



A Quick Heuristic-Dynamic Programming for The Two-Dimensional Cutting Problem*

Yin Aihua, Huang Jianghai, Hu Dongping and Chen Chong

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

May 1, 2019

求解矩形切割问题快速启发式-动态规划算法*

尹爱华, 黄江海⁺, 胡冬萍, 陈冲

江西财经大学 软件与物联网工程学院, 南昌 330013

+ 通讯作者 E-mail: 1982589849@qq.com, 硕士生

摘要: 讨论了含砂眼的二维矩形切割问题。目标是在满足若干约束条件的前提下从含有多个砂眼的大矩形原板上切割出已给定高和宽的小矩形块, 使得切割下来的小矩形块的面积和最大。约束条件为每一次切割操作必须是一刀切并且不能穿过砂眼, 每一类小矩形块的数量不限且保持给定的方向。将该问题看做是用矩形块覆盖原板, 提出一种基于动态规划的快速启发式算法(Quick Heuristic-Dynamic Programming, QHDP)。首先, 根据小矩形块的高和宽建立一维背包问题, 分别生成高效的离散集, 然后, 以离散集中的每一个值作为可能的切割线坐标进行子问题划分。如果某个切割线穿过砂眼, 则剪切掉该递归分支。该算法计算了国际公认的 14 个典型算例, 实验结果表明, 它都得到了全部算例的最优解, 且计算时间少于当前文献中最好算法的十分之一。算法复杂度得到分析和证明。

关键词: 二维 cutting 问题; 砂眼; 最优化; 动态规划; NP-难

文献标志码: A **中图分类号:** TP301.5

A Quick Heuristic-Dynamic Programming for The Two-Dimensional Cutting Problem*

YIN Aihua, HUANG Jianghai⁺, HU Dongping, CHEN Chong

School of Software and Internet of Things Engineering, Jiangxi University of Finance and Economics, Nanchang Jiangxi 330013

Abstract: The two-dimensional rectangular cutting problem with defects is discussed. The goal is to cut a small rectangular block of a given height and width from a large rectangular object containing a plurality of defects on the premise of satisfying several constraints, so that the sum of the area of the cut small rectangular blocks are maximized. The constraint is that each cutting operation must be guillotine and cannot pass through the defects, and the number of small rectangular blocks of each type is not limited and maintains a given direction. The problem is regarded as covering the original plate with small rectangular block, and a Quick Heuristic-Dynamic Programming (QHDP) algorithm is proposed. Firstly, a one-dimensional knapsack problem is established according to the height and width of small rectangular blocks, respectively, and an efficient discrete set is generated respectively. Then, each

*National Natural Science Foundation project Nos. 61862027, 矩形布局问题的全局搜索关键技术; Nos.61702238, 云存储中支持模糊查询的可搜索对称加密研究; Nos.61866014, 面向大规模调度问题的并行参数自适应萤火虫优化算法。

value in the discrete set is used as a possible cutting line coordinate for sub-problem division. If a cutting line passes through the defects, the recursive branch is cut off. The algorithm calculates 14 internationally accepted examples. The experimental results show that it has obtained the optimal solution of all the examples, and the calculation time is less than one tenth of the best algorithm in the current literature. The algorithm complexity is analyzed and proved.

Key words: two-dimensional cutting; defects; optimization; dynamic programming; NP-hard

1 引言

Cutting 问题是一个典型的 *NP-hard* 问题^[1]。在二维欧式空间中,对已知高和宽的大矩形板块和一定数量的小矩形块,在满足若干约束条件的前提下在大矩形板块上切割若干小矩形块,使得切割小矩形块的总面积最大化,同时计算时间达到实际应用的要求。

在实际生产应用领域中, cutting 问题以各种形态存在^[2],例如玻璃、木板和钢板工艺等领域的切割加工需要以节约成本为目的来控制原材料使得利用率达到最大。因此,设计该类问题的高效算法对生活加工领域具有实际价值。

求解二维矩形 cutting 问题(two-dimensional, rectangular single large object placement problem, 2D_SLOPP)的算法分为两大类,约束和非约束,本文讨论的是约束的二维矩形 cutting 问题(2D_SLOPP_D),可以找到最优解,但是约束条件使得所花计算时间较长,因此适用于中小型规模的问题计算。

国内外许多学者采用启发式算法来解决 2D_SLOPP 问题^[3-5],这些启发式规则为该问题的更加深入研究奠定了一定的基础。

Gilmore 等人^[6]运用背包模型计算出二维无限制矩形切割问题,他们得出了利用背包模型快速计算这类问题的方法是可行的。Herz^[7]实现了递归求解切割问题,首次提出离散集作为驱动进行这类问题的求解,在保证优度的同时还提高了计算速度。之后 Beasley^[8]在 Herz 的基础上修改离散集,在算法中限制了切割数,提高了递归过程的性能。崔耀东等人^[9-12]利用递归算法生成最优 T 形排样方式和最优两段排样方式的递归算法,采用由条带组成的 T 形排样讨论了矩形无约束二维切割问题,结果表明在计算时间和优度上都有很好的效果体验。但是他们都是基于无砂眼的问题进行研究的。

