Efficient discretization set for solving rectangular plate cutting problem

Xie Tianlun[1], **Hu Dongping**[1], **Yin Aihua**[1], **Wang Lei**[2], **Wang Jing**[1]
[1]Jiangxi University of Finance and Economics, China
[2]Wuhan University of Science and Technology, China
aihuayin@jxufe.edu.cn

Abstract

In order to improve the efficiency of plate cutting process and the utilization of plate in industry, the cutting problem of rectangular plate with defects is studied, and an efficient recursive algorithm is proposed. The algorithm adopts the idea of dynamic programming. The key is to define a new discretization set structure and generalize the conclusion of Herz theorem, so as to ensure the optimality of the algorithm in this paper. It also defines a new lower bound of pure plate by using the characteristic of transmitting the result from the sub problem to its parent problem, which can significantly reduce the amount of calculation of homogeneous cutting. Experiments show that the computational efficiency of this algorithm is dozens of times faster than the current algorithm.

1 Introduction

Rectangular layout problem is a typical NP-hard problem [Kellerer *et al*., 2004; Na *et al*., 2014], which has been widely studied by industry scholars [Iori *et al*., 2007; Wu *et* al., 2019]. For example, in the manufacturing industry, large plates are usually cut into small items by cutting process to meet the needs of users, and these raw materials such as wood, leather and steel often contain defective areas. In some assembly lines, the location and size of defects are monitored by cameras in real time, and the cutting decision must be made in a very short time (a few seconds) [Aksu and Durak, 2016]. Therefore, in the face of large-scale cutting requirements, it is of great significance and practical value to design efficient cutting methods.

Most of the previous studies on cutting problems only consider a single defect. Carnieri *et al*. [1993] proposed a dynamic programming heuristic algorithm based on Gilmore and Gomory [1996], which analyzes the influence of defect location on the optimal cutting pattern. Neidlein *et al*. [2008] used the discretized cutting set proposed by Herz [1972], combined with depth first and mountain climbing strategy, which improved the efficiency of the algorithm, but reduced the quality of the solution to a certain extent.

Nowadays, more and more literatures present more effective algorithms to solve this problem. Afsharian *et al*. [2014] gave a heuristic dynamic programming method DPD. In their research, multiple defects are allowed on the original plate, and the quantity of items is not limited. They added the size of defects to the composition of cutting discrete set, and symmetrically cut the defective plate in horizontal and vertical directions respectively. The experimental results show that this method obtains high quality of the solution. Wu *et al*. [2019] designed two priority heuristic algorithms by adopting two strategies: defect collision processing and early defect removal. Martin *et al*. [2020] proposed an integer linear programming model based on rectangular block discretization, and designed Benders Decomposition algorithm B&BC and an efficient constrained programming algorithm B&C-CP, which provide a new solution idea for different types of examples. Yin *et al*. [2019] additionally considered the cutting position of one unit from the four sides of the defect, combined with the cutting discrete set of cargo size combination, proposed an improved heuristic dynamic programming method IHDP, and obtained the optimal solution of most typical instances. Gonçalves and Wäscher [2020] designed a hybrid algorithm to solve the problem of non-guillotine pattern. The algorithm is based on a new minimum mean square error model and a partial random key genetic algorithm (BRKGA). The solution is better than that of guillotine mode.

In this paper, the rectangular cutting problem with defects in guillotine pattern is studied, a new subproblem partition strategy is proposed, and a dynamic programming algorithm with efficient discretization set is obtained EDDP (Efficient Discretization set Dynamic Programming). According to the correlation between defects and normalized cutting mode, EDDP intercepts the non-negative combination number of width and height dimensions of the items respectively with the left (lower) and right (upper) of the plate as the baselines, and constructs a new discretization set. With this discretization set, EDDP can obtain the normalized cutting pattern of the defective plate and ensure the optimality of the solution obtained by the algorithm (Theorem 1). This paper also defines a new lower bound of the subproblem without defects, takes only a small part of this kind of subproblem whose lower bound is greater than 0, and constructs a small two-dimensional table to store them, which improves the efficiency of the algorithm.

