



An Improved Heuristic-Dynamic Programming Algorithm for Rectangular Cutting Problem*

Yin Aihua, Chen Chong, Hu Dongping, Huang Jianghai and
Yang Fan

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

May 2, 2019

求解矩形切割问题的改进启发式动态规划算法*

尹爱华, 陈冲⁺, 胡冬萍, 黄江海, 杨帆

江西财经大学 软件与物联网工程学院, 江西 南昌 330013

+ 通讯作者 E-mail: 1303841543@qq.com, 硕士生

摘要: 讨论了含砂眼的二维切割问题, 从包含多个砂眼的大矩形板材切割出若干给定尺寸和方向的小矩形块, 所切割出的每种小矩形块数量不限, 每个切割都限制为一刀切且切割线不能切到砂眼。另外, 所切割出的小矩形块不能含有砂眼, 目标是最大化切割小矩形块的总价值。为了解决上述问题, 提出一种改进启发式-动态规划算法(Improved Heuristic-Dynamic Programming, IHDP), 并证明了关于其复杂度的重要定理。该算法降低了离散集规模, 用小矩形块的宽和高建立一维背包问题, 利用所得解与砂眼的左右边沿和上下边沿构造两个离散集, 以离散集的每一个元素做试切线进行子问题划分, 放弃穿过砂眼的切割线。该算法计算了国际公认的 14 个典型算例, 实验结果表明, 它得到了这些实例的最优解, 其计算效率比最新文献算法高了近十倍。

关键词: 砂眼; 一刀切; 离散集; 动态规划; NP-难

文献标志码: A **中图分类号:** TP301.5

An Improved Heuristic-Dynamic Programming Algorithm for Rectangular Cutting Problem*

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

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

Abstract: The two-dimensional cutting problem with defects is discussed. A small rectangular block of a given size and direction is cut from a large rectangular plate containing a plurality of defects. The number of each small rectangular block cut is not limited, and each cut is to be of the guillotine and the cutting line cannot be cut to the defects. In addition, the small rectangular blocks that are cut cannot contain defects, and the goal is to maximize the total value of cutting small rectangular blocks. In order to solve the above problems, an improved Heuristic-Dynamic Programming (IHDP) algorithm is proposed to prove the important theorem about its complexity. The algorithm reduces the size of the discrete set, establishes the one-dimensional knapsack problem

*The 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。

with the width and height of the small rectangular block, constructs two discrete sets using the obtained solution and the right and upper boundaries of the defects, and performs trial cut lines for each element of the discrete set. The sub-question is divided, and the cutting line passing through the defect is abandoned. The algorithm calculates 14 internationally accepted examples. The experimental results show that it obtains the optimal solution of these examples, and its computational efficiency is nearly ten times higher than that of the latest literature algorithm.

Key words: defect; guillotine; discrete set; Dynamic Programming; NP-hard

1 引言

带砂眼的二维切割问题在切割问题中一个重要的领域。工业化的工厂中,许多切割问题会遇到砂眼的存在。例如,在家具工业中,木板可能包含破损,被标记区域不能用于切割成家具。在钢铁工业中,生产的钢卷可能含有污染区域,像皮革这样的天然产品通常具有质量较差的孔隙或污染区域,很少有可以切割的部件用于工艺。从以往的研究来看,尽管其重要性显而易见,但是在切割问题相关的文献中,研究矩形大板材的无砂眼^[1-4]的算法较为广泛,偶尔也能找到研究多个数量砂眼问题的存在。从大量问题类型来看,到目前为止只有非常有限的数量研究这类问题。所以这类问题目前越发重要,不少研究者正着手研究这类问题。

大部分作者在无砂眼的二维切割问题上提出了许多算法,例如 Cui YD 等人^[5]的生成 T-形毛坯最优 T 形排样方法的动态规划方法、Wang 等人^[6]提出单规则矩形排样问题的一刀切的启发式搜索算法等研究工作。

Song 等人^[7]提出了一种基于动态规划的启发式算法,该算法生成所有可能的切割模式的子集,他们的算法是一种不完整的方法,其中可能无法获得一些复杂的切割模式。Kang 和 Yoon 等人^[8]提出了该算法的改进版,用于求解单个带砂眼的矩形大板材的切割问题,他们的算法有效地消除了支配模式,避免了重复模式。对于单个矩形大板材切割问题有砂眼解算法的详细综述可以在例如 Neidlein 和 Wäscher 等人^[9-10], Birgin 等人^[11]和 Afsharian 等人得到一些启发。Vianna 和 Arenales 等人^[12]审视了这

个问题,他提供了一种基于 AND / OR 的分支限定算法,他们进一步介绍了一种启发式搜索,结合了深度优先搜索和深度限制以及爬山策略,以加快搜索过程。需要注意的是,作者使用他们的方法来解决多个砂眼的一些实例,他们只给出了目标函数值,没有列出运行时间。Neidlein 和 Wäscher 等人^[9]提出了一种可以在这些算法中使用的改进界限。必须注意的是,Vianna 和 Arenales^[12]以及 Neidlein 和 Wäscher^[9]提出的算法中的离散集规模减少了 Herz 等人^[14]提出的离散集规模。此外,他们的方法涉及若干结构和计算约束,如果处理多个砂眼,而不仅仅是单个砂眼,这些约束就变得明显。

