

## 用于二维不规则排样的离散临界多边形模型<sup>\*</sup>

张德富<sup>1,2+</sup>, 陈竞驰<sup>1</sup>, 刘永凯<sup>1</sup>, 陈火旺<sup>2,3</sup>

<sup>1</sup>(厦门大学 计算机科学系,福建 厦门 361005)

<sup>2</sup>(东南融通博士后工作站,福建 厦门 361005)

<sup>3</sup>(国防科学技术大学 计算机学院,湖南 长沙 410073)

### Discrete No-Fit Polygon, A Simple Structure for the 2-D Irregular Packing Problem

ZHANG De-Fu<sup>1,2+</sup>, CHEN Jing-Chi<sup>1</sup>, LIU Yong-Kai<sup>1</sup>, CHEN Huo-Wang<sup>2,3</sup>

<sup>1</sup>(Department of Computer Science, Xiamen University, Xiamen 361005, China)

<sup>2</sup>(Longtop Group Post-Doctoral Research Center, Xiamen 361005, China)

<sup>3</sup>(School of Computer, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: E-mail: dfzhang@xmu.edu.cn, http://59.77.16.229/

**Zhang DF, Chen JC, Liu YK, Chen HW. Discrete no-fit polygon, a simple structure for the 2-D irregular packing problem. *Journal of Software*, 2009,20(6):1511–1520. <http://www.jos.org.cn/1000-9825/3331.htm>**

**Abstract:** This paper presents a model based on discrete no-fit polygon for the two-dimensional irregular packing problem. Burke et al. have presented an effective BLF algorithm to solve the irregular packing problem, however, their algorithm might generate invalid results for some special cases. To solve this problem, a model based on discrete no-fit polygon is proposed, and its correctness has been strictly proved. Only points and intervals are only considered by this model, which greatly decreases the geometry complexity of the original problem and makes the problem easily solved by many heuristic strategies. Computational results show that the algorithm based on discrete no-fit polygon model is very efficient.

**Key words:** irregular packing problem; infeasible interval; no-fit polygon; discrete no-fit polygon

**摘要:** 提出了一个用于求解二维不规则排样问题的离散临界多边形模型。Burke 等人的 BLF 算法是求解排样问题的一种有效算法,但其算法对一些特殊实例会产生非法的解。为了解决这个问题,提出了一种基于离散临界多边形模型,并对其正确性作了严格证明。新模型是只含有点和区间的简单模型,在大大降低原问题几何复杂性的同时,也使许多启发式策略可以更容易地求解该问题。计算结果表明,基于离散临界多边形模型的排样算法是有很有效的。

**关键词:** 不规则排样问题;禁止区;临界多边形;离散临界多边形

中图法分类号: TP301 文献标识码: A

\* Supported by the National Natural Science Foundation of China under Grant No.60773126 (国家自然科学基金); the Natural Science Foundation of Fujian Province of China under Grant No.A0710023 (福建省自然科学基金); the 985 Information Technology Fund of Xiamen University of China under Grant No.0000-X07204 (厦门大学 985 二期信息科技基金); the Academician Start-Up Fund of Xiamen University of China under Grant No.X01109 (厦门大学院士启动基金)

Received 2008-01-22; Accepted 2008-03-31

二维不规则排样问题(two-dimensional irregular packing problem)是指将一系列二维平面图形(或零件)互不交叠地放置在某一面板上,使得未被覆盖的面板面积最小.排样问题在皮革、木材、金属等制造业的下料问题中有着广泛的应用.当所有零件均为矩形时,二维排样问题已经有比较有效的算法<sup>[1-3]</sup>.对于不规则图形的情况,简单地将求解矩形排样问题的算法修改成求解不规则排样问题的算法通常不能得到相应的效果.几何复杂度的提高对算法的效果产生了质的影响<sup>[4]</sup>:矩形排样问题的潜在可行放置位置的分布本质上是离散的,而不规则排样问题则是连续的.

求解不规则排样问题涉及的几何运算主要在于判定不同几何形体间的位置关系.最简单的方法是将不规则图形用简单规则多边形(通常为矩形)包围,把复杂图形位置关系的判定转化成包围后简单图形位置关系的判定,这种方法通常会造成较大的浪费.现有的处理方法主要有两种:一种是将几何形体用离散的像素点近似表示<sup>[5-9]</sup>,通过检测各个像素点的位置判断不同几何形体间是否相交.这种方法程序实现难度小,但是对于非正交的斜边无法精确表示,进而影响求解质量.另一种方法则基于临界多边形(no-fit polygon,简称NFP)<sup>[10-12]</sup>,通过构造两个多边形的临界多边形,将两个多边形位置关系的判定转化成点和多边形位置关系的判定.这种方法的求解精度高,但是算法复杂<sup>[13]</sup>,运行时间长.另外,一些特殊的数据,例如含有孔洞的多边形或者两个多边形恰好拼接在一起的情形,会给临界多边形的求解带来很大的困难,甚至造成错误<sup>[4]</sup>.文献[14]则采用了一种随机求解算法,即首先产生一个初始解(多边形允许重叠),然后通过一个高效的子过程,结合导向式局部搜索(guided local search,简称GLS)不断减少重叠面积,直至得到一个合法解,取得了很好的结果.本文提出了离散临界多边形模型,利用该模型可以降低求解原问题的几何复杂性,从而方便许多启发式策略对其进行高效求解.