2 Problem description

The two-dimensional rectangular cutting problem with defects discussed in this paper is described as follows [Morabito and Pureza, 2010]:

Given a rectangular plate, the width of its size is W_0, the height is H_0, and the original plate contains several defective blocks. The given rectangular items include m types. The width and height of these items are w_i, $h_i (i = 1, 2, \dots, m)$, and their values are expressed by their area as $v_i = w_i \times h_i$. The problem is to cut items that do not have defects from the original plate, and to maximize the sum of the areas of these items.

Each solution of the problem corresponds to a type of cutting pattern of the original plate and each pattern is required to meet the following constraints: each cutting is a guillotine, the direction of items is fixed, both the quantity of items and the cutting stage are unlimited.

Let $\eta = (\eta_1, \eta_2, \dots, \eta_m)$ represent the items information in a cutting pattern, η_i represents the number of the i^{th} item obtained in this pattern, then the objective function of the problem can be described as follows.

$$f = \begin{cases} \max v = \sum_{i=1}^{m} v_i \cdot \eta_i \\ s.t.(\eta_1, \eta_2, \dots, \eta_k) \text{ is a feasible cutting pattern.} \end{cases} \quad (1)$$

A Cartesian coordinate system is established according to the above description. The width direction of the plate is X-axis and the height direction of the plate is Y-axis. The original plate is represented by $R = (0, 0, W_0, H_0)$, as shown in Figure 1. Since each cut is a guillotine, this paper uses one or more rectangles to represent irregular defects. The defects on the original plate can be expressed as n quads $d_j = (x_j^d, y_j^d, w_i^d, h_i^d)$。

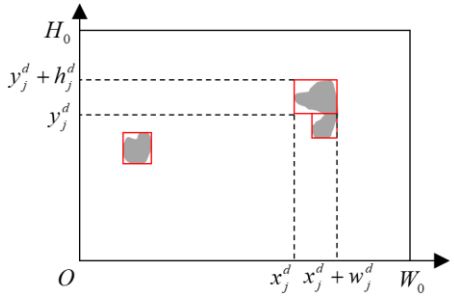

Figure 1: Problem diagram

In summary, the problem discussed in this paper is a two-dimensional cutting problem of single rectangular plate with defects in guillotine pattern, with unlimited quantity of items and number of cutting stages.

3 Algorithm description

This section will describe EDDP algorithm from three aspects: subproblem partition strategy (discretization set), subproblem lower bound and its recursive process.

3.1 Basic concepts

Here, we first introduce some basic concepts.

Intermediate plate After each cutting, one plate becomes two small plates. In the process of solving the problem, many small rectangular plates will be cut and generated. These plates are called intermediate plates. For an intermediate plate with width w and height h, the coordinates of the lower left corner are (x_0, y_0). If it has defects, it is represented by quadruple $R = (x_0, y_0, w, h)$, otherwise it is represented by $R = (w, h)$.

C-plate and D-plate The intermediate plate without defects is called C (clean)-plate, and the corresponding subproblem is C-subproblem; The intermediate plate with defects is called D (defective)-plate, and the corresponding subproblem is D-subproblem.

An intermediate plate $R=(x_0, y_0, w, h)$ contains defect $d_j = (x_j^d, y_j^d, w_j^d, h_j^d)$ or not which can be calculated by formula (2).

$$\lambda_j = \begin{cases} 0, if\ (x_j^d + w_j^d \le x_0\ or\ x_j^d \ge x_0 + w) or \\ \quad (y_j^d + h_j^d \le y_0\ or\ y_j^d \ge y_0 + h)\ for\ j = 1, \dots, n; \\ 1, otherwise. \end{cases} \quad (2)$$

Here, $\lambda_j = 1$ means d_j in the subplate R, i.e. R is a D-plate; $\lambda_j = 0$ means that d_j does not intersect with R. The subplate R not intersecting with any defect is the C-plate.

