

Optimal Two-Section Layouts for the Two-Dimensional Cutting Problem

Jun Ji*, Dun-hua Huang*, Fei-fei Xing*, and Yao-dong Cui**

Abstract

When generating layout schemes, both the material usage and practicality of the cutting process should be considered. This paper presents a two-section algorithm for generating guillotine-cutting schemes of rectangular blanks. It simplifies the cutting process by allowing only one size of blanks to appear in any rectangular block. The algorithm uses an implicit enumeration and a linear programming optimal cutting scheme to maximize the material usage. The algorithm was tested on some benchmark problems in the literature, and compared with the three types of layout scheme algorithm. The experimental results show that the algorithm is effective both in computation time and in material usage.

Keywords

Cutting Stock Problem, Layout, Optimization, Two-Dimensional Cutting

1. Introduction

The cutting stock problem is common to production practice of many industries, problem. Solutions to the problem seek to make the highest use of material, in order to improve the usage of materials effectively and reduce production costs, which is an effective way to increase the efficiency of enterprises. In manufacturing, many parts are rectangular, so much research has taken place on the optimal layout of rectangular parts.

This paper discusses the two-dimensional guillotine-cutting stock (TDGCS) problem [1–5], cutting a given type and number of blanks from rectangular sheets. It is necessary to determine an optimal layout scheme in order to minimize the total value of the sheet consumed. The TDGCS problem belongs to a typical high-complexity NP problem. It has a significant influence on the industries such as manufacturing, ship-building, automobile industry. The unconstrained two-dimensional guillotine-cutting problem as the UTDGC problem [6–9], which describe cutting a sheet into rectangles of a variety of given sizes and values, where the number of times that each rectangle appears in the sheet is unconstrained, and the layout goal is to maximize the total area of the rectangles contained in the sheet. A solution to the TDGCS problem consists of several layouts, each of which may be constructed by an algorithm for the UTDGC problem. Gilmore and Gomory [10] proposed a linear programming (LP) method to solve the TDGCS problem, which has been widely used. The LP model is solved by iteration

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript received April 10, 2020; first revision October 7, 2020; accepted November 8, 2020.

Corresponding Author: Jun Ji (ji_jun2000@sina.com)

* School of Mechanical, Electronic Engineering, Beijing Polytechnic, Beijing, China (ji_jun2000@sina.com, 1448402711@qq.com, xingfeifei@126.com)

** School of Computer, Electronics and Information, Guangxi University, Nanning, China (08116293@bjtu.edu.cn)

as each iteration needs to generate a new optimal layout.

At present, algorithms have been proposed by many researchers to solve the UTDGC problem. Gilmore and Gomory [10] proposed dynamic programming. Beasley [11] improved Gilmore's method and applied the branch-and-bound method. Herz [12] has developed a tree search method. Hifi and Zissimopoulos [13] and Young-Gun et al. [14] proposed the exact algorithms that could optimally solve the small-scale UTDGC problem. Algorithms above could effectively solve the small-scale UTDGC problem. Because the UTDGC problem is an NP problem, the time spent by the above algorithms in solving such a large-scale problem is unacceptably long. Therefore, a common way to approach the problem is to adopt a specific layout, which can obtain a high material usage in a short time, meet the specific process requirements with simple cutting process and have good adaptability in industrial production. Many researchers have studied specific types of layout. Hifi [7] proposed the classic two-stage and three-stage layout methods. Beasley [11] introduced an exact algorithm to generate general layout methods. Cui [8] proposed a general T-shaped layout, and applied new restrictions in the general two-section layout method [9]. Although the material usage of these layouts may be slightly lower, the calculation time is short and the layout process is simple.

This paper proposes a new restriction on the cutting patterns. It simplifies the cutting process by only allowing blanks of the same size to appear in a block. An algorithm (referred to as the H-2SECTION_A) is proposed to generate the cutting layout scheme for the same-shape blocks to solve the TDGCS problem. In the algorithm, the blank of the same size in each size of the same-shape blocks on the sheet is limited, and the strips to which these blanks belong to are parallel or perpendicular to each other. The usage of sheet metal can be improved by determining the number of the optimal same-shape blocks. In this paper, the test problems in the literature are used to compare the layout algorithm with two types of layouts and a general T-shape layout scheme. The experimental results show that the algorithm in this study is effective in calculating time and sheet usage, and it is able to simplify the cutting process. It should be noted that the H-2SECTION_A algorithm presented could also be seen as an improved version of the algorithm of Cui et al. [9].