## 1 临界多边形的概念及性质

### 1.1 多边形的定义

首先定义本文涉及的一些基本概念.

通常采用多边形来近似表示一个平面图形.平面上的点定义为一个二元组 $p=(p_x, p_y) \in R^2$ ,这里采用文献[14]中对边、多边形以及多边形内部的定义:

**定义 1(边).** 一条边 $e$ 通过其两个端点 $e_a, e_b$ 进行定义,用公式表示为 $e(t)=e_a+t(e_b-e_a)(t \in [0, 1])$ .给定点 $p$ 和边 $e$ ,记 $p \in e$ 当且仅当 $p=e(t)(t \in [0, 1])$ .

**定义 2(多边形).** 一个多边形 $P$ 通过边集 $E_P$ 来定义,这些边必须形成一个或多个环,并且任何两条边不相交.多边形的内部定义为一个点集:

$$\tilde{P} = \{p \in R^2 \mid \bar{f}_{E_P}(p) \equiv 1 \pmod{2}\},$$

其中,

$$\bar{f}_{E_P}(p) = |\{e \in E_P \mid \exists x' < p_x \text{ 且 } (x', p_y) \in e\}|.$$

定义2中, $\bar{f}_{E_P}(p)$ 的含义可以理解为:过点 $p$ 平行于 $x$ 轴作一条指向 $x$ 轴负方向的射线,射线与多边形 $P$ 的边集 $E_P$ 的交点个数(不包括与 $p$ 重合的点)就是 $\bar{f}_{E_P}(p)$ 的值.

由于仅考虑对多边形作平移操作,因此,放置一个多边形只要确定该多边形上任意一个顶点的坐标即可,这个顶点称为多边形 $P$ 的参考点,记为 $O_P$ .本文统一将 $P$ 的左下顶点作为其参考点,即取 $O_P \in e(e \in E_P)$ ,满足

$$\forall e \in E_P, \forall p \in e, p_x > O_{P_x} \vee (p_x = O_{P_x} \wedge p_y \geq O_{P_y}).$$

给定多边形 $A$ 与多边形 $B$ ,我们称多边形 $A$ 与多边形 $B$ 相交当且仅当 $\tilde{A} \cap \tilde{B} \neq \emptyset$ ,多边形 $A$ 与多边形 $B$ 相切当且仅当 $\tilde{A} \cap \tilde{B} = \emptyset$ 且存在点 $p$ 使得 $p \in e(e \in E_A)$ 且 $p \in e'(e' \in E_B)$ ,其余情况称为多边形 $A$ 与多边形 $B$ 相离.

### 1.2 临界多边形的概念及性质

利用临界多边形,我们可以很方便地判断两个多边形是否相交.给定多边形 $A$ 与多边形 $B$ ,其临界多边形可以通过下述方式构造:选定多边形 $B$ 上的一个点 $O_B$ 作为参考点,固定多边形 $A$ ,将多边形 $B$ 沿着多边形 $A$ 平移一周,保

证多边形  $B$  和多边形  $A$  始终相切,此时,参考点的移动轨迹所围成的区域(不包括边界)就是多边形  $B$  相对于多边形  $A$  的临界多边形,记为  $NFP_{AB}$ ,其边集记为  $E_{AB}$ .

临界多边形有如下性质:当移动多边形  $B$  使得  $O_B \in NFP_{AB}$  时,  $B$  与  $A$  相交;当移动多边形  $B$  使得  $O_B$  在  $NFP_{AB}$  的边界上,即  $O_B \in e (e \in E_{AB})$  时,  $B$  与  $A$  相切;其他情况下,  $B$  与  $A$  相离.

临界多边形已被证明与两个多边形的 Minkowski 和有密切联系.

给定任意两个点集  $\tilde{A}$  和  $\tilde{B}$ , 其 Minkowski 和定义为

$$\tilde{A} \oplus \tilde{B} = \{a + b \mid a \in \tilde{A}, b \in \tilde{B}\}.$$

下面我们给出一个有用的结论,并给出其简单的证明.

**定理 1.**  $NFP_{AB} = \tilde{A} \oplus -\tilde{B}$ . 其中,  $-\tilde{B} = \{p \in R^2 \mid O_B - p \in \tilde{B}\}$ .

证明:假设将多边形  $B$  从位置  $O_B$  平移到位置  $O'_B$ .

$\forall O'_B \in NFP_{AB}$ , 根据临界多边形的性质,存在顶点  $p'$  满足  $p' \in \tilde{A} \wedge p' \in \tilde{B} \oplus \{O'_B - O_B\}$ . 设  $p'$  在移动前的坐标为  $p$ , 则有  $O'_B = p' + (O_B - p)$ , 因此  $O'_B \in \tilde{A} \oplus -\tilde{B}$ .