对于 2D_SLOPP_D, Carnieri 等人^[13]在 Gilmore

的基础上提出了动态规划的启发式算法,考虑了砂眼的位置,大大加快了计算速率。Neidlein 等人^[14]继续沿用了 Herz 的离散集,减少了离散集的数量,但是他们只涉及一个砂眼,这就包含了几个结构和计算限制,如果处理多个砂眼的情况则会受到结构和计算时间的限制。Afsharian 等人^[15]基于动态规划的启发式算法求解二维限制的切割问题,他们克服了结构和计算时间的限制,重新给出了砂眼区域和离散集的定义。

本文在 Afsharian 的方法上进一步对离散集进行优化,并对如何规避砂眼区域做出了说明,实验表明在优度和计算时间上都有了很大的突破。提出了基于动态规划的启发式方法,在砂眼的存在下,进行试切操作,对于每一次的操作所得到的两个子块再一次进行递归操作,直到子块不能再切为止。如果切割刀砂眼,则继续执行下一刀。用这个方法,在保证优度的前提下同时在计算时间上也能得到有效提升。以离散集作为驱动,所以,离散集的长短决定了计算时间的多少。换言之,减少离散集长度可以有效降低计算量,同时还要保证优度不能降。本文就是通过判断砂眼的位置来对离散集进行合理地删减从而减少了计算量,得到了最优值。

2 问题描述

2D_SLOPP_D 问题具体描述如下^[16]:

在高为 H , 宽为 W 且含有砂眼的矩形件上切割 m 种不同的的小的矩形块,小矩形块的高和宽分别为 h_i 和 $w_i(i = 1, 2, \dots, m)$ 以及每个小矩形块本身具有的权值 s_i 。其目标是从矩形原件上切割下来的小矩形块的权值总和达到最大。

作为一个标准的切割问题,在每一次切割时必须满足从板块的一个边缘到另一个边缘的切割是完整的,并且切割位置平行于板块的边缘,即一刀切类型。同时,问题必须满足以下条件:

- 切割下来的每一类小矩形块的数量不受限

制。

- 小矩形块之间不允许重叠。
- 小矩形块的高和宽是固定的，不允许将其进行旋转，例如必须按照垂直方向高 h 、宽 w 方式切割，不得按照高 w 、宽 h 切割， $w \neq h$ 。
- 切割下来的小矩形块不能含有砂眼，否则为废料。
- 任何的切割动作不能穿过砂眼。

图 1 表示一刀切和非一刀切。

该问题的目标函数如下^[17]。

$$\max \sum_{i=1}^m s_i \alpha_i; \quad \alpha_i \in \mathbb{Z}_+, \quad i=1, 2, 3, \dots, m. \quad (1)$$

α_i 为非负整数，表示切割下来的第 i 类小矩形块的数量。 s_i 表示第 i 类小矩形块的面积，是正整数。

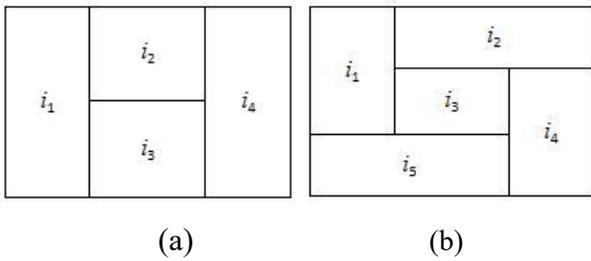


Fig.1 Guillotine(a) and Non-Guillotine(b)

图 1 一刀切(a)和非一刀切(b)

3 算法描述

以原板的左下角为坐标原点 $(0, 0)$ 建立笛卡尔坐标系， X 轴为原板的宽的方向， Y 轴为原板高的方向。

3.1 基本概念

定义 1 (子板) 将用于做切割动作的高度为 h 宽度为 w 的矩形板块称之为子板，任何的切割动作都是在子板上完成的。子板可以再分解成若干个更小的子板。子板上可能存在砂眼，为了区分有砂眼和没有砂眼且尺寸相同的子板，加入子板左下角坐标信息用来表示某一块子板，所以，子板表示为四元组 $P = (ox, oy, w, h)$ 。对于原始板块表示为 $P = (0, 0, W, H)$ 。

定义 2 (砂眼) 砂眼存在于任意子板上，假设每一个砂眼在子板上是矩形的^[18]，并且砂眼的各条边平行或垂直于子板的边缘，砂眼不能被分割，

砂眼的绝对位置是固定不变的而且包含砂眼的子板不能被计算在结果内，否则将会当做废料处理。砂眼的信息用四元组 $D = (x_i^d, y_i^d, h_i^d, w_i^d)$ 表示，其中 (x_i^d, y_i^d) 是砂眼左下角的坐标， (h_i^d, w_i^d) 是砂眼在垂直方向的高和水平方向的宽。