3.2 Cut point discretization set

In order to solve the two-dimensional rectangular cutting problem without defects, researchers have proved that any feasible cutting pattern can be transformed into an equivalent normalized pattern, in which the items are cut from the lower left corner of the plate [Herz, 1972; Christofides and Whitlock, 1977]. Considering the nonnegative integer combination of the width and height of the items, they constructed the vertical and horizontal classical discretization sets to obtain the normalized pattern, as shown below.

$$T_x^r(w) = \left\{ c_w \middle|\ c_w = \sum_{i=1}^{m} \alpha_i w_i, 1 \le c_w < w, \alpha_i \in Z_+ \cup \{0\} \right\} \quad (3)$$

$$T_y^r(h) = \left\{ c_h \middle|\ c_h = \sum_{i=1}^{m} \beta_i h_i, 1 \le c_h < h, \beta_i \in Z_+ \cup \{0\} \right\} \quad (4)$$

The cutting point of the intermediate plate $R = (x_0, y_0, w, h)$ is obtained from its discretization set elements, and the distance from a cutting point to the left (lower) edge of R is equal to the corresponding element value.

Considering that the items may be at the top right of some defects and the normalized pattern requires the items to be close to the bottom left of the plate, this paper defines a new discretization set, as shown below.

$$T_x^{dr}(w) = \left\{ \begin{array}{l} c_w \mid c_w = \lambda_j \left(x_j^d + w_j^d - x_0 + \sum_{i=1}^{m} \alpha_i w_i \right), \\ 1 \le c_w \le w - 1,\ j \in \{1, 2, \dots, n\}, \alpha_i \in Z_+ \cup \{0\} \end{array} \right\} \quad (5)$$

$$T_y^{dr}(h) = \left\{ \begin{array}{l} c_h \mid c_h = \lambda_j \left(y_j^d + h_j^d - y_0 + \sum_{i=1}^{m} \beta_i h_i \right), \\ 1 \le c_h \le h - 1,\ j \in \{1, 2 \dots, n\}, \beta_i \in Z_+ \cup \{0\} \end{array} \right\} \quad (6)$$

Equation (5) indicates that a step baseline is established on the right side of each defect of the intermediate plate in the horizontal direction, and the vertical discretization set $T_x^{dr}(w)$ is constructed. Equation (6) represents the corresponding horizontal discretization set $T_y^{dr}(h)$. For any intermediate plate $R = (x_0, y_0, w, h)$, the new discretization set in this paper is obtained by combining equations (3-4) and equations (5-6), as shown in the equations (7-8):

$$T_x(w) = T_x^r(w) \cup T_x^{dr}(w) \tag{7}$$

$$T_y(h) = T_y^r(h) \cup T_y^{dr}(h) \tag{8}$$

Compared with the complete discretization set, the discretization sets $T_x(w)$ and $T_y(h)$will not reduce the quality of the solution obtained by our recursive algorithm in this paper, and guarantee it obtain the optimal solution of the problem. This conclusion is described in Theorem 1 as follows.

Theorem 1: *Any cutting pattern of defective plate can be equivalent converted to another cutting pattern. The converted pattern can make the cutting points fall into the discretization sets $T_x(w)$ and $T_y(h)$, and the objective function value obtained by the former does not decrease*.

Proof. Call an arbitrary cutting pattern of defective plate as a general cutting pattern. In this pattern, the left (lower) side of the cutting line(point) on the left (lower) side of the items may not fit the plate, defect or other items, so the cutting line(point) and the items on the right (upper) side can move to the left (lower). After this translation, the left (lower) side of the cutting line where the left (lower) side of the items is located must fit the plate, defects or other items. The items should be at the lower left of the plate as far as possible, and the waste should be concentrated at the upper right, as shown in Figure 2.