在本文中,提出了一种改进的启发式^[15-18],动态规划结合的方法。在存在若干砂眼的情况下,该方法再次重新考虑了 Afsharian 等人^[19]的离散集规模定义。并给出 14 个例子的最优切割模式。

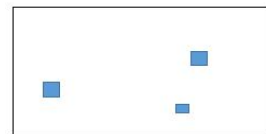


Fig.1 A large plate with multiple defects

图 1 含砂眼的大板材

2 问题描述

记 m 种不同类型的小矩形块 i ($i = 1, \dots, m$), 宽度 w_i , 高度 h_i , 小矩形块的价值 v_i , 必须从宽度为 W_0 且高度为 H_0 的单个矩形大板材切割, 使得由切割过程产生的小矩形块的总值最大化。该问题的解是一个切割模式, 即从大板块所产生的小矩形块的形式以及根据这些小矩形块被布置在大板块上的布局的描述。在这种布局中, 小矩形块必须平行

于大板块排列。对于此处要考虑的问题，应满足以下约束条件：

(1) 切割出的每种小矩形块的数量是不限定的；

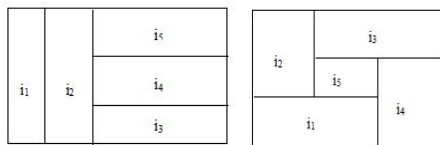
(2) 切割小矩形块时，必须按照给定的宽、高来切割(即切割小矩形块时不允许做 90° 旋转)；

(3) 所有的切割动作必须是一刀切(图 2)，即矩形块的一个边延伸到其相对边，从而产生两个子矩形块；

(4) 切割出的小矩形块不能含有砂眼，否则为废料。

大板材存在 n 个矩形砂眼，其中第 $j(j=1, 2, \dots, n)$ 个砂眼的宽度为 w_j^d 且高度为 h_j^d 。本文求解的切割问题要求所有上述约束都得到满足，即它是一个二维、无约束的、一刀切、无限制并带砂眼的单个矩形大板材切割问题(2D_UG_SLOCP_D)。令 $P = (p_1, p_2, \dots, p_m)$ 为切割模式，其中 $p_i(i=1, 2, \dots, m)$ 表示第 i 种小矩形块的切割个数。则目标是最大化切割小矩形块的价值，那么可以表示如下：

$$\begin{cases} \max V = \sum_{i=1}^m v_i \cdot p_i \\ \text{s.t. } P \text{ 为一个合理的切割模式} \end{cases} \quad (1)$$



(a) A guillotine (b) A Non-guillotine
(a) 一刀切 (b) 非一刀切

Fig.2 Two ways of cutting

图 2 切割的两种方式

3 算法描述

本文所提出的算法，称为改进启发式动态规划(IHDP)算法，DP 算法生成所有可能切割模式的子集，是以离散集的每一个元素做试切线进行子问题划分，并放弃穿过砂眼的切割线。

3.1 基本定义

定义 1 (子块) 一刀切切割大板材，其通过原大板材的切割或原板材经过一系列切割所形成，一刀切总能产生两个子块(图 3)。与没有砂眼的问题

不同，它在每个子块的坐标与尺寸是唯一的。这里对原板材，以左下角为原点， X 轴为板材宽方向， Y 轴为板材高方向建立笛卡尔坐标系，因此，原板材可由 $(0, 0, W_0, H_0)$ 表示。子块由左下角位于大板材上的位置表示其坐标为 (ox, oy) 。设子块的尺寸为 (x, y) ，然后该子块可由 (ox, oy, x, y) 表示。通过该定义在子块具有不同坐标的两个子块 $T_1 = (ox_1, oy_1, x, y)$ 和 $T_2 = (ox_2, oy_2, x, y)$ 是不同的子问题，即使它们有相同的尺寸(图 4)。在切割位置 z_x 处对子块 $T = (ox, oy, x, y)$ 执行垂直切割(平行于 y 轴)之后，两个生成的子块由 (ox, oy, z_x, y) 和 $(ox + z_x, oy, x - z_x, y)$ 。同样地，对于切割位置 z_y (平行于 x 轴)，形成两个子块 (ox, oy, x, z_y) 和 $(ox, oy + z_y, x, y - z_y)$ (图 5)。

定义 2 (砂眼的表示) 砂眼区域其实是不规则的图形，通过矩形或一组整数尺寸的非重叠矩形来近似表示。让砂眼 j 的尺寸为 (w_j^d, h_j^d) ，并让它在在大板块上的位置由 (x_j^d, y_j^d) 表示，然后它可以用 $(x_j^d, y_j^d, w_j^d, h_j^d)$ 表示(图 6)。