图 2 是子板和砂眼的坐标和尺寸表示。

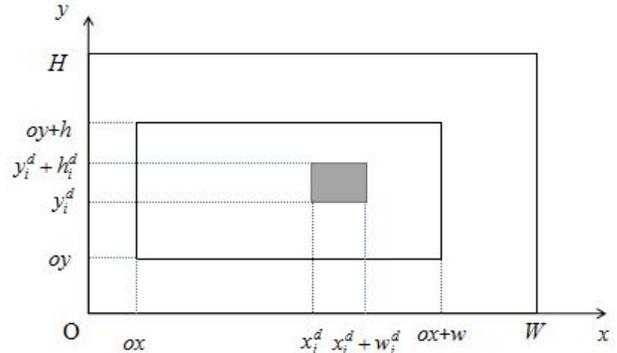
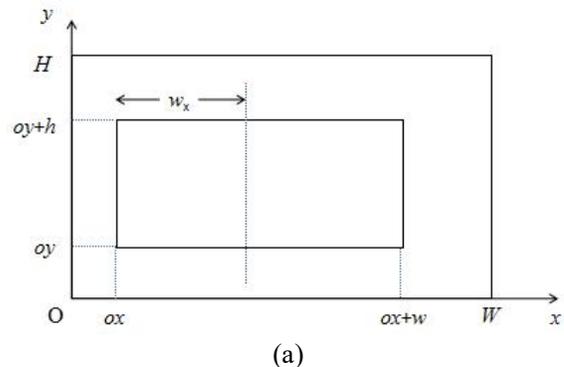


Fig.2 Representation of sub-plate and defects

图 2 子板和砂眼的表示

定义 3 (P-板和 C-板) 子板标记为两种，一种是不含砂眼的叫 P (pure)板，另一种是带有砂眼的叫 C (Contaminated)板。对于相同尺寸 P -板将不再重复计算，而相同尺寸 C -板的由于砂眼的位置和尺寸的不同将继续递归计算。经过这样区分可以降低计算量。

定义 4 (切割线) 每一次竖直或水平切割将子板一分为二，得到的两块子板的分界线即为切割线，用 w_x 或 h_y 表示。对于某块子板 $P = (ox, oy, w, h)$ ，如果是竖直切割则生成的左右两块子板表示分别为 (ox, oy, w_x, h) 和 $(ox + w_x, oy, w - w_x, h)$ ，如果是水平切割则生成的上下两块子板表示分别为 (ox, oy, w, h_y) 和 $(ox, oy + h_y, w, h - h_y)$ 。切割线仅对当前子板而言，切割线示意图如图 3。



(a)

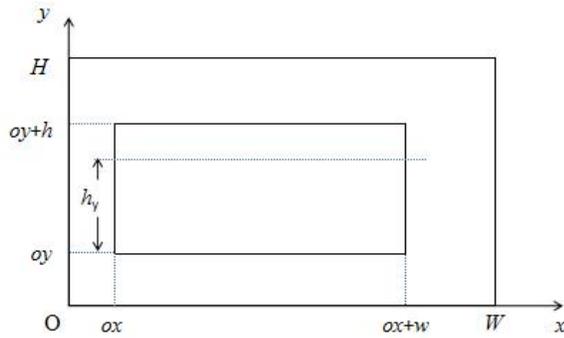


Fig.3 Dimensions and locations of vertical/horizontal
图 3 竖直切割(a)和水平切割(b)

定义 5 (离散集) 对当前子板, 我们把从竖直(水平)切割线用 $w_x(h_y)$ 表示, 当前同一方向所有的切割位置是一个有序的集合。最新研究表明, 离散集的大小决定了计算次数的可能性, 离散集的数量越多可能导致计算时间越长。此外, 若当前切割位置穿过某个砂眼, 则跳过当前试切动作继续考虑下一个元素。我们用 $\psi_x(w)$ 表示子板在 x 轴方向竖直切割线的集合, $\psi_y(h)$ 表示子板在 y 轴方向水平切割线的集合, 离散集定义如下:

$$\psi_x(w) = \left\{ \begin{array}{l} w_x \mid w_x = \sum_{i=1}^m \alpha_i w_i, 1 \leq w_x \leq w - w_0, \alpha_i \in Z_+ \cup \{0\}, \\ \text{if } (h_i > h \text{ or } w_i > w) \Rightarrow \alpha_i = 0, \forall i \end{array} \right\} \quad (2)$$

$$\psi_y(h) = \left\{ \begin{array}{l} h_y \mid h_y = \sum_{i=1}^m \beta_i h_i, 1 \leq h_y \leq h - h_0, \beta_i \in Z_+ \cup \{0\}, \\ \text{if } (h_i > h \text{ or } w_i > w) \Rightarrow \beta_i = 0, \forall i \end{array} \right\} \quad (3)$$

w_0 和 h_0 分别表示最小矩形块的宽和高, α_i 和 β_i 为非负整数分别表示第 i 类小矩形块的数量。公式表示如下:

$$h_0 = \begin{cases} \min(h_i); & i = 1, 2, \dots, m, \\ \text{where } h_i \leq y \text{ and } w_i \leq x, \end{cases} \quad (4)$$

$$w_0 = \begin{cases} \min(w_i); & i = 1, 2, \dots, m, \\ \text{where } w_i \leq x \text{ and } h_i \leq y, \end{cases} \quad (5)$$