反之,若  $O'_B \in \tilde{A} \oplus -\tilde{B}$ , 则存在点  $a$  和点  $b$ , 满足  $a \in \tilde{A}, b \in \tilde{B}$  且  $O'_B = a + (O_B - b)$ , 此时顶点  $b$  的坐标为  $b' = b + O'_B - O_B = a \in \tilde{A}$ , 即多边形  $A$  和多边形  $B$  相交, 根据临界多边形的性质有  $O'_B \in NFP_{AB}$ . □

## 2 Burke 等人的算法及其不足

Burke 等人在文献[15]中描述了一种新的 BLF 算法.不同于像素法在两个方向均作离散化(如图 1(a)所示),新算法仅在水平方向离散化(如图 1(b)所示),其实验结果显示,新算法的求解质量和求解速度相对于传统方法均有显著提高.

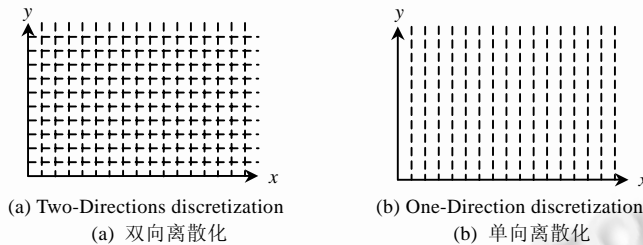


Fig.1 Discretization

图 1 离散化

尽管 Burke 等人的模型对大部分数据表现良好,但是算法本身仍存在不足.如图 2(a)所示,假定图形  $A$  固定,通过调整  $B$  来解决已有冲突.按照 Burke 等人的算法,经过若干次迭代后得到如图 2(b)所示的图形.由于图形  $A$  和图形  $B$  的任意两条边并不相交(对于交点恰好落在边端点上的情况也视为两条边不相交),且  $A$  和  $B$  的任一顶点并不包含在另一图形当中,算法判定图形  $A$  和图形  $B$  不相交,对图形  $B$  的调整结束.然而此时,图形  $A$  和图形  $B$  仍然重叠,从而得到错误的排样结果.

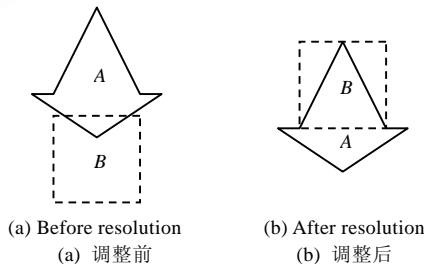


Fig.2 A counter example

图 2 一个反例

### 3 禁止区

事实上,由于放置一个图形时其水平方向的选择是有限的,因此许多工作可以预先完成.具体来说,如图 3(a)所示,假设图形A固定,图形B沿直线 $l:x=t$ 平移,当移动B使得点 $O_B$ 位于图中暗色区间内时,图形B与图形A相交.显然,如果两图形在水平方向上的相对位置固定,则这些区间也是确定的(如图 3(b)所示).我们将图上这些暗色区间称为禁止区.

禁止区与临界多边形有着紧密联系.如图 3(c)所示,图形A和图形B所对应的临界多边形为 $NFP_{AB}$ (图中灰色曲线所围区域).我们将直线 $l:x=k\mu$ (其中, $k$ 为整数, $\mu$ 为每个切片的宽度)与 $NFP_{AB}$ 相交的区域用暗色粗线标记出来.假设 $O_B$ 沿着某一直线 $l:x=k\mu$ 移动,根据临界多边形的性质,当点 $O_B$ 移动到这些暗色区间上时,图形B与图形A相交;当移动到这些区间的端点时,图形B与图形A相切;否则,图形B与图形A相离.

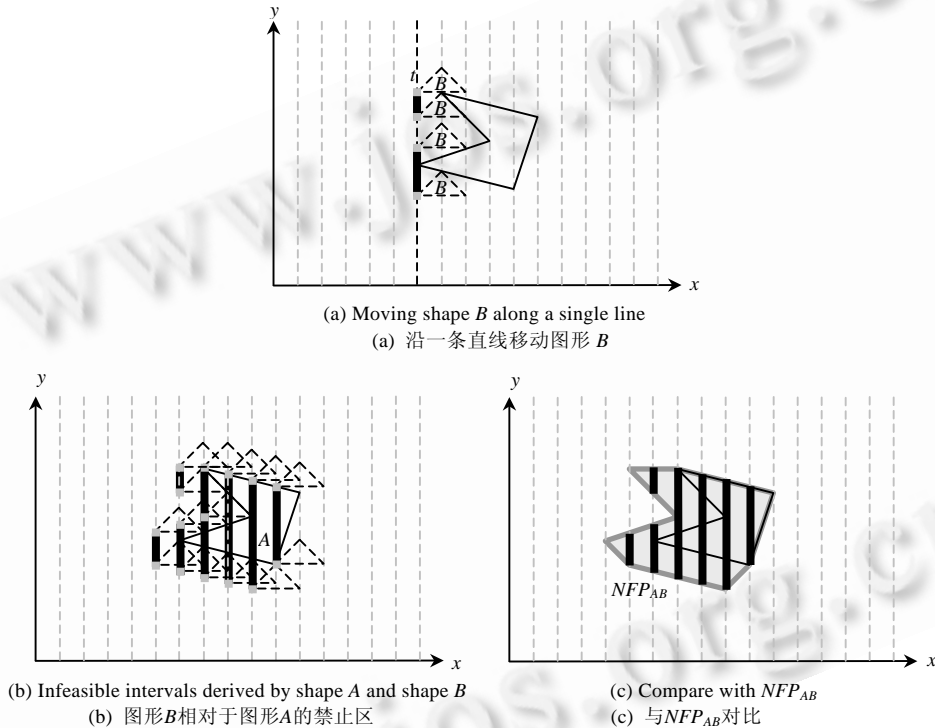


Fig.3 Relationship between infeasible interval and discrete no-fit polygon

图3 禁止区与临界多边形的关系

一般地,我们将禁止区定义为直线 $l:x=t$ 与临界多边形 $NFP_{AB}$ 相交的区间,形式化描述如下:

定义 3(禁止区). 给定多边形A,B的临界多边形 $NFP_{AB}$ ,和实数 $t$ ,禁止区定义为

$$R_{AB}(t) = \{p \in R^2 \mid p \in NFP_{AB} \wedge p_x = t\}.$$

### 4 离散临界多边形的含义及其构造算法

#### 4.1 离散临界多边形

如前所述,临界多边形的求解过程相当繁琐.我们只需求出直线  $l:x=k\mu$ 与临界多边形相交的区间而不需要构造整个临界多边形,换句话说,只需要计算一个离散化的临界多边形——离散临界多边形.

定义 4(离散临界多边形). 给定实数  $t$ 和实数 $\mu$ ,其中 $\mu > 0$ ,多边形 A 和多边形 B 的离散临界多边形定义为

$$DNFP_{AB} = \bigcup_{t \bmod \mu=0} R_{AB}(t).$$

接下来具体描述离散临界多边形的构造算法.首先给出两个定理及其证明,以保证算法的正确性.给定多边形  $A$  和  $B$ ,不妨设多边形  $A$  的内部  $\tilde{A}$  由  $n$  个多边形(的内部)  $\tilde{A}_1, \tilde{A}_2, \dots, \tilde{A}_n$  组成,即  $\tilde{A} = \bigcup_{1 \leq i \leq n} \tilde{A}_i$ ,类似地,  $\tilde{B} = \bigcup_{1 \leq j \leq m} \tilde{B}_j$ ,我们有:

**定理 2.** 设  $\tilde{A} = \bigcup_{1 \leq i \leq n} \tilde{A}_i, \tilde{B} = \bigcup_{1 \leq j \leq m} \tilde{B}_j$ , 则  $NFP_{AB} = \bigcup_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} NFP_{A_i B_j}$ .

证明:由定理 1,

$$NFP_{AB} = \tilde{A} \oplus -\tilde{B} = \left( \bigcup_{1 \leq i \leq n} \tilde{A}_i \right) \oplus - \left( \bigcup_{1 \leq j \leq m} \tilde{B}_j \right) = \bigcup_{1 \leq i \leq n} \left( \tilde{A}_i \oplus \left( \bigcup_{1 \leq j \leq m} -\tilde{B}_j \right) \right) = \bigcup_{1 \leq i \leq n} \bigcup_{1 \leq j \leq m} (\tilde{A}_i \oplus -\tilde{B}_j) = \bigcup_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} NFP_{A_i B_j}. \quad \square$$

**定理 3.**  $DNFP_{AB} = \bigcup_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} DNFP_{A_i B_j}$ .

证明:由定理 2,

$$\begin{aligned} R_{AB}(t) &= \{p \in R^2 \mid p \in NFP_{AB} \wedge p_x = t\} = \{p \in R^2 \mid p \in \bigcup_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} NFP_{A_i B_j} \wedge p_x = t\} = \bigcup_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} \{p \in R^2 \mid p \in NFP_{A_i B_j} \wedge p_x = t\}, \\ DNFP_{AB} &= \bigcup_{t \bmod \mu=0} R_{AB}(t) = \bigcup_{t \bmod \mu=0} \bigcup_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} \{p \in R^2 \mid p \in NFP_{A_i B_j} \wedge p_x = t\} \\ &= \bigcup_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} \bigcup_{t \bmod \mu=0} \{p \in R^2 \mid p \in NFP_{A_i B_j} \wedge p_x = t\} = \bigcup_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} DNFP_{A_i B_j}. \quad \square \end{aligned}$$

利用定理 2 和定理 3 的结论,不难得到离散临界多边形的构造算法.

## 4.2 离散临界多边形的构造算法

### 4.2.1 算法描述

给定多边形  $A$  和  $B$ ,其离散临界多边形  $DNFP_{AB}$  的构造算法如下:

**算法 1.**  $DNFP_{AB}$  构造算法.

开始

步骤 1. 计算  $-B$ ;

步骤 2. 分别对多边形  $A$  和多边形  $-B$  作梯形剖分,将  $A$  分成  $n$  个梯形  $A_i (i=1, 2, \dots, n)$ , 将  $-B$  分成  $m$  个梯形  $-B_j (j=1, 2, \dots, m)$ , 即  $A = \bigcup_{1 \leq i \leq n} A_i, -B = \bigcup_{1 \leq j \leq m} -B_j$  (其中  $A_i, -B_j$  为梯形);

步骤 3. 对每个  $A_i$  和每个  $-B_j$  计算  $DNFP_{A_i B_j}$ ;

步骤 4.  $DNFP_{AB} = \bigcup_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} DNFP_{A_i B_j}$ ;

结束.