The remainder of this paper is organized as follows. Section 2 provides some definitions of the H-2SECTION layout, followed by the process to solve the TDGCS problem in Section 3. An exact algorithm is based on knapsack problems combined with dynamic programming techniques to generate the H-2SECTION layout in Section 4. A set of test problem instances and performance analysis of the proposed algorithms are then presented in Section 5. The results indicate that the H-2SECTION_A was indeed better than the optimal above three solutions. Section 6 concludes the paper. Fig. 1 shows a block diagram of overall the design this paper.

2. Layout and Layout Scheme

2.1 The General Two-Section Layout

The general two-section layout divides the sheet into two sections by a horizontal or vertical cutting line. As shown in Fig. 2, first, the vertical cutting line divides the sheet into left and right sections. Second, the horizontal cutting lines divide each of the section into strips. Finally, the punch divides the strips into blanks, where the number in the box indicates the item number of each size of blanks.

2.2 The Same-Shape Two-Section Layout

A same-shape block, that is, in a sheet, is only composed of blanks of the same size, and the strips that these blanks belong to are parallel or perpendicular to each other. Fig. 3 shows a same-shape block. The number of cuts is numbered in order according to the strip direction. When cutting, the first division cuts two horizontal stripes, the second division cuts five vertical stripes, the third dividing cuts two horizontal stripes, and the fourth dividing cuts two vertical stripes.

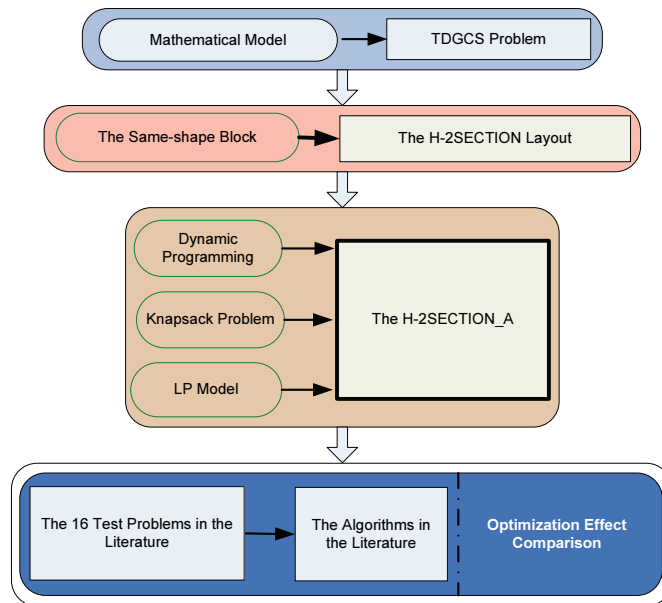


Fig. 1. The Block-diagram of the design idea of this paper.

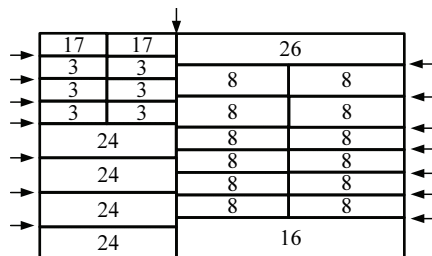


Fig. 2. General two-section layout and its section cutting line.

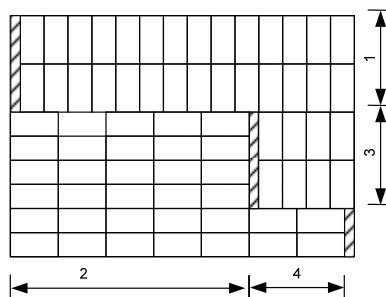


Fig. 3. A same-shape block.

A segment includes multiple same-shape blocks. If it is composed of a series of same-shape blocks arranged horizontally from left to right, it is called an X segment; if it consists of a series of same-shape blocks arranged vertically from top to bottom, it is called a Y segment.

A section consists of multiple segments. If it includes a series of X segments arranged vertically from top to bottom, it is called X section; if it is composed of Y segments arranged horizontally from left to right, it is Y direction section.

In this paper, a same-shape two-section layout (H-2SECTION) is proposed. The H-2SECTION layout means that the sheet is divided into two sections by a horizontal or vertical cutting line. When the cut line is vertical, it is called an HX-2SECTION layout (Fig. 4); when the cut line is horizontal, it is called an HY-2SECTION layout. As shown in Fig. 4, the number represents the cutting line. First, cutting line 1 divides the sheet into two sections. Second, cutting line 2 in each section divides the sections into segments. Finally, cutting lines 3 divides the segments into same-shape blocks.