After translation, the positions of the cutting lines on the left and right sides of all items must exist in the discretization set $T_x(w)$, three cases are discussed here. For any vertical cutting line, First, if the left side of the cutting line fits the plate, its position is 0, and the right cutting line of the right items is in $T_x^r(w)$. Then, if the left side of the cutting line is defective, the position of the cutting line is in $T_x^{dr}(w)$. Finally, if the left side of the cutting line fits the items, the cutting line belongs to the same discretization set as the left cutting line of the left items.

The proof of the horizontal cutting line in the vertical direction is the same, which will not be repeated here. Theorem is proved. □

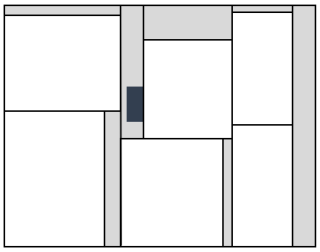

Figure 2: Normalized pattern of the defective plate

Theorem 1 extends the conclusion of Herz [1972] to the problem with defects. This generalization is an important breakthrough for us to solve the cutting problem with defects. The most important thing is that it provides a theoretical guarantee for the algorithm in this paper to obtain the optimal solution. This paper also refers to the above cutting pattern after translation as the normalized pattern of the defective plate.

3.3 Lower bound of C-subproblem

At present, a lower bound based on homogeneous cutting is widely used in recursive algorithms for solving problems without defects. The lower bound can not only deal with the problem of solution decline caused by symmetry cutting [Beasley, 1985], but also be used in various heuristic strategies. This lower bound is defined as follows.

$$g(w,h) = \max\left\{\left(0, \left\lfloor \frac{w}{w_i} \right\rfloor \times \left\lfloor \frac{h}{h_i} \right\rfloor\right) \middle| \begin{matrix} w_i \le w, h_i \le h, \\ i = 1, \ldots, m \end{matrix}\right\} \tag{9}$$

According to the analysis, while solving a large-scale C-problem, there are many repeated calculations between g (w, h) and recursion. This is because a C-problem is divided into multiple C-subproblems with the same solution in recursion, and the sum of the solutions of these C-subproblems is equal to $g(w, h)$. In this paper, a new lower bound $C(w, h)$ is defined to greatly reduce the amount of repeated calculation and improve the efficiency of the algorithm.

$$C(w,h) = \max\left\{0, v_i \middle| \begin{matrix} w_i \le w < 2w_i, h_i \le h < 2h_i, \\ i = 1, \ldots, m \end{matrix}\right\} \tag{10}$$

For a C-plate, $R = (w, h)$, if an item i meets $R = (w, h)$ and $h_i \le h < 2h_i$, then the value of $C(w, h)$ is the maximum area of the item meeting these conditions. Otherwise, none of the items meets these conditions, $C(w, h) = 0$.

Let $w_{max}=\max(w_i|1 \le i \le m)$, $h_{max}=\max(h_i|1 \le i \le m)$, according to equation (10), the width and the height of C-plate with a new lower bound greater than 0 is less than $2w_{max}$ and $2hmax$, respectively. Therefore, a $2w_{max} \times 2h_{max}$ matrix (two-dimensional table), called a K-table, can be constructed to store the value of the new lower bound. It is not difficult to know that m types of items need to be traversed when calculating the value of each element of the K-table. Therefore, the time complexity of calculating K-table is $O(m \cdot w_{max} \cdot h_{max})$.

Not only the time complexity of calculating the K-table is very small, but also the K-table can be calculated before the beginning of recursion, so as to facilitate the subsequent calculation of the lower bound by looking up the table. However, the calculation range of the lower bound $g(w, h)$ includes all C-subproblems, and its calculation amount is much larger. This will lead to great advantages in calculation efficiency when the size of items differs greatly from that of the original plate.