接下来解释算法的各个步骤.步骤 2 可以采用任何一种梯形剖分算法(如扫描线算法),复杂度为  $O(n \lg n)$ ,其中,  $n$  为多边形的顶点个数.步骤 3 的主要难点在于计算  $A_i$  和  $B_j$  的临界多边形  $NFP_{A_i B_j}$ , 假设  $NFP_{A_i B_j}$  的最左顶点和最右顶点的  $x$  坐标值为  $\alpha, \beta$ , 以  $\mu$  为单位离散化, 则  $DNFP_{A_i B_j}$  可以通过计算  $NFP_{A_i B_j}$  与直线  $l: x = k\mu (\lfloor \alpha/\mu \rfloor \leq k \leq \lfloor \beta/\mu \rfloor)$  求交得到. 由于  $A_i$  和  $B_j$  均为梯形, 因此, 我们可以利用求给定两个凸多边形的临界多边形的简单算法<sup>[4]</sup>: 将多边形  $A$  的边定为逆时针走向, 多边形  $B$  定为顺时针走向, 然后平移所有这些向量, 使得向量的起始点重合在一个点上, 最后, 将所有向量按逆时针序首尾相接排列得到一个多边形, 该多边形即为  $A$  和  $B$  的临界多边形. 步骤 4 中主要涉及的操作为区间合并, 可以用许多方法来实现. 假设要合并  $t$  个区间, 一种简单的做法是, 先对  $t$  个区间按其首端点排序, 然后通过一遍扫描在线形时间内完成合并操作, 总的时间复杂度为

$O(t \lg t)$ .

4.2.2 算法效率

这里,分析算法 1 的总时间耗费.设多边形  $P$  共有  $n_P$  个顶点,其最左顶点和最右顶点的  $x$  坐标差值记为  $\Delta_P$ ,令  $\Delta = \Delta_A + \Delta_B$ .

步骤 1 的时间耗费为  $O(n_B)$ .

步骤 2 的时间耗费为  $O(n_A \lg n_A + n_B \lg n_B)$ .

步骤 3 中,对于给定的梯形  $A_i$  和  $B_j$ ,计算  $NFP_{A_i B_j}$  的时间可视为常数  $C_1$ ,计算一条直线  $l$  与  $NFP_{A_i B_j}$  相交的时间也可视为常数  $C_2$ .在最坏情形下,最多要计算  $\lfloor (\Delta_A + \Delta_B) / \mu \rfloor + 1$  条直线与  $NFP_{A_i B_j}$  的交.由于  $n \leq n_A, m \leq n_B$ ,步骤 3 的总时间耗费为  $nm(C_1 + C_2(1 + \lfloor (\Delta_A + \Delta_B) / \mu \rfloor)) = O(n_A n_B \Delta / \mu)$ .

步骤 4 中一共要在  $\lfloor \Delta / \mu \rfloor + 1$  个区域内作区间合并操作,每个区域最多有  $nm$  个区间,总时间耗费最多为  $nm \lg(nm) \cdot (\lfloor \Delta / \mu \rfloor + 1) = O\left(\left(\frac{n_A n_B \Delta}{\mu}\right) \lg(n_A n_B)\right)$ .

因此,算法 1 的总时间复杂度为

$$O(n_B) + O(n_A \lg n_A + n_B \lg n_B) + O(n_A n_B \Delta / \mu) + O\left(\left(\frac{n_A n_B \Delta}{\mu}\right) \lg(n_A n_B)\right) = O\left(\left(\frac{n_A n_B \Delta}{\mu}\right) \lg(n_A n_B)\right).$$

事实上,正如后面实验结果所显示,由于这个界比较宽松,实际运行效率将会更高.

4.2.3 修正

算法 1 构造的离散临界多边形与前述定义不完全一致,需要作一些修正.如图 4(a)所示,固定多边形  $A$  和多边形  $B$  沿直线  $l$  移动,算法 1 计算出的禁止区为  $(a,c)$ ,而  $B$  在区间  $(b,c)$  上移动时并不与多边形  $A$  相交.这个差别并不影响放置算法的正确性,但是可能会影响到求解质量.如图 4(b)所示,固定多边形  $C$ ,多边形  $D$  沿直线  $l$  移动,算法 1 计算出的禁止区为  $(d,f)$ ,而当  $D$  位于点  $e$  时,两个多边形并不会相交.由此可见,这个细微的差别可能会丢失一些很好的解.

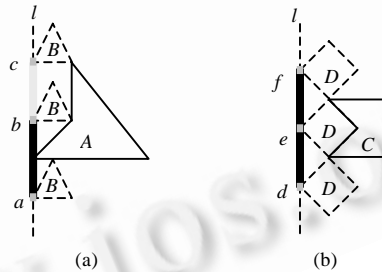


Fig.4 Two special cases

图 4 两个特例

只要对算法 1 略作修改就可以解决这个问题.首先,这里所有区间均视为开区间,这样即使两个区间的首尾端点重合时也不进行合并操作.如图 5 所示,算法 1 的步骤 3 中,已知两个凸多边形  $A_i, B_j$  的临界多边形  $NFP_{A_i B_j}$  计算  $DNFP_{A_i B_j}$  时,对左边界上的禁止区加上“+”号,对右边界上的禁止区加上“-”号,中间的禁止区则兼有“+”号和“-”号(图 5(c)).构造  $DNFP_{AB}$  时,所有的禁止区只保留兼有“+”号和“-”号的部分(图 5(d)).步骤 4 作区间合并时,可以用一个变量记录当前处理的区间的符号,只保留兼有“+”,“-”两种符号的区间,这样,处理区间合并的复杂度仍然是  $O(t \lg t)$  ( $t$  为区间个数).