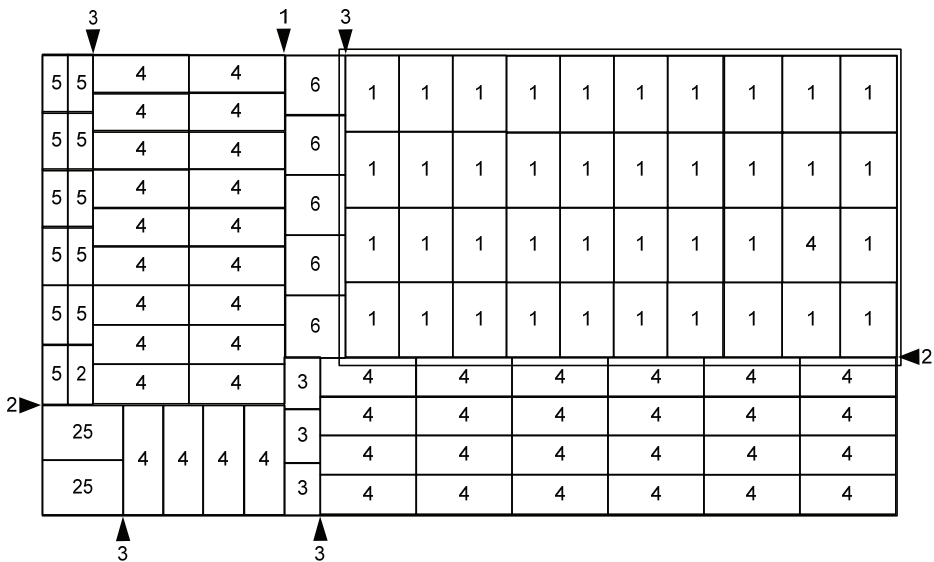


Fig. 4. HX-2SECTION layout and its cutting process.

In Fig. 4, the same-shape block with blank No. 1 in the upper right corner wireframe has just one blank. The H-2SECTION layout is a superset of the general two-section layout. It is made up of strips comprising the same size of blanks. In the stage of separating the strip into blanks, two connected feeding distances are called one step, and all steps are the same within each strip. It can be concluded that the cutting process of H-2SECTION layout is simple.

2.3 Layout Scheme

A layout scheme can have one or more layout modes, which indicates the number of sheets cut according to each layout mode. The “optimal layout scheme” is the layout scheme with the highest cutting usage; a “near optimal layout scheme” is one with the cutting usage close to the highest.

3. The Mathematical Model of TDGCS Problem

3.1 Symbol and Function

Table 1 lists some symbols and functions. Most of them will be re-introduced when they are used for the first time. It may be found that it is more convenient to look for the parameter definition in the table rather than in the text.

Table 1. Symbol and function

Parameter	Definition
L, W	Sheet length and width, $L \geq W$
m	Number of different blank sizes
l_i, w_i, v_i, d_i	Length, width, value and demand respectively of the i th blank, $1 \leq i \leq m$
\mathbf{D}	$\mathbf{D}=[d_1, d_2, \dots, d_m]^T$, d_i indicates the demand of the i th blank
n	Number of cutting patterns considered
\mathbf{X}	$\mathbf{X} = [x_1, x_2, \dots, x_n]^T$, x_j indicates the number of sheets by the j th layout, $1 \leq j \leq n$
\mathbf{A}	Matrix of m rows and n columns with the elements a_{ij} representing the number of the i th blanks in each sheet for by the j th layout, $1 \leq j \leq n$
\mathbf{C}	$\mathbf{C} = [c_1, c_2, \dots, c_n]$, n is the number of layout methods, c_j the value sheet area of the sheet by the j th layout
\mathbf{V}	Vector of the current blank values, $\mathbf{V} = [v_1, v_2, \dots, v_m]$
\mathbf{P}	$\mathbf{P} = [p_1, p_2, \dots, p_m]^T$, p_i indicates the number of the i th blank size included, $1 \leq i \leq m$.
$n^{(i)}(x, y)$	The maximum number of i th blank in the same-shape block $x \times y$
$s(x, y)$	The value of the same-shape block $x \times y$
$f1(x, y)$	The maximum value of X segment $x \times y$
$f2(x, y)$	The maximum value of Y segment $x \times y$
$g1(x, y)$	The maximum value of X section $x \times y$
$g2(x, y)$	The maximum value of Y section $x \times y$
$v1$	The maximum value of the HX-2SECTION layout
$v2$	The maximum value of the HY-2SECTION layout
$v_{\text{H-2SECTION}}$	The maximum value of the H-2SECTION layout

3.2 The Process to Solve the TDGCS Problem

The LP method is reasonable for solving the TDGCS problem if the blank demands are large. Fig. 5 shows the process of solving the problem. The solution includes multiple cutting layouts and indicates the number of sheets to be cut according to each cutting layout. To begin, an initial feasible solution must be constructed. To improve the solution, it is necessary to determine the current value of the blank according to the demand. An approach must be found to generate the current optimal layout P, which yields the maximum total value of blanks among all cutting layouts. If the value of the layout P is greater than previous sheet used, introduce it to form a new solution, and go to the next cycle. Otherwise, the current solution is the optimal one.