定义 3 (C-块与 D-块) 如果子块 $T = (ox, oy, x, y)$ 含有砂眼，则称为 D (Defect)-块；否则它被称为 C (Clean)-块。若 $(x_j^d + w_j^d \leq ox \text{ or } x_j^d \geq ox + x)$ 或 $(y_j^d + h_j^d \leq oy \text{ or } y_j^d \geq oy + y)$ ，则 $T = (ox, oy, x, y)$ 是一个 C -块。反之，则是 D -块，这可以通过以下公式确定：

$$c \& p(ox, oy, x, y) = \begin{cases} 0, & \text{if } (x_j^d + w_j^d \leq ox \text{ or } x_j^d \geq ox + x) \text{ or} \\ & (y_j^d + h_j^d \leq oy \text{ or } y_j^d \geq oy + y) \\ 1, & \text{otherwise} \end{cases} \quad (2)$$

如上式所知，若当前子块为 C -块，则 $c \& p(ox, oy, x, y)$ 的值为 0，否则为 D -块，值为 1。

定义 4 (切割限制) 切割时总要注意切割位置，切割线不能穿到砂眼，这样会造成板材全部浪费，我们在计算的时候，放弃这类切割线(图 7)。

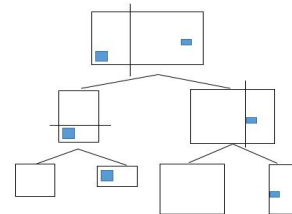


Fig.3 a series of branch blocks formed by guillotine cuts

图 3 原板材一刀切形成一系列子块

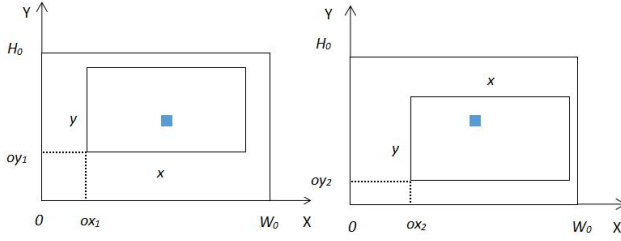


Fig.4 Two branch blocks with the same width and height and different positions

图 4 宽高相同，位置不同的两个子块

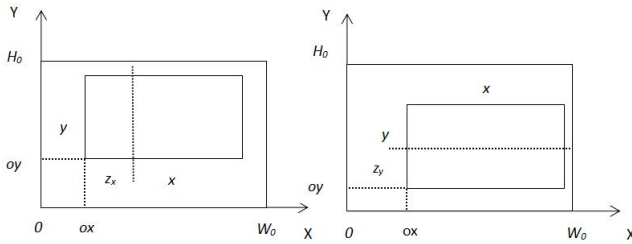


Fig.5 After a guillotine, the branch block gets two new branch blocks

图 5 一刀切后子块得到两个新的子块

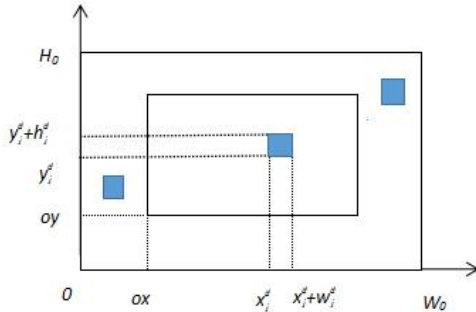


Fig.6 Branch block with defect

图 6 带砂眼的子块

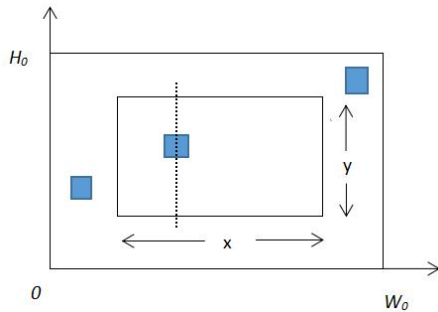


Fig.7 Cutting line through the defect

图 7 切割线穿过砂眼

3.2 离散集

在子块上的切割位置称为离散集。当子块为 C-

块，则采用如下(3)、(4)、(5)公式计算出的离散集^[8]。否则使用如下(6)、(7)、(8)公式所计算出的离散集，所有公式里的 Z_+ 属于正整数集，本文将原板材当做最大的子块。

3.2.1 C-块的离散集

若当前子块为 C-块，则使用下面公式所求的离散集，这与 Afsharian 等人^[6]所提出的离散集并无差异，这些离散集由小矩形的宽和高建立一维背包问题，所得解构成的两个离散集。让 $\Phi_x(x)$ 表示子块的垂直离散集，同样地，让 $\Phi_y(y)$ 表示子块水平离散集，则有如下公式：

$$\Phi_x(x) = \left\{ z_x \mid z_x = \sum_{i=1}^m \alpha_i w_i^s, 1 \leq z_x \leq W_0 - w_0, \right. \\ \left. \alpha_i \in Z_+ \cup \{0\}, \forall i \right\} \quad (3)$$

(垂直方向离散集，垂直切割位置集合)

$$\Phi_y(y) = \left\{ z_y \mid z_y = \sum_{i=1}^m \beta_i h_i^s, 1 \leq z_y \leq H_0 - h_0, \right. \\ \left. \beta_i \in Z_+ \cup \{0\}, \forall i \right\} \quad (4)$$

(水平方向离散集，水平切割位置集合)

$$w_0 = \min \{w_i^s : h_i^s < y\}_1^m, \\ h_0 = \min \{h_i^s : w_i^s < x\}_1^m \quad (5)$$