修正后,算法 1 的时间复杂度仍然为  $O\left(\left(\frac{n_A n_B \Delta}{\mu}\right) \lg(n_A n_B)\right)$ .

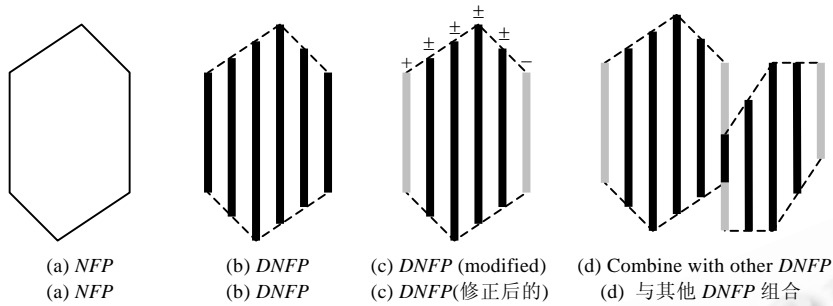


Fig.5 Modification of algorithm 1 (an example)

图 5 对算法 1 的修正(一个实例)

4.3 基于离散临界多边形的放置算法

我们以一个具体的放置为例,简单说明基于离散临界多边形的放置算法.假设置多边形  $P$ (参考点为左下点)时,面板上已经根据某种策略放了若干个图形.如图 6(a)所示,黑色区间表示  $P$  的禁止区,当  $P$  的参考点移动到这些位置时,将由于  $P$  的某一部分与面板上的多边形相交或者  $P$  的某一部分超出面板而成为不合法的放置.一般地,我们考虑将禁止区的端点作为可选择点(图上的灰色点).可以设计一个评价函数对这些点进行选择,例如,每次均选择最左下角的可行点(图中的点  $x$ ),则对应于 Burke 等人的 BLF 算法.图 6(b)、图 6(c)显示的是采用 BLF 放置策略将  $P$  放置在点  $x$  后的情形.

利用离散临界多边形的概念,我们事实上将包含复杂几何运算的原始模型转换成只含有点和区间运算的简单模型,极大地降低了原问题的几何复杂性,从而使许多启发策略可以更容易地求解这个问题.

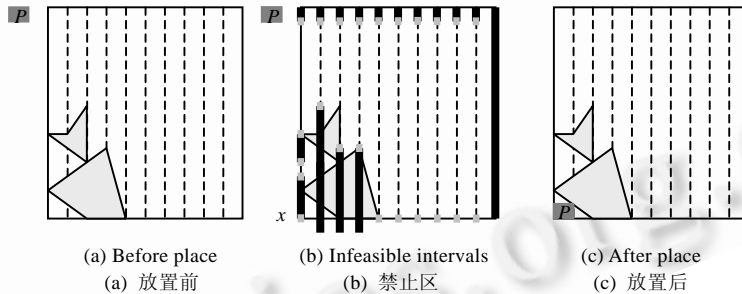


Fig.6 Placement of a shape  $P$

图 6 图形  $P$  的放置

最后,我们考察一下基于离散临界多边形的放置算法对图 2 所举反例的计算情况.图 7(a)中的黑色区间表示图形  $B$  相对于图形  $A$  的禁止区,图 7(b)显示的是将  $B$  放置在禁止区上端点后的情况.此时图形  $B$  与图形  $A$  不相交,算法得到一个正确的放置.

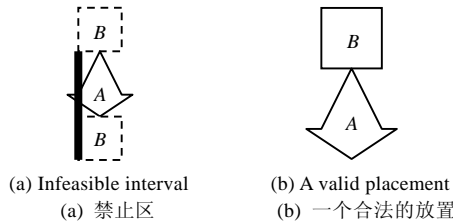


Fig.7 Placement based on DNFP

图 7 基于离散临界多边形的放置

## 5 实验与结果

我们在新模型中实现了Burke等人的BLF算法(简记为离散临界多边形模型+BLF),并与Burke等人的基于单向离散模型的BLF算法(简记为单向离散模型+BLF)<sup>[15,16]</sup>从求解质量和求解速度两个方面作了比较.测试数据包括了文献[15]所列出的除poly和trousers(文献[15]中,这两组数据与我们所得到的数据有出入,因此本文暂不考虑)以外的所有数据,这些数据可以在ESICUP的站点<http://paginas.fe.up.pt/~esicup>上找到.求解结果见表 1.表 1 中,单向离散模型+BLF的计算结果直接取自文献[15],这些实验数据是在P4 2GHz 256MB RAM的环境下得到的.离散临界多边形模型+BLF的计算结果则是在P4 1.6GHz 320MB RAM的环境下得到的.与单向离散模型+BLF算法相同,离散临界多边形模型+BLF也采用简单的爬山搜索(hill climbing)作为局部搜索算法,每次搜索作 100 次迭代,邻域大小为 5,算法作 40 次运行,选择求解结果最好的一组数据(所有这些参数及数据选择方式均与文献[15]所列一致).