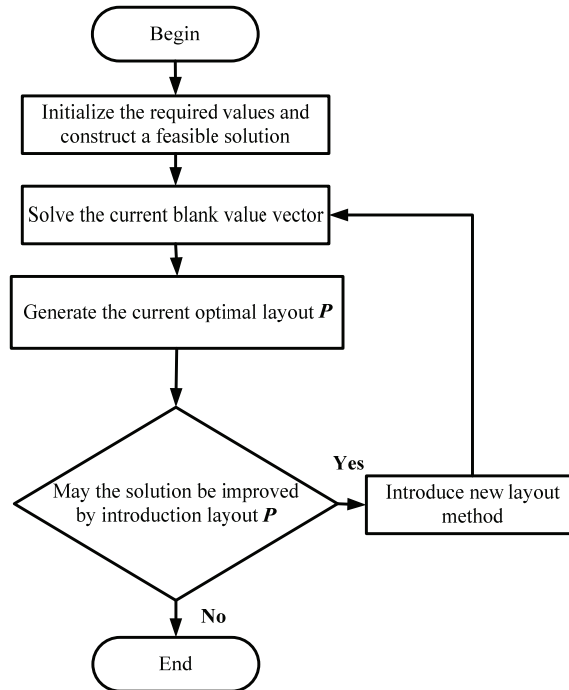


Fig. 5. Process of solving the problem.

3.3 The Mathematical Model of the TDGCS Problem and Its Solution

The exact algorithm for generating the H-2SECTION layout and LP of column generation are combined to generate a layout scheme algorithm to solve the TDGCS problem. The following linear programming model was used.

$$\begin{aligned}
 \min z_1 &= C \cdot X \\
 \text{s. t. } &AX \geq D \\
 &X \geq 0
 \end{aligned} \tag{1}$$

$C = [c_1, c_2, \dots, c_n]$, where n is the number of layout methods, and c_j is the value of the sheet area for the sheet by the j th layout.

$X = [x_1, x_2, \dots, x_n]^T$, where x_j is the number of sheets for the j th layout, $1 \leq j \leq n$.

A is a two-dimensional matrix whose elements a_{ij} represent the number of blanks in each sheet for the j th layout, $1 \leq i \leq m$.

$D = [d_1, d_2, \dots, d_m]^T$, where d_i indicates the demand of the i th blank.

Using column generation to solve the above model, in each iteration cycle, it is necessary to call the generation algorithm of each of the above layout methods according to the current value of the blank to generate a layout scheme.

Model (1) is an LP model, which can be solved according to the following steps [15].

Step 1: Determine the initial feasible solution for model (1). Let A be the unit matrix, and the initial feasible solution is $X = D$. The i th method includes only one blank of the i th blank, $1 \leq i \leq m$.

Step 2: Determine the current blank value vector, $V = CA^{-1} = [v_1, v_2, \dots, v_m]$, where v_i indicates the value of the i th blank.

Step 3: Find a layout that may improve the objective function. Suppose the layout mode under investigation is $P = [p_1, p_2, \dots, p_m]^T$, where p_i is the number of the i th blank size included, $1 \leq i \leq m$. According to the theory of LP, if $c - V \cdot P < 0$, it is possible to improve the goal value by introducing layout P . Use a column P in the replacement matrix A , and then go to the second step. Otherwise, there is no layout that can improve the objective function of the model (1). The current solution is the optimal solution. P can be solved using model (2).

$$z_2 = \max \left(\sum_{i=1}^m v_i p_i \right) \tag{2}$$

s. t. P is a feasible layout

4. The Generation of H-2SECTION Layouts

This section describes the algorithm for generating the optimal layout H-2SECTION layout, where i is the number of the type of blank, $1 \leq i \leq m$, l_i , w_i and v_i are the length, width and value of the i th blank size, respectively; $1 \leq i \leq m$; L and W are the length and width of the sheet, $0 \leq x, y \leq L$, x and y are the length and width of the same-shape block, segments and H-2SECTION layout, respectively.