3.4 The new dynamic programming

In the recursive process, all subproblems are divided by the new discretization set elements, and the solution of the current problem is the maximum of its new lower bound $C(w, h)$

and the sum of its subproblems' solutions. The new recursive function for solving the C-subproblem is defined as follows.

$$F(w,h)=\max\left\{\begin{array}{c} F(c_w,h)+F\left(p(w-c_w),h\right),\\ F(w,c_h)+F\left(w,q(h-c_h)\right),\\ C(w,h),\\ c_w\in T_x(w),c_h\in T_y(h),\\ w_{min}\le c_w\le w/2,h_{min}\le c_h\le h/2 \end{array}\right\} \quad (11)$$

In the process of solving the C-subproblem, the value of the cutting point stops at half of the size of the intermediate plate, which is the result of avoiding the symmetrical cutting pattern of the C-plate [Herz, 1972]. In addition, each time the subproblem is divided, the size of the right (upper) sub-plate can be further reduced, as shown in equation (12).

$$\begin{cases} p(w)=\max\left(0,c_w\mid c_w\le w,c_w\in T_x(w)\right)\\ q(y)=\max\left(0,c_h\mid c_h\le h,c_h\in T_y(h)\right)\end{cases} \quad (12)$$

$p(w)$ represents the position of the cutting point closest to the right of the intermediate plate while $q(h)$ represents the position of the cutting point closest to the upper edge of the intermediate plate. This method limits the size of the plate to a discrete set, so that some different subproblems can be transformed into repeated subproblems. It does not reduce the quality of the solution obtained by the algorithm, and improves the efficiency of calculation [Beasley, 1985].

Therefore, the recursive function for solving the *D*-subproblem is defined as follows.

$$F(x_0,y_0,w,h)=\left\{\begin{array}{l} F(w,h),\\ \quad if\ R=(x_0,y_0,w,h)\,is\ C-plate;\\ \max\left\{\begin{array}{l} F(x_0,y_0,c_w,h)+\\ F\left(x_0+c_w,y_0,p(w-c_w),h\right),\\ F(x_0,y_0,w,c_h)+\\ F\left(x_0,y_0+c_h,w,q(h-c_h)\right),\\ c_w\in T_x(w),c_h\in T_y(h),\\ 1\le c_w\le w-1,1\le c_h\le h-1 \end{array}\right\},\\ \quad if\ R=(x_0,y_0,w,h)\,is\ D-plate. \end{array}\right. \quad (13)$$

Since the cutting point separating the defect from the items may appear at the edge of the intermediate plate, the selection range of discrete sets in equation (13) is [1, *w*-1] and [1, *h*-1]. It is clear that $F(x_0,y_0,w,h)$ is also used to solve the original problem R=(0, 0, W_0, H_0).

4 Experiment

The experiment is coded in C++ and runs and evaluates EDDP in the environment of i5-4200H CPU 2.80GHZ.