3.2.2 D-块的离散集

若当前子块为 D-块，则将使用下面公式所求的离散集，这些离散集由小矩形的宽和高建立一维背包问题构成，利用所得解与砂眼的左右边沿与上下边沿构成如下两个离散集，这很大程度上减少了离散集规模 Afsharian 等人^[9]所提出的离散集，从而在都得到最优解的情况下，例子的运行时间有所减少近十倍(下面实验会呈现)：

$$\Phi_x(x) = \left\{ z_x \mid z_x = \sum_{i=1}^m \alpha_i w_i^s + \delta_j (x_j^d - ox) + \delta_j (x_j^d + w_j^d - ox), \right. \\ \left. 1 \leq z_x \leq x - w_0, \alpha_i \in Z_+ \cup \{0\}, \delta_j \in \{0,1\}, \right. \\ \left. \text{if } w_i^s > x \Rightarrow \alpha_i = 0, \text{ if } h_j^d > y \Rightarrow \delta_j = 0, \forall i, \forall j \right\} \quad (6)$$

(垂直方向离散集，垂直切割位置集合)

$$\Phi_y(y) = \left\{ z_y \mid z_y = \sum_{i=1}^m \beta_i h_i^s + \mu_j (y_j^d - oy) + \mu_j (y_j^d - oy), \right. \\ \left. 1 \leq z_y \leq y - h_0, \beta_i \in Z_+ \cup \{0\}, \mu_j \in \{0,1\}, \right. \\ \left. \text{if } w_i^s > x \Rightarrow \beta_i = 0, \text{ if } w_j^d > x \Rightarrow \mu_j = 0, \forall i, \forall j \right\} \quad (7)$$

(水平方向离散集, 水平切割位置集合)

$$\begin{aligned} w_0 &= \min \left(\{w_i^s : h_i^s < y\}_1^m \cup \{w_j^d : h_j^d < y\}_0^l \right), \\ h_0 &= \min \left(\{h_i^s : w_i^s < x\}_1^m \cup \{h_j^d : w_j^d < x\}_0^l \right) \end{aligned} \quad (8)$$

3.3 动态规划

为了确定子块 $T = (ox, oy, x, y)$ 的切割模式的最优值, T 是否存在砂眼是重要的。因此, 本文定义了最优目标的两个递归函数, 一个用于 C -块, 一个用于 D -块。对于 D -块, 必须确定其在大板材上的坐标和尺寸, 相应的计算函数由 $F(ox, oy, x, y)$ 表示。对于 C -板, 仅确定尺寸就足够了, 相应的计算函数将简单地用 $F(x, y)$ 表示。此外, 为了确定 C -块 $T = (x, y)$ 相关的目标函数值的一个可能下界, 用计算函数 $g(x, y)$, 当 $(z_x \geq x/2)$ 或 $(z_y \geq y/2)$, 其被计算所有同一种类型的小矩形的最大值, 其公式如下:

$$g(x, y) = \max \left(0, v_i \cdot \left\lfloor \frac{x}{w_i^s} \right\rfloor \cdot \left\lfloor \frac{y}{h_i^s} \right\rfloor \mid w_i^s > x, h_i^s > y, i=1, 2, \dots, m \right) \quad (9)$$

(1) 当子块为 C -块时, $F_x(x, y)$ 表示垂直方向上的递归函数, 则公式为:

$$F_x(x, y) = F(z_x, y) + F(p(x - z_x), y), z_x \in \Phi_x(x), \downarrow \leq z_x \leq x/2 \quad (10)$$

(2) $F_y(x, y)$ 表示水平方向的递归函数, 则公式为:

$$F_y(x, y) = F(x, z_y) + F(x, q(y - z_y)), z_y \in \Phi_y(y), 1 \leq z_y \leq y/2 \quad (11)$$

即 $F(x, y)$ 的最优值为:

$$\begin{aligned} F(x, y) &= \max(g(x, y), F_x(x, y), F_y(x, y), \\ & x \in \Phi_x(W_0) \cup \{W_0\}, y \in \Phi_y(H_0) \cup \{H_0\}) \end{aligned} \quad (12)$$

其中当 $x < w_0$ 或 $y < h_0$, $F(x, y) = 0$ 时, 当 $x < w_0$ 或 $y < h_0$ 。值得注意的是, 由于重复切割模式的出现, (12) 中的垂直(水平)切口的离散集被限制为分支板的长度(宽度)的一半。以下用于确定带砂眼子块 $T = (ox, oy, x, y)$ 的最优切割模式的递归函数:

(1) $F_x(ox, oy, x, y)$ 表示垂直方向的递归函数, 则公式为:

$$\begin{aligned} F_x(ox, oy, x, y) &= \\ \max \left(\begin{aligned} &F(ox, oy, z_x, y) + F(ox + z_x, oy, p(x - z_x), y), \\ &F(ox, oy, p(x - z_x), y) + F(ox + x - z_x, oy, z_x, y), \end{aligned} \right. \quad (13) \\ & z_x \in \Phi_x(x) \end{aligned}$$