Step 1: Determine the value $s(x, y)$ of the same-shape block $x \times y$

Denote $n^{(i)}(x, y)$ as the maximum number of the i th blank in the same-shape block of dimensions $x \times y$.

$$n^{(i)}(x, y) = \max \left[n^{(i)}(x, y - l_i) + \text{int} \left(\frac{x}{w_i} \right), n^{(i)}(x - l_i, y) + \text{int} \left(\frac{y}{w_i} \right) \right] \tag{3}$$

$$s(x, y) = \max [n^{(i)}(x, y) \times v_i]$$

Step 2: Determine the optimal same-shape blocks layout of segments

$f1(x, y)$ is referred to as the maximum value of X segment of dimensions $x \times y$, and $f2(x, y)$ is referred to as the maximum value of Y segment of dimensions $x \times y$, where k_i is the number of the same-shape blocks.

$$f1(x, y) = \left\{ \max \left[\sum_{i=1}^M k_i s(x_i, y) \right]; \sum_{i=1}^M k_i x_i \leq x; k_i \in N \right\} \tag{4}$$

$$f2(x, y) = \left\{ \max \left[\sum_{i=1}^N k_i s(x, y_i) \right]; \sum_{i=1}^N k_i y_i \leq y; k_i \in N \right\}$$

Step 3: Determine the optimal segments layout of sections

$g1(x, y)$ is referred to as the maximum value of X section of dimensions $x \times y$, and $g2(x, y)$ is referred to as the maximum value of Y section of dimensions $x \times y$, where k_i is the number of the segments.

$$g1(x, y) = \left\{ \max \left[\sum_{i=1}^N k_i f1(x, y_i) \right]; \sum_{i=1}^N k_i y_i \leq y; k_i \in N \right\} \tag{5}$$

$$g2(x, y) = \left\{ \max \left[\sum_{i=1}^M k_i f2(x_i, y) \right]; \sum_{i=1}^M k_i x_i \leq x; k_i \in N \right\}$$

Step 4: Determine the optimal H-2SECTION layout

When the sheet is divided into two sections by a vertical cutting line, $v1$ refers to the value of the HX-2SECTION layout. As shown in Fig. 6(a), the left and right sections can be either an X section or a Y section. There are three cases. When both sections are X sections, $v1_{XX}$ is the layout value. When one is an X section and the other is a Y section, $v1_{XY}$ is the layout value. When both sections are Y sections, $v1_{YY}$ is the layout value.

$$\begin{aligned}
 v1_{XX} &= \max[g1(x, W) + g1(L - x, W)] \\
 v1_{XY} &= \max[g1(x, W) + g2(L - x, W)] \\
 v1_{YY} &= \max[g2(x, W) + g2(L - x, W)] \\
 v1 &= \max(v1_{XX}, v1_{XY}, v1_{YY})
 \end{aligned}
 \tag{6}$$

When the sheet is divided into two sections by a horizontal cutting line, $v2$ is the value of the HY-2SECTION layout. As shown in Fig. 6(b), the upper and lower sections can be either X-or Y-sections. There are three cases. When both sections are X sections, $v2_{XX}$ is the layout value. When one is an X section and the other is a Y section, $v2_{XY}$ is the layout value. When both sections are Y sections, $v2_{YY}$ is the layout value.

$$\begin{aligned}
 v2_{XX} &= \max[g1(L, y) + g1(L, W - y)] \\
 v2_{XY} &= \max[g1(L, y) + g2(L, W - y)] \\
 v2_{YY} &= \max[g2(L, y) + g2(L, W - y)] \\
 v2 &= \max(v2_{XX}, v2_{XY}, v2_{YY})
 \end{aligned}
 \tag{7}$$

Let $v_{H-2SECTION}$ be the maximum value of H-2SECTION layout.

$$v_{H-2SECTION} = \max(v1, v2)
 \tag{8}$$

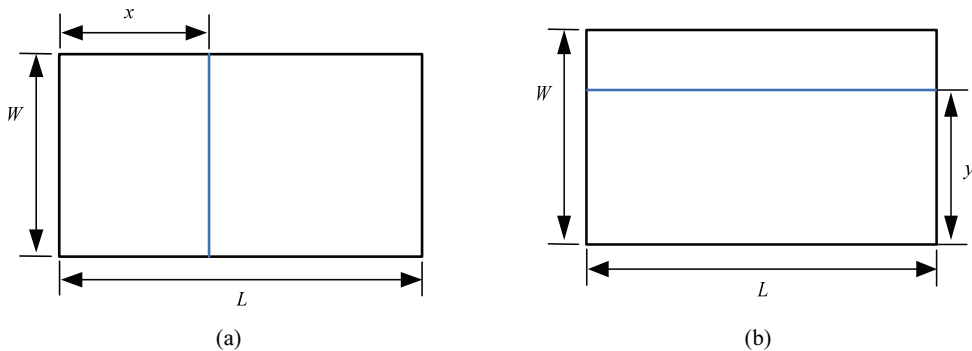


Fig. 6. The cutting line divides the sheet into two sections: (a) vertical cutting line and (b) horizontal cutting line.