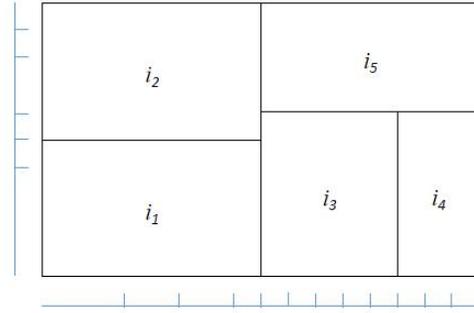


Fig.4 Cutting position sets of vertical/horizontal
图 4 横向和纵向的切割位置集

每个试切位置都是根据小矩形块的尺寸组合得到的, 并经过试切计算后得到图 4 所展示的切割方式。

公式(2)、(3)是完全背包问题, 可以用动态规划求解^[19]。

Herz 在所研究的是基于无砂眼的矩形切割, 在提出离散集时只需将小矩形块的尺寸进行组合从而得到切割位置信息。在本文有砂眼的情况是存在的, 而砂眼不能被计算在结果内并且不能被切割线穿过, 这就可能导致部分切割方式被遗弃继而可能损失最优值。

在确定切割线 w_x 或 h_y 时可能会穿过砂眼, 当这种情况出现时跳过当前切割线继续试切下一元素, 如图 5 所示, 如果竖直切割线 w_x 穿过砂眼, 则放弃当前动作继续执行切割线 w'_x 。水平切割线的确定同理。

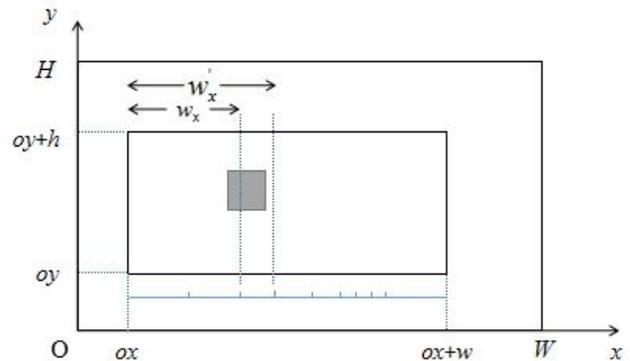


Fig.5 If a cutting line passes through a defect, the next cutting line is tested

图 5 如果割线穿过砂眼时就跳过当前试切动作

判断切割线是否穿过砂眼如下表示:

$$if_pass = \begin{cases} 1, & \text{if } (x_i^d - ox < w_x \text{ and } w_x < x_i^d + w_i^d - ox) \text{ or} \\ & (y_i^d - oy < h_y \text{ and } h_y < y_i^d + h_i^d - oy), \forall i \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

当 if_pass 值为 1 时表示切割线穿过砂眼，否则未穿过。

3.2 基本算法

任何一个子板都可以分解为若干个小的方块，用 $f(w, h)$ 表示子板 w (宽) \times h (高)经过切割后可得到的最大价值， α_i 为非负整数，表示切割第 i 类方块的个数， s_i 为第 i 类方块的权值，则有如下公式：

$$f(w, h) = \sum_{i=1}^m s_i \alpha_i, \quad w \leq W, h \leq H \quad (9)$$

若子板 $P = (ox, oy, w, h)$ 没有砂眼，则尝试在全部切割同一类型的小矩形块，方法是遍历所有类型的矩形块并依次计算每一类最大“覆盖”整块子板所得到的值，取最大的一个用 $g(w, h)$ 表示，当前结果跟之后试切的结果做比较，最大值即为 $f(w, h)$ 。具体公式表示如下：

$$\begin{cases} g(w, h) = \max(0, \lfloor \frac{w}{w_i} \rfloor \cdot \lfloor \frac{h}{h_i} \rfloor \cdot s_i) \\ i = 1, 2, 3, \dots, m \end{cases} \quad (10)$$

将当前子板水平(竖直)切割成上下(左右)两个高度(宽度)相同的两个子板，对于水平切割形成的子块的最大价值总和记做 $f_y(w, h)$ ，对于竖直切割形成的子块的最大总价值和记做 $f_x(w, h)$ 。每次切割形成的子块都可以当做单独的个体，并且这两个子块所分解的小矩形块的总面积和等于切割前的子板。