Class	*m*	ρ	DPC		DPD		EDDP	
			OFV	CT(s)	OFV	CT(s)	OFV	CT(s)
1	5	6	**59589.37**	5615.48	58707.6	123.68	**59589.37**	0.01125
2	10	6	**65763.73**	5542.994	64857.25	225.89	**65763.73**	0.1575
3	15	6	**69424.47**	7015.74	68832.68	265.06	**69424.47**	0.47125
4	20	6	**72494.38**	7403.52	71708.11	501.14	**72494.38**	2.56475
5	25	6	**75017.87**	7312.41	74345.13	505.45	**75017.87**	4.195
6	5	8	**57585.43**	5255.65	56809.91	96.13	**57585.43**	0.01175
7	10	8	**69057.08**	6790.57	68244.58	311.65	**69057.08**	1.986
8	15	8	**73679.27**	8888.12	73047.51	529.24	**73679.27**	3.71375
9	20	8	**75528.63**	8649.94	74767.43	819.22	**75528.63**	9.43675
10	25	8	**77178.77**	8459.96	76713.05	1422.33	**77178.77**	28.723
11	5	10	**59970.58**	5933.43	59584.45	150.97	**59970.58**	0.09725
12	10	10	**73352.73**	7444.78	72717.7	398.99	**73352.73**	0.9645
13	15	10	**74058.07**	8096.66	73389.63	577.09	**74058.07**	4.531
14	20	10	**78722.25**	9313.24	78297.61	1545.06	**78722.25**	44.00575
15	25	10	**78921.28**	9852.68	78211.6	1558.21	**78921.28**	52.9255
16	5	6	**55345.02**	7974.38	54692.4	122.87	**55345.02**	0.00625
17	10	6	**68157.33**	9686.68	67097.4	297.51	**68157.33**	0.23775
18	15	6	**72302.37**	12955.8	71595.45	417.33	**72302.37**	0.94425
19	20	6	**71964.07**	12039.17	71311.08	384.57	**71964.07**	1.732
20	25	6	**75596.92**	12915.62	75154.26	844.27	**75596.92**	7.8115
21	5	8	**58758.78**	7249.74	57095.48	120.32	**58758.78**	0.01175
22	10	8	**68186.45**	11271.39	67306.91	330.42	**68186.45**	0.82675
23	15	8	**73113.17**	13020.78	72500.68	653.24	**73113.17**	4.50075
24	20	8	**77282.15**	13083.66	76691.43	957.87	**77282.15**	12.03525
25	25	8	**78269.4**	13094.77	77629.33	1268.14	**78269.4**	33.48175
26	5	10	**64746.97**	9467.29	63169.15	154.15	**64746.97**	0.07625
27	10	10	**67936.45**	12128.87	67114.36	378.62	**67936.45**	0.92
28	15	10	**76180.58**	13044.18	75415.03	849.88	**76180.58**	7.82975
29	20	10	**78203.08**	12457.37	77600.43	1098.54	**78203.08**	32.81775
30	25	10	**78254.72**	12714.31	77686.03	1509.38	**78254.72**	65.31825
Average			**70821.379**	9489.306	70076.455	613.907	**70821.379**	10.745

Notes: bold is the best solution; OFV means Objective Function Values; CT means Cost Time。

Table 1: Comparison of the results of three algorithms on 1800 large-scale instances

The instances were taken from the largest set of instances generated by Afsharian [Afsharian *et al*., 2014], and can be found at www.dep.ufscar.br/munari/cuttingpacking. The dimensions of the original plates in the instances are all (300, 300) or (450, 200), and their total number contains 1800 instances based on item information, defect information, and other factors.

Based on the analysis of experimental results of several current state-of-the-art algorithms in the literature [Afsharian *et al*., 2014; Martin *et al*., 2020], two typical algorithms are selected for comparison with the EDDP in this paper. One is DPD, which is the algorithm with the best overall quality, and the other is DPC, which is a fully dynamic programming algorithm that obtains the optimal solution exactly.

The computational results of the above three algorithms are shown in Table 1 above. Where DPD fails to obtain the optimal solution for a set of instances, EDDP obtains the optimal solution for all instances. In terms of computational time efficiency, the average computation time of DPD is 613 seconds, while the average computation time of EDDP is 10 seconds, which is more than 60 times faster than that of DPD. Overall, the significant advantage of EDDP is demonstrated both in terms of the quality of the computational results and in terms of the computational speed.

5 Discussion and Conclusion

This paper discusses the two-dimensional cutting problem with defects under guillotine constraints. First, an efficient discrete set is proposed and a normalized pattern of the defected plate is obtained, which ensures the optimality of the algorithm and is an important breakthrough in solving the cutting problem with defects. Second, a simpler method for lower bounding the subproblem is designed and the K-table for storing the lower bound is constructed to further improve the operation speed of the algorithm. The experimental results show that EDDP obtains the optimal solution while the computational speed is much faster than similar algorithms, providing a quality solution for this type of problem.