5. The Computational Results

All experiments were carried out on a computer with a processor speed of 2.7 GHz of and 8 GB of RAM. Through the test problems in the literature, the algorithm in this paper was compared with two types of two-section layout algorithms and general T-shape layout scheme respectively. The following

notations are used to denote the algorithms:

H-2SECTION_A	The exact algorithm for the problem of H-2SECTION layout
BOPT	Bealy's exact algorithm [11] for the of general (non-staged) guillotine cutting problem
TA	Cui's exact algorithm [8] for the of T-shape layout problem
TSEC	Cui's exact algorithm [9] for the of two-section layout problem

5.1 The First Group Problems

Using the six test questions [9, Table 4], the sheet size (cm × cm) is 3000 × 1500. The blank size is evenly distributed in the range [200,700], and the blank value is equal to its area. For each test problem, this study generates an algorithm for the optimal two-section layout of the same-shape block. In [11], the general layout is generated by an exact algorithm, and a two-section layout was generated by the algorithm in [9]. where $v1$, $v2$, and $v3$ denote the values of the H-2SECTION_A, BOP and the TSEC, respectively, and $t1$ denotes the computation time of the H-2SECTION_A. Table 2 shows the experimental results.

Table 2. Calculation results of six test problems

No.	$v1$	$v2$	$v3$	$v2/v1$ (%)	$v3/v1$ (%)	$t1$
P1	4492074	4490544	4480372	99.97	99.74	0.40
P2	4493050	4488944	4484676	99.91	99.81	0.34
P3	4497594	4489836	4489341	99.83	99.82	0.37
P4	4492558	4487967	4483009	99.90	99.79	0.36
P5	4493361	4485616	4482627	99.83	99.76	0.35
P6	4496824	4494340	4491008	99.94	99.87	0.35

Among the six test problems, the optimization results of the algorithm in this study are better than those of the two types of layout algorithms. Both the H-2SECTION_A and TSEC can will generate optimal two-section patterns. The H-2SECTION_A may be seen as an improved version of the TSEC. The algorithm in this paper took 2.17 seconds to solve the six test problems in total, with average calculation time of each test problem being 0.36 seconds, but TSEC took 0.01 seconds on a computer with processor speed of 1.9. Therefore, in future work, it will be necessary to simplify the structure of the same-shape block to further shorten the calculation time.

For the six test problems, the calculation time variance of H-2SECTION_A was 0.021. From the calculation time variance, the reliability of the H-2SECTION_A in this study is good.

Fig. 7 is a layout diagram of the six test problems generated by the algorithm of the optimal layout method in this study.

5.2 The Second Group Problems

Using the 10 test questions [8, Table 7], the H-2SECTION_A layout was compared with the TA. In [8], the sheet size was evenly distributed in the range [1500,3000], the blank type in the range [2,16], the blank size in the range [150,600], and the blank demand in the range [1000,20000].

Among the 10 test problems, the material usage of the two algorithms was compared, and the fourth problem had the largest improvement. Fig. 8 shows a layout plan diagram for the fourth problem.

