

板材排样中非拟合多边形的构造实现方法^①



毛良献^{1,2}, 王 品^{2,3}

¹(中国科学院大学, 北京 100049)

²(中国科学院 沈阳计算技术研究所, 沈阳 110168)

³(沈阳高精数控智能技术股份有限公司, 沈阳 110168)

通讯作者: 毛良献, E-mail: 1029193287@qq.com

摘 要: 非拟合多边形可用于处理两维的不规则形状的板材排样问题. 先前, 基于非拟合多边形的构造很难实现, 并且也没有通用的方式来处理多种特殊情况, 从而非拟合多边形并没有被广泛的使用. 本文介绍了一种基于环绕的实现方式来构造非拟合多边形, 对各种特殊情况能统一解决, 例如互锁, 交叉等. 通过对 ESICUP 的数据集测试, 表明该方法具有一定的效果, 可以对板材排样的解决思路提供一定的借鉴.

关键词: 板材排样; 非拟合多边形; 环绕方法

引用格式: 毛良献, 王品. 板材排样中非拟合多边形的构造实现方法. 计算机系统应用, 2019, 28(9): 168-173. <http://www.c-s-a.org.cn/1003-3254/7061.html>

Construction and Implementation of No-Fit Polygon for Packing and Cutting Problem

MAO Liang-Xian^{1,2}, WANG Pin^{2,3}

¹(University of Chinese Academy of Sciences, Beijing 100049, China)

²(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

³(Shenyang Golding NC & Intelligence Tech. Co. Ltd., Shenyang 110168, China)

Abstract: The No-Fit Polygon (NFP) can be used for handling of stock packing and cutting problems with two-dimensional non-regular shape. Previously, NFP has not been widely applied because it is difficult to be implemented and is lack of generic approaches that can cope with all problem cases without specific case-by-case handling. This paper introduces an orbital method. The method can handle the typical degenerate cases, such as holes, interlocking concavities. And we make benchmark for ESICUP datasets which from the literature, proving that this approach can be efficient with almost every situation. It has certain reference significance for the research of the packing and cutting problem.

Key words: packing and cutting; no-fit polygon; orbital approach

板材排样问题影响着许多领域, 例如纺织品、塑料、金属切削等等, 其中最主要的就是两维不规则形状的排样. 这些问题通常可以描述为: 将一系列的不规则的二维形状放置在一个或多个板子上, 使得他们尽可能的接触, 这样就可以节约板材的面积, 提高板材利用率. 这种板材排样是一个 NP 难问题, 因此有许多方法用于解决它, 包括线性规划方法, 启

发式放置方法等等^[1-4]. 通常, 绝大多数情况都可以利用三角函数^[5]来解决, 但 NFP 提供了一种高效的解决策略.

在本文中, 介绍了非拟多边形及实现它一种方法. 非拟合多边形结构能处理一些传统方法 (基于三角函数) 很难解决的特殊情况, 说明非拟合多边形在排样处理中可以作为一种解决思路.

① 基金项目: 国家重大科技专项 (2018ZX04035001)

Foundation item: National Science and Technology Major Program (2018ZX04035001)

收稿时间: 2019-03-01; 修改时间: 2019-03-29; 采用时间: 2019-04-01; csa 在线出版时间: 2019-09-05

1 非拟合多边形

本文提出了一种非拟合多边形 (No-Fit Polygon, NFP) 的构造方式, 原理是利用环绕的方法来产生 NFP^[6], 该算法是对文献[6]提出的算法的一种改进实现. 比起传统的利用三角函数的重叠测试^[7-10], 实现该算法简单, 同时具有时间复杂度较低的优点.

算法的原理如下, 输入参数是两个二维多边形, 形状可以是不规则的. 假定其中一个多边形 (记为 A) 保持不动, 另一个多边形 (记为 B) 绕 A 运动. 这样当 B 环绕 A 一周后, 就可以生成一个非拟合多边形 (下面简称为 NFP).

假定多边形 A 为固定多边形, B 为环绕多边形. 算法实现过程分为 4 步. (1) 首先需要 A 与 B 接触; (2) 移动 B; (3) 找到下一个起点; (4) 将平移向量合并.

1.1 多边形 A 与 B 的接触

在 B 绕 A 运动之前, 需要把 B 放置在 A 的边上, 保证 B 接触 A, 但又和 A 不相交. 然后, 可以对 B 沿某方向平移, 使之一直是和 A 保持接触.

例如, 图 1 给出的多边形 A 和 B, 同时标记了 A 的最低点 A_minY 和 B 的最高点 B_maxY.