**Table 1** Computational results for two algorithms

表 1 两种算法的实验结果对比

| Problem | One-Way discretization model+BLF <sup>[15]</sup> |                          |                | DNFP model+BLF   |                          |                |                       |
|---------|--|--------------------------|----------------|------------------|--------------------------|----------------|-----------------------|
|         | Filling rate (%)                                 | Time for per-packing (s) | Final time (s) | Filling rate (%) | Time for per-packing (s) | Final time (s) | Initializing time (s) |
| Albano  | 84.6   | 1.18                     | 93.39          | <b>84.8</b>      | 0.05                     | 4.92           | 2.36                  |
| Blaz1   | <b>77.7</b>                                      | 0.32                     | 21.00          | 77.1             | 0.01                     | 0.60           | 0.08                  |
| Blaz2   | <b>74.5</b>                                      | 0.16                     | 10.94          | 72.4             | 0.01                     | 0.39           | 0.14                  |
| Dagli   | 83.7   | 2.04                     | 188.80         | <b>84.0</b>      | 0.06                     | 4.33           | 1.63                  |
| Dighe1  | 77.4   | 0.22                     | 8.87           | <b>80.7</b>      | 0.08                     | 6.56           | 9.88                  |
| Dighe2  | 79.4   | 0.10                     | 7.12           | <b>80.7</b>      | 0.03                     | 2.53           | 4.91                  |
| Fu      | <b>86.9</b>                                      | 0.24                     | 20.78          | 86.4             | 0.02                     | 1.40           | 0.80                  |
| Jakobs1 | <b>82.6</b>                                      | 0.74                     | 43.49          | 81.7             | 0.04                     | 0.64           | 2.31                  |
| Jakobs2 | <b>74.8</b>                                      | 2.13                     | 81.41          | 74.2             | 0.08                     | 2.21           | 5.02                  |
| Mao     | 79.5   | 0.38                     | 29.74          | <b>80.0</b>      | 0.06                     | 2.45           | 9.72                  |
| Marques | 86.5   | 0.25                     | 4.87           | 86.5             | 0.06                     | 1.87           | 2.17                  |
| Shapes  | 67.6   | 0.60                     | 31.36          | <b>70.0</b>      | 0.04                     | 3.13           | 0.20                  |
| Shapes0 | 60.5   | 0.93                     | 21.33          | <b>62.3</b>      | 0.02                     | 1.10           | 0.01                  |
| Shapes1 | 66.5   | 0.82                     | 2.19           | <b>68.8</b>      | 0.03                     | 1.58           | 0.05                  |
| Shirts  | 84.6   | 4.99                     | 58.36          | <b>85.7</b>      | 0.15                     | 8.76           | 0.17                  |
| Swim    | <b>68.4</b>                                      | 12.39                    | 607.37         | 68.1             | 0.18                     | 10.87          | 7.08                  |
| Average | 77.2   | 1.72                     | 76.94          | <b>77.7</b>      | 0.06                     | 3.33           | 2.91                  |

表中第 2 列~第 4 列为单向离散模型+BLF(one-way discretization model+BLF)在文献[15]中所列出的填充率(filling rate)、每次排样用时(time for per-packing)、得到最终解用时(final time).第 5 列~第 7 列为离散临界多边形模型+BLF(DNFP model+BLF)算法相应的求解结果及运行时间,最后一列为模型初始化构造离散临界多边形所用时间(Initializing time),时间单位为秒(s).文献[15]中列出了两种方法计算填充率,后一种计算方法得到的填充率高于前者.表 1 中的填充率均采用第 1 种方法进行计算,即

$$\text{填充率} = \frac{\text{排样图形的总面积}}{\text{固定宽度矩形面板的面积}} \times 100\% .$$

所有数据以 $\mu$ 为单位离散化, $\mu$ 值的计算公式如下:

$$\mu = \begin{cases} 1.0, & L < 100 \\ L/100, & L \geq 100 \end{cases}$$

其中,

$$L = \frac{\text{排样图形的总面积}}{\text{矩形面板的宽度}} .$$

从表 1 可以看出,离散临界多边形模型+BLF 有 9 个数据可以获得更高的填充率(图 8 显示的是其中两个排样结果),而单向离散模型+BLF 有 6 个数据获得更高的填充率.在找到最终解的平均速度上,离散临界多边形模型+BLF 比单向离散模型+BLF 快了 20 倍以上.可见,离散临界多边形模型+BLF 算法在求解质量和求解速度上



均优于单向离散模型+BLF.这里需要指出的是,离散临界多边形模型+BLF 算法只需要很少的时间用于初始化构造离散临界多边形.当多次运行程序时,离散临界多边形只需要初始化 1 次.因此,当程序重复运行次数较多时,初始化时间几乎可以忽略.