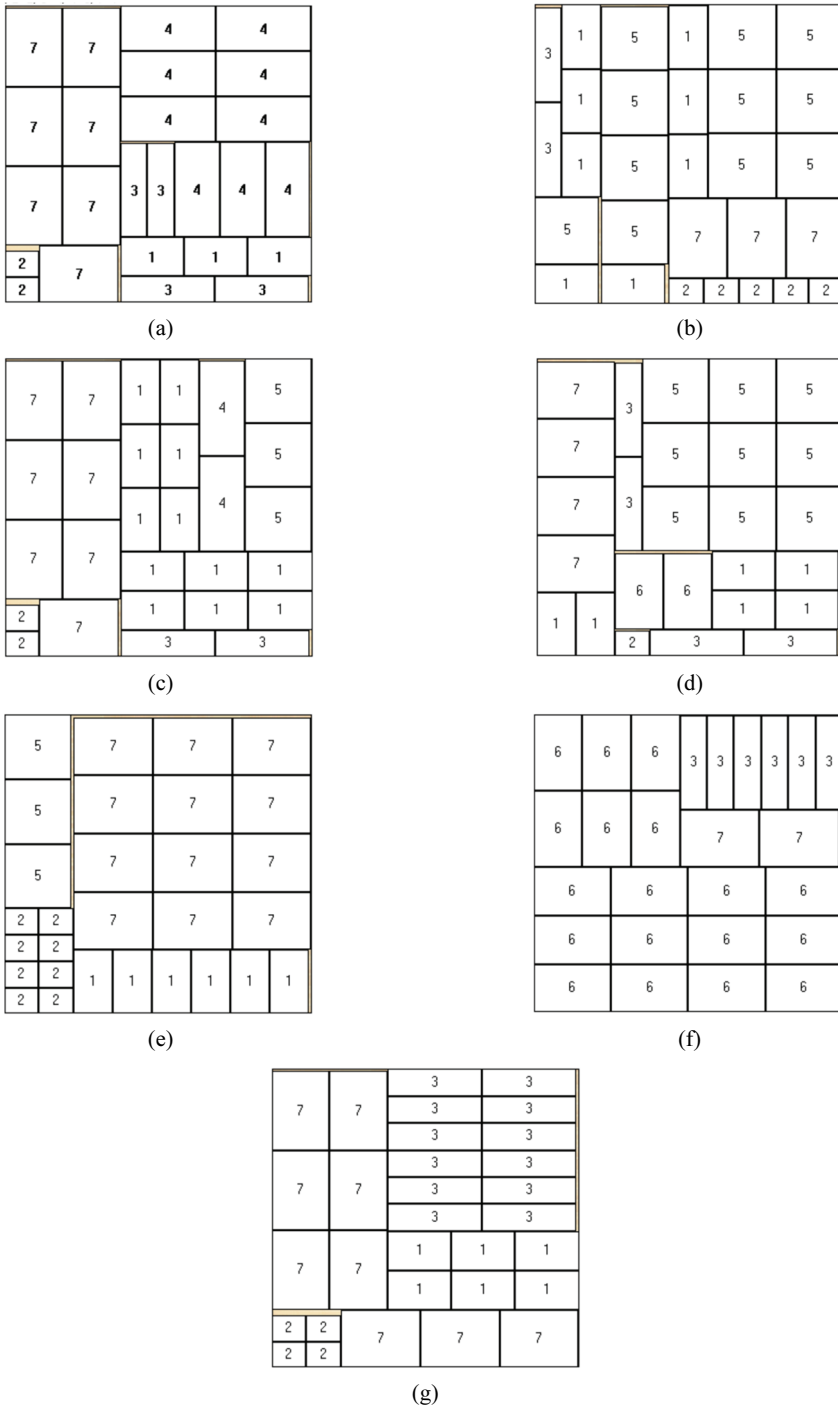


Fig. 8. The layout scheme for the sixth problem (3374 pieces; usage of 98.86%): (a) 963 pieces, (b) 1324 pieces, (c) 315 pieces, (d) 101 pieces, (e) 14 pieces, (f) 426 pieces, and (g) 231 pieces.

6. Conclusion

The H-2SECTION_A can generate optimal two-section patterns for solving TDGCS problem, can improve the material usage, and have a good adaptability at the same time, which is especially suitable for the case of cutting and stamping processes. In this study, the test problems previously described in the literature were used. Compared with two types of layout algorithms and the general T-shape layout scheme algorithm, The H-2SECTION_A improved the material usage with a reasonable time, so it is worth recommending. There are two avenues of investigation for follow-up work. We will further simplify the structure of the same-shape block to improve the calculation speed. In addition, when applying the LP method combined with the layout algorithm to solve large-scale problems, the utilization rate of the plates can be further improved by increasing the number of plates so that better layout schemes can be selected on more plates.

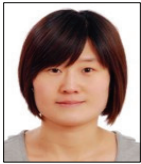
Acknowledgement

This research was supported by the National Natural Science Foundation of China (No. 71371058, 61363026), Education Commission Scientific Foundation of Beijing (No. KM201910858004) and Natural Science Foundation of Beijing Polytechnic (No. 2019Z002-003-KXB).