(2) $F_y(ox, oy, x, y)$ 水平方向的递归函数, 则公式

为:

$$\begin{aligned} F_y(ox, oy, x, y) &= \\ \max \left(\begin{aligned} &F(ox, oy, x, z_y) + F(ox, oy + z_y, x, q(y - z_y)), \\ &F(ox, oy, x, q(y - z_y)) + F(ox, oy + y - z_y, x, z_y), \end{aligned} \right. \quad (14) \\ & z_y \in \Phi_y(y) \end{aligned}$$

即 $F(ox, oy, x, y)$ 的最优值为:

$$\begin{aligned} F(ox, oy, x, y) &= \\ \left\{ \begin{aligned} &F(x, y) \quad \text{if } T = (ox, oy, x, y) \text{ 不存在砂眼;} \\ &\max(F_x(ox, oy, x, y), F_y(ox, oy, x, y)), \\ &1 \leq z_x \leq x - w_0, 1 \leq z_y \leq y - h_0, \\ &x \in \Phi_x(W_0) \cup \{W_0\}, y \in \Phi_y(H_0) \cup \{H_0\}, \\ &ox \in \Phi_x(W_0) \cup \{0\}, oy \in \Phi_y(H_0) \cup \{0\}, \\ &\text{if } T = (ox, oy, x, y) \text{ 存在砂眼} \end{aligned} \right. \quad (15) \end{aligned}$$

其中当 $x < w_0$ 或 $y < h_0$, $F(ox, oy, x, y) = 0$ 。必须注意的是, 必须满足上述的切割限制。另外, 为使得切割图形标准化(图 8), 引入上述递归函数中的 $p(x)$, $q(y)$, 计算公式如下:

$$p(x) = \max(0, z_x \mid z_x \leq x, z_x \in \Phi_x(x), x < W_0, p(W_0) = W_0 \quad (16)$$

$$q(y) = \max(0, z_y \mid z_y \leq y, z_y \in \Phi_y(y), y < H_0, q(H_0) = H_0 \quad (17)$$

$p(x)$ 表示最接近子板宽度 x 的切割点, 相应地, $q(y)$ 是最接近子板高度 y 的切割点。



(a) Normalized cutting pattern (b) Non-normalized cutting pattern

(a) 标准切割模式 (b) 非标准切割模式

Fig.8 Normalized cutting pattern(shaded is scrap)

图 8 标准切割模式(阴影部分为废料)

3.4 算法的复杂性

在本节中, 我们研究算法的计算方面。首先, 我们分析其最坏情况的时间复杂性。然后我们将确定决定其运行时间的关键参数。

定理 1 通过上一节提出的算法具有多个砂眼的 $2D_UG_SLOCP$ 解的最坏情况时间复杂度是

$$O(|\Phi_x(W_0)| |\Phi_y(H_0)| (|\Phi_x(W_0)| + |\Phi_y(H_0)|)) \quad (18)$$

证明 对于给定单个大板材 (W_0, H_0) , 使得递归

函数每次迭代需要 $O(|\Phi_x(W_0)||\Phi_y(H_0)|)$ 次操作。因此，计算总共涉及时间复杂度为公式(18)。

定理 2 令 $w_0 = \min(\{w_i^s\}_i^m)$ 和 $h_0 = \min(\{h_i^s\}_i^m)$ 。并令 $\rho_w = \text{floor}(W_0/w_0) \leq W_0/w_0$, $\rho_h = \text{floor}(H_0/h_0) \leq H_0/h_0$, 则有:

$$|\Phi_x(W_0)| = \sum_{t=1}^{\rho_w} C_{t+m-1}^t + 2n \quad (19)$$

$$|\Phi_y(H_0)| = \sum_{t=1}^{\rho_h} C_{t+m-1}^t + 2n \quad (20)$$

$$C_{t+m-1}^t = \frac{(t+m-1)!}{t!(m-1)!} \quad (21)$$

证明 通过定义, $\Phi_x(W_0)$ 中的每个元素是小矩形的长度的可行组合。因此, 计算的问题 $\Phi_x(W_0)$ 具有与 $\sum_{t=1}^{\rho_w} (\sum_{i=1}^m w_i^s)^t$ 计算多项式中的项数的问题相同的结构。为了获得 $\sum_{i=1}^m \alpha_i w_i^s \leq W_0$, 在上述多项式中, t 必须取 1 到 ρ_w 。那是因为如果 $t > \rho_w$, 则有 $\sum_{i=1}^m \alpha_i w_i^s > W_0$ 。而 $\sum_{t=1}^{\rho_w} (\sum_{i=1}^m w_i^s)^t =$

$\sum_{t=1}^{\rho_w} C_{t+m-1}^t$ 。加上砂眼的左右边沿得到上式(19); 同理, (20)式亦是如此。

4 计算结果与分析

本文所用的算法使用 c/c++ 编程语言实现, 所用计算机的配置: 处理器--Intel(R), Core(TM) i7-4790 cpu @ 360HZ, RAM 8Gb, 64bit operator。

在表 1 中, 列出了通过本文的方法获得的目标函数值和计算运行时间并与其他算法做对比, 优越由切割出的小矩形的价值的总和。列出了 Afsharian 等人[6]的实验的结果, 他们没有给出国际