则有如下公式：

$$f_x(ox, oy, w, h) = \begin{cases} \max \{ f(ox, oy, w_x, h) + f(ox + w_x, oy, w - w_x, h), \\ f(ox, oy, w - w_x, h) + f(ox + w - w_x, oy, w_x, h) \}, \\ 1 < w_x < w, \text{ 子板是 } C\text{-板}; \\ \max \{ g(w, h), f(w_x, h) + f(w - w_x, h) \}, \\ 1 < w_x \leq \frac{w}{2}, \text{ 子板是 } P\text{-板}; \\ w_x \in \psi_x(x), w \leq W, h \leq H; \end{cases} \quad (11)$$

$$f_y(ox, oy, w, h) = \begin{cases} \max \{ f(ox, oy, w, h_y) + f(ox, oy + h_y, w, h - h_y), \\ f(ox, oy, w, h - h_y) + f(ox, oy + h - h_y, w, h_y) \}, \\ 1 < h_y < h, \text{ 子板是 } C\text{-板}; \\ \max \{ g(w, h), f(w, h_y) + f(w, h - h_y) \}, \\ 1 < h_y \leq \frac{h}{2}, \text{ 子板是 } P\text{-板}; \\ h_y \in \psi_y(y), w \leq W, h \leq H; \end{cases} \quad (12)$$

$$f(ox, oy, w, h) = \max(f_x(ox, oy, w, h), f_y(ox, oy, w, h)) \quad (13)$$

式(11)中 w_x 表示竖直切割位置，式(12)中 h_y 表示水平切割的位置。

若子板为 P -板，当满足 $w < w_0$ 或 $h < h_0$ 时，则有 $f(w, h) = 0$ 成立。若子板为 C -板，当砂眼四周与子板边缘的相对距离均小于最小矩形块的尺寸时，则有 $f(ox, oy, w, h) = 0$ 成立，判断当前子板是否继续切割的公式如下：

$$return = \begin{cases} 1, & \text{if } (w < w_0 \text{ or } h < h_0) \\ & \text{or } (x_i^d - ox < w_0 \text{ and } ox + w - x_i^d - w_i^d \\ & y_i^d - oy < h_0 \text{ and } oy + h - y_i^d - h_i^d); \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

公式(14)中当 $return$ 值为 1 时停止计算返回子板的计算结果，否则继续递归计算。

定理 1 该问题最坏情况的时间复杂度是 $O(|\psi_x(W)| |\psi_y(H)| (|\psi_x(W)| + |\psi_y(H)|))$

证明 对于一块尺寸为 (W, H) 的原板可能横切或竖切，这就有 $|\psi_x(W)| |\psi_y(H)|$ 种可能性，在每一次试切后递归每一块子板，所以整体时间复杂度是 $O(|\psi_x(W)| |\psi_y(H)| (|\psi_x(W)| + |\psi_y(H)|))$

定理 2 假设 $\rho_w = \lfloor W/w_0 \rfloor$ 表示水平方向最多切割的小矩形块数量， $\rho_h = \lfloor H/h_0 \rfloor$ 表示竖直方向最多切割的小矩形块数量，则有 $|\psi_x(W)| \leq \sum_t^{\rho_w} C_{t+m-1}^t$ 和 $|\psi_y(H)| \leq \sum_t^{\rho_h} C_{t+m-1}^t$ 成立。

证明 根据离散集定义可知， $\psi_x(W)$ 中的元素是有小矩形块的宽组合得到的，该集合的项数满

足多项式 $\sum_{t=1}^{\rho_w} \left(\sum_{i=1}^m w_i \right)^t$ 的项数 ($t = 1, 2, \dots, \rho_w$)。根据问题可以知道 $\sum_{i=1}^m \alpha_i w_i \leq W$ ，所以对于任意正整数 t 在多项式 $\left(\sum_{i=1}^m w_i \right)^t$ 中的项数是 C_{t+m-1}^t [20]，即多项式 $\sum_{t=1}^{\rho_w} \left(\sum_{i=1}^m w_i \right)^t$ 的项数是 $\sum_{t=1}^{\rho_w} C_{t+m-1}^t$ 。因此 $|\psi_x(W)| \leq \sum_{t=1}^{\rho_w} C_{t+m-1}^t$ 是成立的，同理 $|\psi_y(H)| \leq \sum_{t=1}^{\rho_h} C_{t+m-1}^t$ 也是成立的。

4 实验结果

实验用主频 2.6GHz, RAM 4G 的 PC 机做实验, 使用 C 语言进行编码。

使用文献[14]中实现的 14 组算例, 算例中分别有单个和多个砂眼。计算结果分别与文献[14]的 AND/OR graph approach 算法和文献[15]的 DPD 算法进行优度和计算时间(time)两方面的比较。原板宽度和高度分别为 $W = 200, H = 100$ 。计算结果如表 1 所示。

Table 1 Comparing with Neidlein and other algorithms
表 1 和 Neidlein 等两个算法比较

算例	AND/OR graph approach		DPD		QHDP	
	优度	计算时间(s)	优度	计算时间(s)	优度	计算时间(s)
	A1	166	0.52	166	18.86	166
A2	160	0.77	166	16.43	166	0.16
A3	162	1.77	166	16.47	166	0.14
A4	160	0.27	164	18.25	164	0.03
A5	164	4.11	164	76.92	164	0.09
A6	164	1.44	164	0.90	164	0.02
A7	158	1.07	158	0.81	158	0.08
A8	154	0.50	154	1.21	154	0.07
A9	-	-	160	14.32	160	0.17
A10	-	-	158	2.22	158	1.90
A11	-	-	151	26.78	151	6.08
A12	-	-	156	1126.44	156	3.07
A13	-	-	150	9.06	150	0.09
A14	-	-	160	1.00	160	0.22

