou, X. W. Guo, and L. Qi, "An improved binary cuckoo search algorithm for solving unit commitment problem: method description," *IEEE Access*, vol. 6, no. 4, pp. 43535–43545, 2018.
- [29] M. Delorme, M. Iori, and S. Martello, "Bin packing and cutting stock problems: Mathematical models and exact algorithms," *European J. Operational Research*, vol. 255, no. 1, pp. 1–20, 2016.



control.

Xiuqin Shang received the B.Eng. degree in automation from China University of Petroleum, Qingdao, China, in 2005, and the Ph.D. degree in control theory and control engineering from Zhejiang University, China, in 2010. Since 2011, she has been an Assistant Research Scientist in the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, China. Her research interests include social manufacturing, industrial process con-and control.



Dayong Shen received the bachelor degree and master degree in system engineering from NUDT in 2011 and 2013 respectively. He received the Ph.D. degree in management science and engineering from NUDT in 2018. His current research interests include intelligent scheduling, artificial intelligence algorithm, and parallel social systems.



Fei-Yue Wang (S'87–M'89–SM'94–F'03) received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, New York in 1990. He joined the University of Arizona in 1990 and became a Professor and Director of the Robotics and Automation Laboratory (RAL) and Program in Advanced Research for Complex Systems (PARCS). In 1999, he founded the Intelligent Control and Systems Engineering Center at the Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Oversea Chinese Talents Program from the State Planning Council and "100 Talent Program" from CAS, and in 2002, was appointed as the Director of the Key Laboratory of Complex Systems and Intelligence Science, CAS. From 2006 to 2010, he was Vice President for Research, Education, and Academic Exchanges at the Institute of Automation, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory for Management and Control of Complex Systems.

Dr. Wang's current research focuses on methods and applications for parallel systems, social computing, parallel intelligence and knowledge automation. He was the Founding Editor-in-Chief (EiC) of the *International Journal of Intelligent Control and Systems* (1995–2000), Founding EiC of *IEEE ITS Magazine* (2006–2007), EiC of *IEEE Intelligent Systems* (2009–2012), and EiC of *IEEE Transactions on ITS* (2009–2016). Currently he is EiC of *IEEE Transactions on Computational Social Systems*, Founding EiC of *IEEE/CAA Journal of Automatica Sinica*, and *Chinese Journal of Command and Control*. Since 1997, he has served as General or Program Chair of more than 20 IEEE, INFORMS, ACM, and ASME conferences. He was the President of IEEE ITS Society (2005–2007), Chinese Association for Science and Technology (CAST, USA) in 2005, the American Zhu Kezhen Education Foundation (2007–2008), the Vice President of the ACM China Council (2010–2011), and the Vice President and Secretary General of Chinese Association of Automation (CAA, 2008–2018). Since 2019, he has been the President of CAA Supervision Council. Dr. Wang has been elected as Fellow of IEEE, IN-COSE, IFAC, ASME, and AAAS. In 2007, he received the National Prize in Natural Sciences of China and was awarded the Outstanding Scientist by ACM for his research contributions in intelligent control and social computing. He received IEEE ITS Outstanding Application and Research Awards in 2009, 2011 and 2015, and IEEE SMC Norbert Wiener Award in 2014.



Timo R. Nyberg received the M.Sc. (Tech) degree in mechanical engineering from Helsinki University of Technology, Espoo, Finland, in 1988, and the Dr. (Tech) degree in automation from Helsinki University of Technology, Espoo, Finland, in 1993. From 1996 to 2002, he was a Professor of Automation and Director of Graduate School, Tampere University of Technology, Finland. From 2001 to 2002, he was a Senior Vice President, Novo Astra Oy, Finland. From 2002 to 2005, he was Laboratory Director and Management Board member, Helsinki University of Applied Sciences, Finland. Since 2010, he has worked as a Senior Research Fellow and Research Director of School of Science at Aalto University, Finland. Since 2012, he has also been a Visiting Professor of Cloud Computing Center, CAS, China. His research interests include enterprise automation systems, business strategies in software, intelligent wearable device and social manufacturing. He has finished more than 30 research projects funded by Finnish TEKES, Academy of Finland, EU, and Finnish companies. He is the author or co-author of about 100 refereed journal and conference proceeding papers, and 35 EU/American/Finnish patents. He has experience from both the academic world and industries (Metso, Logica, and Nokia). He has over 15 years of experience from R&D cooperation with Chinese and American companies and universities (Tsinghua University, Tongji University, MIT, and Stanford University). He has been the board member in Finland & China Trade Association.