References

- [1] C. H. Cheng, B. R. Feiring, and T. C. E. Cheng, "The cutting stock problem—a survey," *International Journal of Production Economics*, vol. 36, no. 3, pp. 291-305, 1994.
- [2] P. E. Sweeney and E. R. Paternoster, "Cutting and packing problems: a categorized, application-orientated research bibliography," *Journal of the Operational Research Society*, vol. 43, no. 7, pp. 691-706, 1992.
- [3] R. Alvarez-Valdes, F. Parreno, and J. M. Tamarit, "A tabu search algorithm for a two-dimensional non-guillotine cutting problem," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1167-1182, 2007.
- [4] R. Baldacci and M. A. Boschetti, "A cutting-plane approach for the two-dimensional orthogonal non-guillotine cutting problem," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1136-1149, 2007.
- [5] P. C. Gilmore and R. E. Gomory, "Multistage cutting stock problems of two and more dimensions," *Operations Research*, vol. 13, no. 1, pp. 94-120, 1965.
- [6] J. Ji, Y. P. Lu, J. Z. Cha, J. Q. Yu, and Y. D. Cui, "An exact algorithm for large-scale unconstrained three staged cutting problems with same-size block requirement," *International Journal of Information and Management Sciences*, vol. 23, no. 1, pp. 59-78, 2012.
- [7] M. Hifi, "Exact algorithms for large-scale unconstrained two and three staged cutting problems," *Computational Optimization and Applications*, vol. 18, no. 1, pp. 63-88, 2001.
- [8] Y. Cui, "Generating optimal T-shape cutting patterns for rectangular blanks," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 218, no. 8, pp. 857-866, 2001
- [9] Y. Cui, D. He, and X. Song, "Generating optimal two-section cutting patterns for rectangular blanks," *Computers & Operations Research*, vol. 33, no. 6, pp. 1505-1520, 2006.

- [10] P. C. Gilmore and R. E. Gomory, "A linear programming approach to the cutting stock problem—Part II," *Operations Research*, vol. 11, no. 6, pp. 863-888, 1963.
- [11] J. E. Beasley, "Algorithms for unconstrained two-dimensional guillotine cutting," *Journal of the Operational Research Society*, vol. 36, no. 4, pp. 297-306, 1985.
- [12] J. C. Herz, "Recursive computational procedure for two-dimensional stock cutting," *IBM Journal of Research and Development*, vol. 16, no. 5, pp. 462-469, 1972.
- [13] M. Hifi and V. Zissimopoulos, "A recursive exact algorithm for weighted two-dimensional cutting," *European Journal of Operational Research*, vol. 91, no. 3, pp. 553-564, 1996.
- [14] G. Young-Gun, Y. J. Seong, and M. K. Kang, "A best-first branch and bound algorithm for unconstrained two-dimensional cutting problems," *Operations Research Letters*, vol. 31, no. 4, pp. 301-307, 2003.
- [15] J. V. De Carvalho, "LP models for bin packing and cutting stock problems," *European Journal of Operational Research*, vol. 141, no. 2, pp. 253-273, 2002.



Jun Ji <https://orcid.org/0000-0002-8272-9908>

She received B.S. degree in computer science from University of Jinan in 2003, M.S. degree in computer science from Guangxi Normal University in 2006, and Ph.D. degree in School of Mechanical Electronic and Control engineering from Beijing Jiaotong University in 2012. She is currently an associate professor in the School of Mechanical, Electronic Engineering, Beijing Polytechnic. Her research interests include optimization and computation techniques, computer added design.



Dun-hua Huang <https://orcid.org/0000-0002-5537-2149>

He received B.S. degree in mechanical engineering and automation from Beijing Technology and Business University in 2002, M.S. degree in control engineering from Beihang University in 2009. He is with philosophy in organization development from Assumption University as a Ph.D. candidate. He is currently a professor in the School of Mechanical, Electronic Engineering, Beijing Polytechnic. His research interests include intelligent control.



Fei-fei Xing <https://orcid.org/0000-0002-3396-1823>

She received B.S. degree in applied mathematics from University of Jinan in 2004, M.S. degree in applied mathematics from Shandong Normal University in 2007, and Ph.D. degree in School of Mechanical Electronic and Control engineering from Beijing Jiaotong University in 2013. She is currently a lecturer in the School of Mechanical, Electronic Engineering, Beijing Polytechnic. Her research interests include optimization and computation techniques, computer added design.



Yao-dong Cui <https://orcid.org/0000-0002-3472-1442>

He received B.S., M.S., and Ph.D. degree in computer science from Nanjing University of Aeronautics and Astronautics in 1982, 1985, and 2002, respectively. He is currently a professor in School of Computer, Electronics and Information, Guangxi University. His research interests include optimization and computation techniques, computer added design.