注: 符号“-”表示对应的数据没有公布

从表 1 可以看出, 在 14 个测试算例中, 在优度上, 算例 2、3 和 4 在优度上均优于 Neidlein 等人,

除此之外, 计算时间全部优于 AND/OR graph approach。和 DPD 算法的比较, QHDP 算法优度保持一样, 计算时间少于 DPD 算法的十分之一, 其中, 算例 12 差别最为明显。14 个算例的切割方案由附录展示。

5 结束语

本文基于动态规划处理二维矩形 cutting 问题, 重新对离散集进行定义在不影响优度的前提下削减切割过程中的搜索宽度。其次, 对于相同的子板而言仅计算一次, 每次对子板试切前做判断, 对之前计算过的子板直接使用计算结果而避免重复计算, 这样又减少了搜索深度。最后, 通过考虑砂眼的位置及尺寸来判断子板有无砂眼以及在试切过程是否因切到砂眼而造成不必要的计算。综合以上因素, 该方法拥有优度和计算时间上的双重优势, 算法的表现优于目前文献中所发表的最好结果。在今后的工作中, 拟进一步提升算法效率, 考虑在中大规模的实例也能具有时间优势。

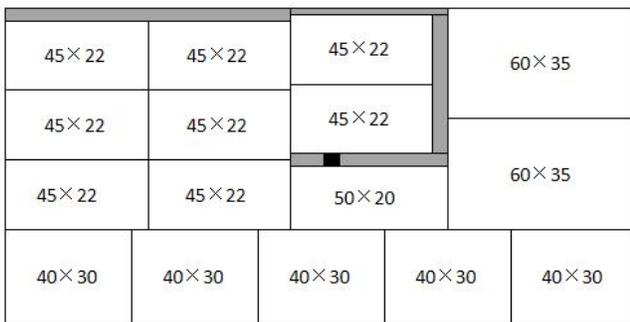
References

- [1] Na B, Ahmed S, Nemhauser G, et al. A cutting and scheduling problem in float glass manufacturing[J]. Journal of Scheduling, 2014, 17(1):95-107.
- [2] Wuttkte D A, Heese H S. Two-dimensional cutting stock problem with sequence dependent setup times[J]. European Journal of Operational Research, 2017:S037722171730663X.
- [3] Wei L, Hu Q, Lim A, et al. A best-fit branch-and-bound heuristic for the unconstrained two-dimensional non-guillotine cutting problem[J]. European Journal of Operational Research, 2018:S0377221718303175.
- [4] Park K T, Ryu J H, Lee H K, et al. Developing a heuristics for glass cutting process optimization: A case of two-dimensional two-stage guillotine cutting with multiple stock sizes[J]. Korean Journal of Chemical Engineering, 2013, 30(2):278-285.
- [5] Xu Y, Yang G, Pan C. A Heuristic Based on PSO for Irregular Cutting Stock Problem[J]. IFAC Proceedings Volumes, 2013, 46(13):473-477.
- [6] Gilmore P C, Gomory R E. The Theory and Computation of Knapsack Functions[J]. Operations Research, 1966, 14(6):1045-1074.
- [7] Herz, J. C. Recursive Computational Procedure for Two-dimensional Stock Cutting[J]. IBM Journal of Research and Development, 1972, 16(5):462-469.
- [8] Beasley, J. E. Algorithms for Unconstrained Two-Dimensional Guillotine Cutting[J]. Journal of the Operational Research Society, 1985, 36(4):297-306.