公认的 14 个算例的最优切割模式, 都求出了最优解, 本文的运行效率提升近十倍, 可以看出减少离散集规模很大程度上优化了运行效率, 在本文中给出了 A1 与 A14 两个算例的最优切割截图(图 9-10), (其它算例见附录); 本文算法与 Vianna and Arenales[8]进行对比, 他们只给出了单个砂眼的运行时间, 多个砂眼的运行时间没有给出, 并且他们的若干例子没有得到最优解; Neidlein 等人只给出了单个砂眼的优越与时间, 部分优越没有得到最优解, 多个砂眼的算例没有给出线索。

Plate(200*100) (1)5 piece--40*30 (2)9 piece--45*22
(3)1 piece--50*20 (4)2 piece--60*35

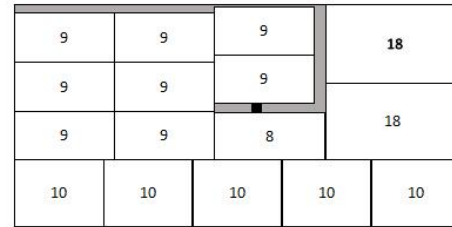


Fig.9 Screenshot of A1 cutting result (shaded part is scrap and black part represents defect)
图 9 A1 切割结果的截图(阴影部分为废料, 黑色部分为砂眼)

Plate(200*100) (1)3 piece--40*30 (2)6 piece--45*22
(3)5 piece--50*20 (4)2 piece--60*35

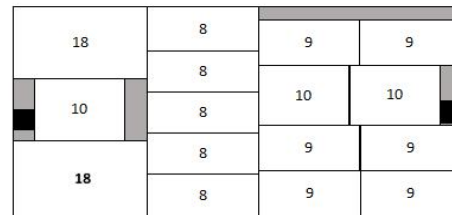


Fig.10 Screenshot of A14 cutting result (shaded part is scrap and black part represents defect)
图 10 A14 切割结果的截图(阴影部分为废料, 黑色部分为砂眼)

Table 1 Operation results and comparison of 4 algorithms
表 1 4 种算法的运行结果与对比

算例	IHDP		Afsharian etc.(2014)		Vianna etc.(2006)		Neidlein etc.(2008)	
	优度	计算时 间(s)	优度	计算时 间(s)	优度	计算时 间(s)	优度	计算时 间(s)
A1	166	0.86	166	18.86	166	4.61	166	0.52
A2	166	0.67	166	16.43	160	3.57	160	0.77
A3	166	0.74	166	16.47	162	4.40	162	1.77
A4	164	0.25	164	18.25	160	3.15	160	0.27
A5	164	0.31	164	76.96	164	13.51	164	4.11
A6	164	0.47	164	0.90	164	1.32	164	1.44
A7	158	0.21	158	0.81	158	12.47	158	1.07
A8	154	0.13	154	1.21	154	8.07	154	0.50
A9	160	0.87	160	14.32	153	-	-	-
A10	158	0.49	158	2.22	148	-	-	-
A11	151	6.29	151	26.78	143	-	-	-
A12	156	119.30	156	1126.44	150	-	-	-
A13	150	2.19	150	9.06	142	-	-	-
A14	160	0.09	160	1.00	160	-	-	-

5 结束语

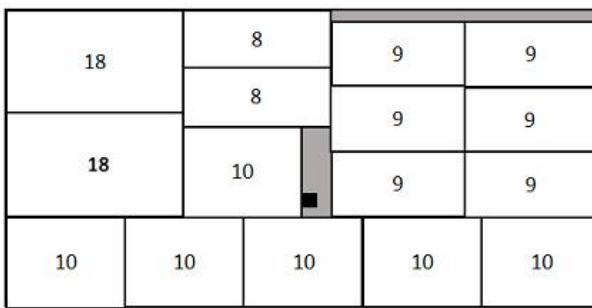
本文采用较小的离散集, 改进了原有算法求解二维带砂眼的切割问题, 提出了一中基于改进型启发式-动态规划的算法, 并证明了并证明了关于其复杂度的重要定理得到中等规模典型算例的最优解, 并提高了计算效率, 也优于最新文献中所发表的最好结果。今后的研究在更多算例上进行, 例如小矩形块的类型个数、大板材的尺寸等, 求解更大规模的算例。