References

[Afsharian *et al*., 2014] Mohsen Afsharian, Ali Niknejad, and Gerhard Wäscher. A heuristic, dynamic programming-based approach for a two-dimensional cutting problem with defects. *OR spectrum*, 36(4):971–999,2014.

[Aksu and Durak, 2016] Dilek Tuzun Aksu and Bahadir Durak. A dynamic programming algorithm for the online cutting problem with defects and quality grades. *IFAC-PapersOnLine*, 49(12):17–22, 2016.

[Beasley, 1985] JE Beasley. Algorithms for unconstrained two-dimensional guillotine cutting. *Journal of the Operational Research Society*, 36(4):297–306, 1985.

[Carnieri *et al.*, 1993] Celso Carnieri, Guillermo A Mendoza, and William G Luppold. Optimal cutting of dimension parts from lumber with a defect: a heuristic solution procedure. *Forest Products Journal*, 43(9):66, 1993.

[Christofides and Whitlock, 1977] Nicos Christofides and Charles Whitlock. An algorithm for two-dimensional cutting problems. *Operations Research*, 25(1):30–44, 1977.

[Gilmore and Gomory, 1966] PC Gilmore and Ralph E Gomory. The theory and computation of knapsack functions. *Operations Research*, 14(6):1045–1074, 1966.

[Gonçalves and Wäscher, 2020] José Fernando Gonçalves Gerhard Wäscher. A MIP model and a biased random-key genetic algorithm based approach for a two-dimensional cutting problem with defects. *European Journal of Operational Research*, 286(3):867–882, 2020.

[Herz, 1972] JC Herz. Recursive computational procedure for two-dimensional stock cutting. *IBM Journal of Research and Development*, 16(5):462–469, 1972.

[Iori *et al*., 2007] Manuel Iori, Juan-José Salazar-González, and Daniele Vigo. An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation science*, 41(2):253–264, 2007.

[Kellerer *et al.*, 2004] Hans Kellerer, Ulrich Pferschy, and David Pisinger. Multidimensional knapsack problems. *In Knapsack problems*, pages 235–283. Springer, 2004.

[Martin *et al*., 2020] Mateus Martin, Pedro HDB Hokama, Reinaldo Morabito, and Pedro Munari. The constrained two-dimensional guillotine cutting problem with defects: an ILP formulation, a benders decomposition and a CP based algorithm. *International Journal of Production Research*, 58(9):2712–2729, 2020.

[Morabito and Pureza, 2010] Reinaldo Morabito and Vitória Pureza. A heuristic approach based on dynamic programming and and/or-graph search for the constrained two-dimensional guillotine cutting problem. *Annals of Operations Research*, 179(1):297–315, 2010.

[Na *et al*., 2014] Byungsoo Na, Shabbir Ahmed, George Nemhauser, and Joel Sokol. A cutting and scheduling problem in float glass manufacturing. *Journal of Scheduling*, 17(1):95–107, 2014.

[Neidlein *et* al., 2008] Vera Neidlein, Andrèa CG Vianna, Marcos N Arenales, and Gerhard Wäscher. The two-dimensional, rectangular, guillotineable-layout cutting problem with a single defect. *Working Paper Series*, 2008.

[Wu *et* al., 2019] Keqiang Wu, Xiaoping Min, and Defu Zhang. Research on two-dimensional cutting problem with defects. In *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, pages 1–8. IEEE, 2019.

[Wuttke and Heese, 2018] David A Wuttke and H Sebastian Heese. Two-dimensional cutting stock problem with sequence dependent setup times. *European Journal of Operational Research*, 265(1):303–315, 2018.

[Yin *et al*., 2020] Aihua Yin, Chong Chen, Dongping Hu, Jianghai Huang, and Fan Yang. An improved heuristic-dynamic programming algorithm for rectangular cutting problem. *Computer Science and Information Systems*, 17(3):717–735, 2020.