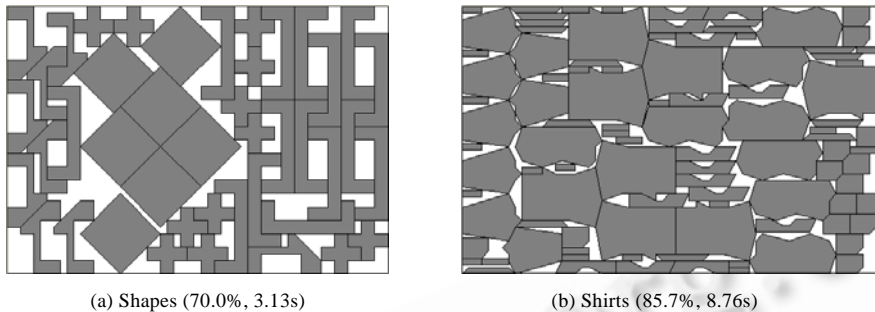


Fig.8 Nesting results for Shapes and Shirts

图 8 对 Shapes 和 Shirts 的排样结果

## 6 结束语

Burke 等人的研究表明,单向离散化模型是求解排样问题的一个有效模型,但是其算法仍存在一些不足.针对这个问题,我们提出了禁止区和离散临界多边形的概念,给出了离散临界多边形的构造算法,并对其正确性作了严格证明.对 16 个测试数据的计算结果表明,离散临界多边形模型+BLF 算法不仅在求解质量上优于单向离散模型+BLF,而且速度更快.此外,正如文中所述,基于离散临界多边形的放置算法还可以实现除 BLF 以外的其他放置策略.因此,我们今后的研究重点将在该模型的基础上,深入研究各种启发搜索策略,推广禁止区的概念,以处理图形中含有弧线的情况.

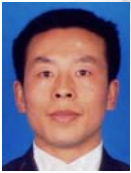
## References:

- [1] Cui YD, He DL, Song XX. Generating optimal two-section cutting patterns for rectangular blanks. *Computers and Operations Research*, 2006,33(6):1505–1520.
- [2] Huang WQ, Chen DB, Xu RC. A new heuristic algorithm for rectangle packing. *Computers and Operations Research*, 2007,34(11): 3270–3280.
- [3] Zhang DF, Han SH, Ye WG. A bricklaying heuristic algorithm for the orthogonal rectangular packing problem. *Chinese Journal of Computers*, 2008,31(3):509–515 (in Chinese with English abstract).
- [4] Bennell JA, Oliveira JF. The geometry of nesting problems: A tutorial. *European Journal of Operational Research*, 2008,184(2): 397–415.
- [5] Babu AR, Babu NR. A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms. *Computer-Aided Design*, 2001,33:879–891.
- [6] Bouganis A, Shanahan M. A vision-based intelligent system for packing 2-D irregular shapes. *IEEE Trans. on Automation Science and Engineering*, 2007,4(3):382–394.
- [7] Zhang YP, Zhang CL, Jiang SW. An effective approach for leather nesting. *Journal of Software*, 2005,16(2):316–323 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/16/316.htm>
- [8] Heckmann R, Lengauer T. A simulated annealing approach to the nesting problem in the textile manufacturing industry. *Annals of Operations Research*, 1995,57:103–133.
- [9] Fischer AD, Dagli CH. Employing subgroup evolution for irregular-shape nesting. *Journal of Intelligent Manufacturing*, 2004,15: 187–199.
- [10] Gomes AM, Oliveira JF. A 2-exchange heuristic for nesting problems. *European Journal of Operational Research*, 2002,141(2): 359–370.

- [11] Gomes AM, Oliveira JF. Solving irregular strip packing problems by hybridising simulated annealing and linear programming. *European Journal of Operational Research*, 2006,171(3):811–829.
- [12] Liu HY, He YJ. Algorithm for 2-D irregular-shaped nesting problem based on the NFP algorithm and lowest-gravity-center principle. *Journal of Zhejiang University—Science A*, 2006,7(4):570–576.
- [13] Bennell JA, Song X. A comprehensive and robust procedure for obtaining the nofit polygon using Minkowski sums. *Computers and Operations Research*, 2008,35:267–281.
- [14] Egeblad J, Nielsen BK, Odgaard A. Fast neighborhood search for two- and three-dimensional nesting problems. *European Journal of Operational Research*, 2007,183(3):1249–1266.
- [15] Burke E, Hellier R, Kendall G, Whitwell G. A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem. *Operations Research*, 2006,54(3):587–601.
- [16] Burke E, Hellier R, Kendall G, Whitwell G. Complete and robust no-fit polygon generation for the irregular stock cutting problem. *European Journal of Operational Research*, 2007,179(1):27–49.

#### 附中文参考文献:

- [3] 张德富,韩水华,叶卫国.求解矩形 Packing 问题的砌墙式启发式算法. *计算机学报*,2008,31(3):509–515.
- [7] 张玉萍,张春丽,蒋寿伟.皮料优化排样的有效方法. *软件学报*,2005,16(2):316–323. <http://www.jos.org.cn/1000-9825/16/316.htm>



张德富(1971—),男,湖南宁远人,博士,副教授,CCF 高级会员,主要研究领域为计算智能,数据挖掘.



刘永凯(1982—),男,硕士,主要研究领域为计算智能.



陈竞驰(1984—),男,硕士,主要研究领域为计算智能.



陈火旺(1936—2008),男,教授,博士生导师,主要研究领域为计算机软件及理论.