References

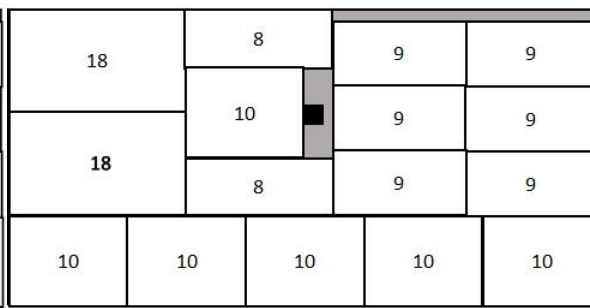
- [1] Beasley, J. E . Algorithms for Unconstrained Two-Dimensional Guillotine Cutting[J]. Journal of the Operational Research Society, 1985, 36(4):297-306
- [2] Liu Y , Chu C , Wang K . A new heuristic algorithm for a class of two-dimensional bin-packing problems[J]. International Journal of Advanced Manufacturing Technology, 2011, 57(9-12):1235-1244.
- [3] Clautiaux F , Jouglet A , Hayek J E . A new lower bound for the non-oriented two-dimensional bin-packing problem[J]. Operations Research Letters, 2007, 35(3):365-373.
- [4] University X , Xiamen, Fujian. A Bricklaying Heuristic Algorithm for the Orthogonal Rectangular Packing Problem[J]. Chinese Journal of Computers, 2008, 31(3):509-515.
- [5] Cui Yaodong. Recursive Algorithm for Generating Optimal T-Shape Cutting Patterns of Rectangular Blanks.Vol.18, No.1, Jan, 2006 (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).
- [6] Lei W , Qiang L , Xin C . Heuristic Search Algorithm for the Rectangular Fixed-Size Guillotine Bin Packing Problem[J]. Journal of Software, 2017,28:1640-1654[doi:10.13328/j.cnki.jos.005100] (王磊, 刘强, 陈新. 单规格一刀切矩形排样问题的启发式搜索算法[J]. 软件学报, 2017(7).
- [7] 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.
- [8] Yoon K , Ahn S , Kang M . An improved best-first branch-and-bound algorithm for constrained two-dimensional guillotine cutting problems[J]. International Journal of Production Research, 2013, 51(6):1680-1693.
- [9] 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.
- [10] Wäscher G, Haubner H, Schumann H. An improved topology of cutting and packing problems[J]. European Journal of Operational Research, 2007, 183(3):1109-1130. [doi:10.1016/j.ejor.2005.12.047]
- [11] Birgin E G , Lobato R D , Morabito R . Generating unconstrained two-dimensional non-guillotine cutting patterns by a Recursive Partitioning Algorithm[J]. Journal of the Operational Research Society, 2012, 63(2):183-200.
- [12] Vianna ACG, Arenales MN. Problema de corte de placas defeituosas. Pesqui Operacional 2006, 26:185-202
- [13] Beasley, J. E . Algorithms for Unconstrained

- Two-Dimensional Guillotine Cutting[J]. Journal of the Operational Research Society, 1985, 36(4):297-306
- [14] Herz, J. C . Recursive Computational Procedure for Two-dimensional Stock Cutting[J]. IBM Journal of Research and Development, 1972, 16(5):462-469.
- [15] Zhang D , Deng A , Kang Y . A Hybrid Heuristic Algorithm for the Rectangular Packing Problem[J]. Lecture Notes in Computer Science, 2005, 3514:783-791.
- [16] Zhang DF, Kang Y, Deng A. A new heuristic recursive algorithm for the strip rectangular packing problem. Computers & Operations Research, 2006, 33(8):2209-2217. [doi:10.1016/j.cor.2005.01.009].
- [17] Berkey J O , Wang P Y . Two-Dimensional Finite Bin-Packing Algorithms[J]. Journal of the Operational Research Society, 1987, 38(5):423-429.
- [18] Alvarez-Valdes, Ramón, Martí, Rafael, Tamarit J M , et al. GRASP and Path Relinking for the Two-Dimensional Two-Stage Cutting-Stock Problem[J]. INFORMS Journal on Computing, 2007, 19(2):261-272.
- [19] 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.