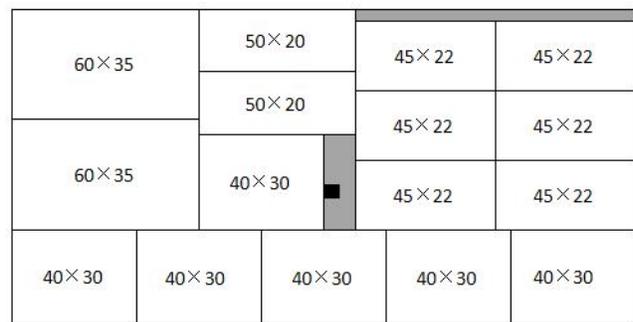
- [9] Cui Yaodong. Recursive Algorithm for Generating Optimal T-Shape Cutting Patterns of Rectangular Blanks[J]. JOURNAL OF COMPUTER-AIDED DESIGN & COMPUTER GRAPHICS, 2006, 18(1):125-127.
(崔耀东. 生成矩形毛坯最优 T 形排样方式的递归算法[J]. 计算机辅助设计与图形学学报, 2006, 18(1):125-127.)
- [10] Cui Yaodong, JiJun, Zeng TiaoJun. Recursive Algorithm for Generating Optimal Two-Segment Cutting Patterns of Rectangular Blanks[J]. Journal of Nanjing University of Aeronautics & Astronautics, 2006(01):111-114.
(崔耀东, 季君, 曾窈俊. 生成矩形毛坯最优两段排样方式的递归算法[J]. 南京航空航天大学学报, 2006, 38(1).)
- [11] Cui Y. Dynamic programming algorithms for the optimal cutting of equal rectangles[J]. Applied Mathematical Modelling, 2005, 29(11):1040-1053.
- [12] Cui Y, He D, Song X. Generating optimal two-section cutting patterns for rectangular blanks[J]. Computers and Operations Research, 2006, 33(6):1505-1520.
- [13] Carnieri C, Mendoza G A, Luppold W G. Optimal cutting of dimension parts from lumber with a defect: A heuristic solution procedure[J]. Forest Products Journal, 1993.
- [14] Neidlein, V. Vianna, A.C.G.; Arenales, M.N.; Wäscher, G. The Two-Dimensional Guillotineable-Layout Cutting Problem with a Single Defect - An AND/OR-Graph Approach. In[J]: Operations Research Proceedings 2008 (Eds.: Fleischmann, B. et al.). Berlin, Heidelberg: Springer-Verlag, 85-90.
- [15] Afsharian M, Niknejad A, Wäscher G. A heuristic, dynamic programming-based approach for a two-dimensional cutting problem with defects[J]. OR Spectrum, 2014, 36(4):971-999.
- [16] Morabito R, Vitória Pureza. A heuristic approach based on dynamic programming and and/or-graph search for the constrained two-dimensional guillotine cutting problem[J]. Annals of Operations Research, 2010, 179(1):297-315.
- [17] Song X, Chu C B, Lewis R, et al. A worst case analysis of a dynamic programming-based heuristic algorithm for 2D unconstrained guillotine cutting[J]. European Journal of Operational Research, 2010, 202(2):368-378.
- [18] Beasley J E. A population heuristic for constrained two-dimensional non-guillotine cutting[J]. European Journal of Operational Research, 2004, 156(3):601-627.
- [19] Caprara A, Monaci M. On the two-dimensional Knapsack Problem[J]. Operations Research Letters, 2004, 32(1):5-14.
- [20] Hohn F E. The Number of Terms in a Polynomial[J]. American Mathematical Monthly, 1941, 48(10):686-687.