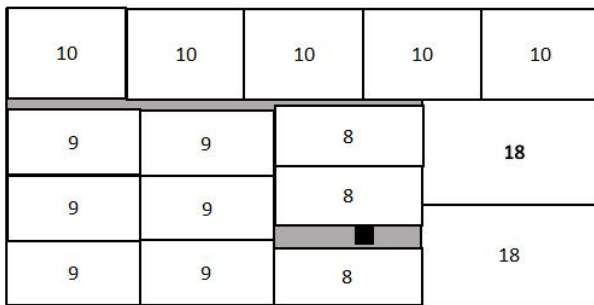
附录



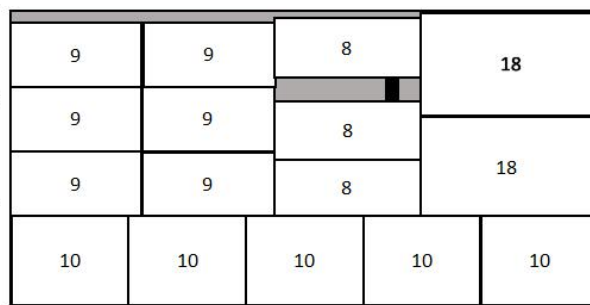
A2 最优切割模式



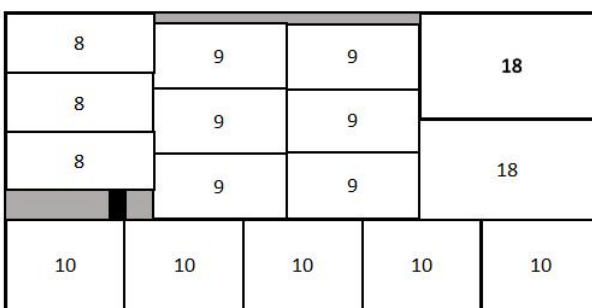
A3 最优切割模式



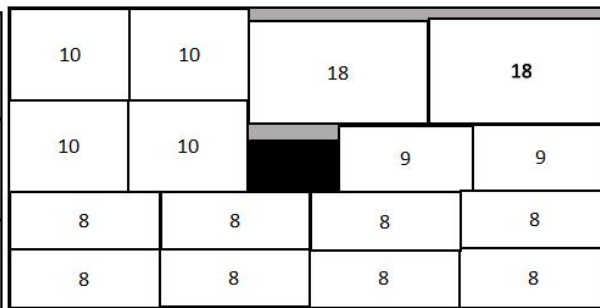
A4 最优切割模式



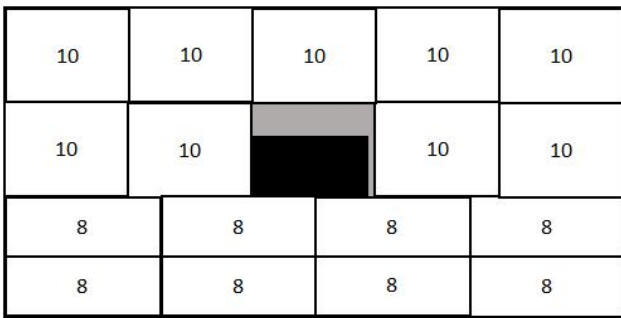
A5 最优切割模式



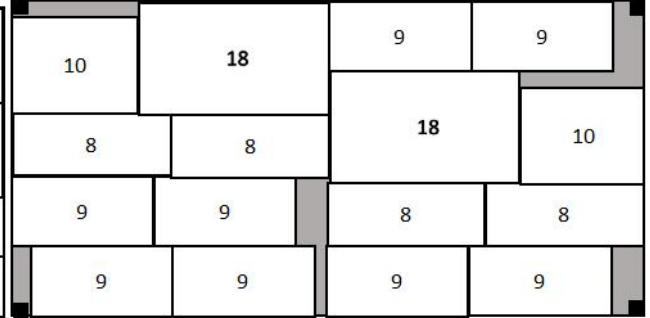
A6 最优切割模式



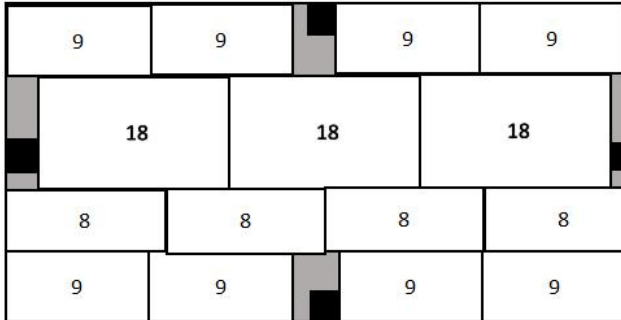
A7 最优切割模式



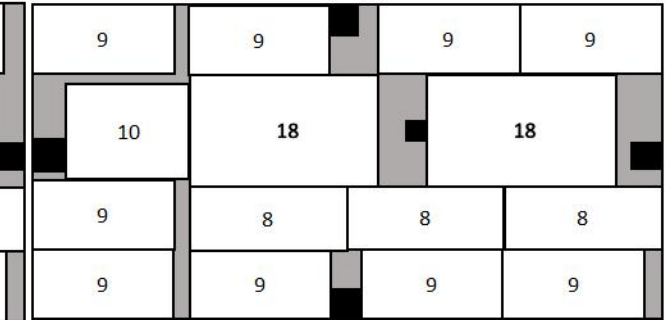
A8 最优切割模式



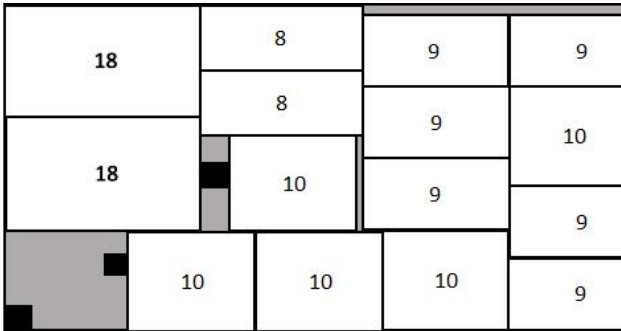
A9 最优切割模式



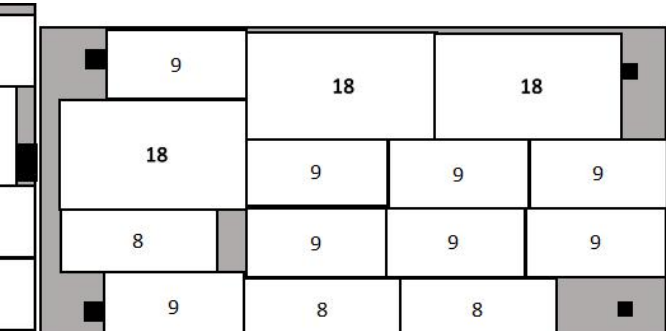
A10 最优切割模式



A11 最优切割模式



A12 最优切割模式



A13 最优切割模式