附录

算例 A1~A14 的最优解切割模式



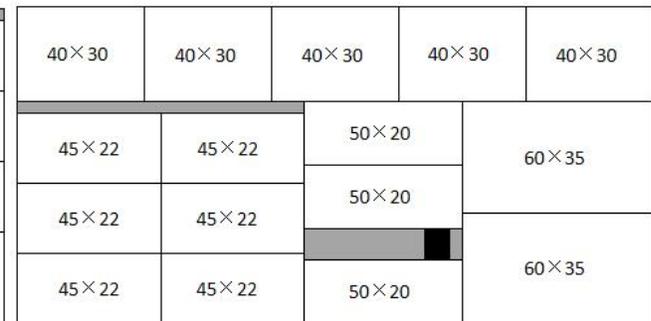
算例 A1



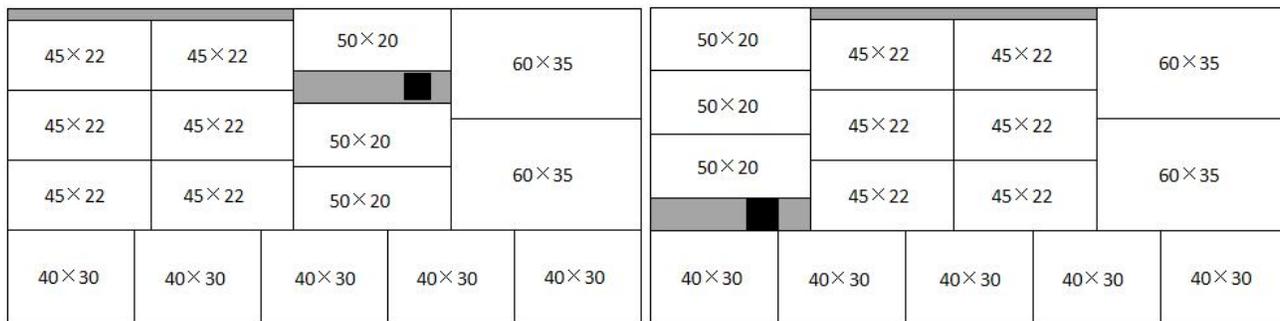
算例 A2



算例 A3

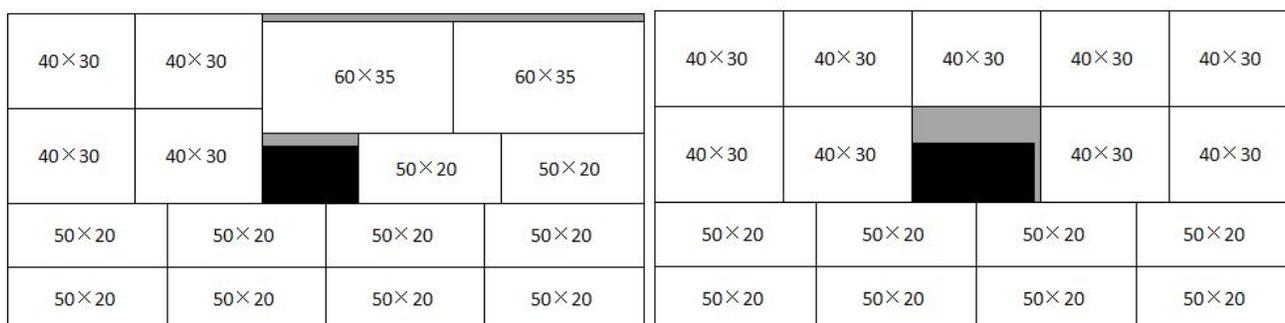


算例 A4



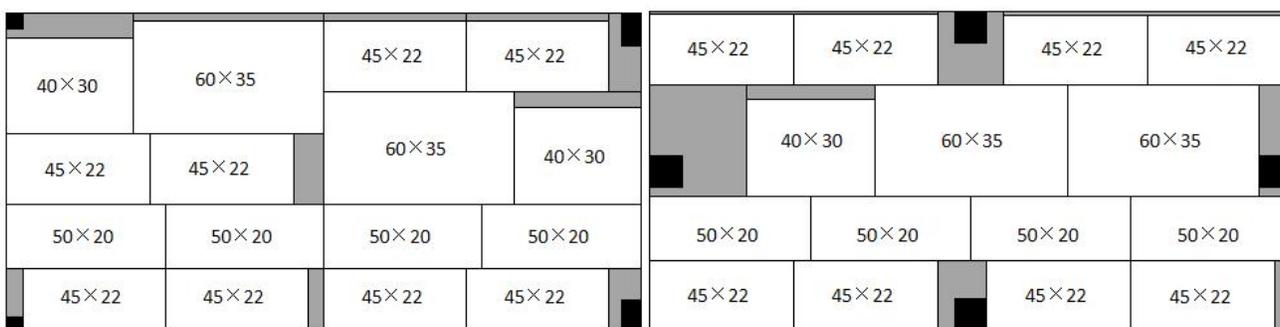
算例 A5

算例 A6



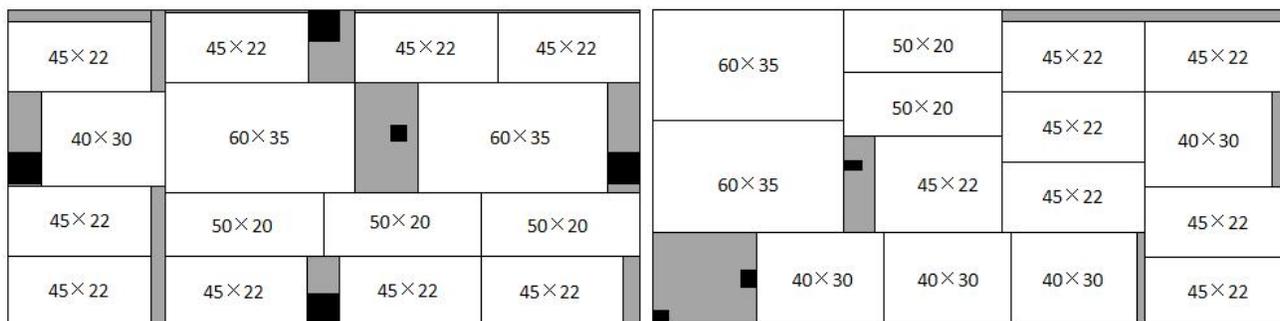
算例 A7

算例 A8



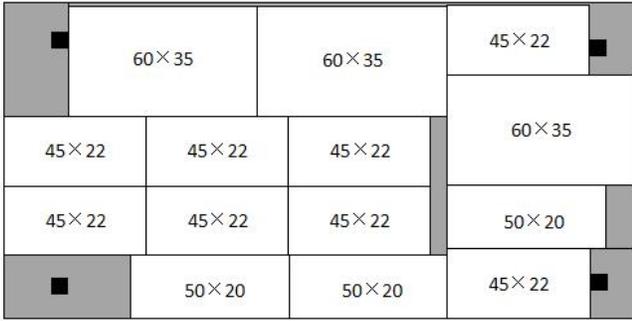
算例 A9

算例 A10

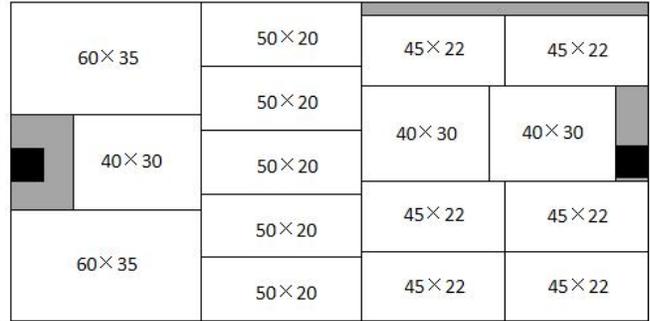


算例 A11

算例 A12



算例 A13



算例 A14