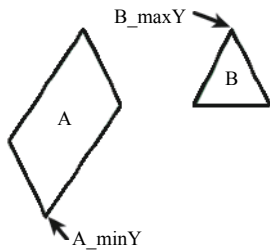


图 1 多边形 A 和 B

那么, 可以让 B 的最高点与 A 的最低点接触, 这样 A 与 B 是一定接触, 但不相交 (也可以有其他方法, 例如 B 最左点和 A 的最右点接触, 等等). 平移向量

$$\vec{T} = A_minY - B_maxY \quad (1)$$

那么 B 按 T 平移后, 如图 2 所示, B 与 A 接触并且不交叉. 此时, B 所在位置是 B 绕 A 运动的开始位置, 并记录起点和 B 的参考点, 保证 B 在绕 A 平移过程中可以回到初始位置. 设 B 的参考点和起点都为 A_minY. 那么, 当 B 的参考点再次平移到起点 (A_minY) 时, 就可以确定 B 绕 A 一周.

1.2 移动 B

在 A 与 B 接触后, 需要对 B 做平移操作. 这个过

程细分为以下 5 步.

1.2.1 找出接触边对

当前 A 和 B 接触, 通过遍历 A 和 B 的全部边, 找出所有接触点. 继而确定全部接触边对.

如图 3 所示, 找到 4 组接触边对, 分别是 <a1, b1>, <a1, b3>, <a2, b1>, <a2, b3>.

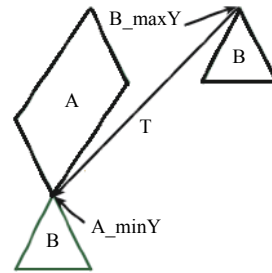


图 2 多边形 B 沿 T 向量平移

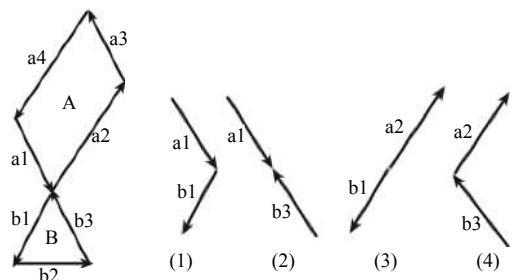


图 3 接触点及接触边对

1.2.2 从接触边对中找到平移向量

在 1.2.1 节中, 找到 4 组接触边对. 接触边对的作用是: 一组接触边对能生成一个平移向量, 故能得到全部的平移向量. 例如, 在图 3 的 4 组边对中, 1 号边对 <a1, b1> 可以生成一个平移向量 $\vec{T} = -\vec{b1}$, \vec{T} 是由边 b1 生成的. 那为什么是由边 b1 生成, 而非 a1 生成? 算法假定平移向量是接触到边的终点, 故 a1 生成的是空向量. \vec{T} 是 $\vec{b1}$ 的反向, 这是认为 B 必须绕 A 做逆时针运动. 那么 2 号边对 <a1, b3> 就不能生成平移向量. 因为接触点是 a1 (也是 b3) 的终点. 而 3 号边对 <a2, b1> (<a2 与 b1 是平行关系) 可以生成平移向量 $\vec{T} = \vec{a2}$ (或者 $\vec{T} = -\vec{b1}$). 也就是说, 在这种情况下可以选择任何一边来生成平移向量. 4 号边对 <a2, b3> 中, $\vec{T} = \vec{a2}$. 接触边对对应的平移向量见表 1.

1.2.3 删除不可行的平移向量

对 1.2.2 节中找到的所有平移向量, 需要依次判断多边形 A 与 B 这些向量移动会不会交叉. 若有交叉,

那么该向量就是不可行的. 按如下方式思考: 因为 B 绕 A 运动, 那么 1.2.1 节中的接触边对中来自 B 的边是运动的. 也就是说, 我们可以对每一个平移向量, 依

次针对每一组接触边对, 判断来自 B 的边按该向量平移, 是否与来自 A 的边相交. 若有一组接触边相交, 那么就可以判断该平移向量是不可行的, 直接舍弃.

表 1 多边形 A 和 B 的接触边对对应的平移向量

接触点		移动边 (来自 B) 相对固定边 (来自 A) 的位置	平移向量
固定边 a(来自 A)	移动边 b(来自 B)		
起点	起点	左边	$-\vec{b}$ (\vec{b} 取反)
起点	起点	右边	\vec{a}
起点	终点	左边	无
起点	终点	右边	\vec{a}
终点	起点	左边	无
终点	起点	$-\vec{b}$	$-\vec{b}$
终点	终点	任意	无
终点	终点	平行	\vec{a} 或 $-\vec{b}$

那么怎么验证接触边对来自 B 的边按平移向量平移, 会不会与来自 A 的接触边交叉. 对于每一组接触边对而言, 可以计算出弧度区间, 使得来自 B 的边沿区间中的某个弧度方向平移, 不会和来自 A 的边相交. 故只需要判断平移向量的弧度方向是否在该区间内即可. 例如, 对 1.2.1 节中的四组接触边对, 可以确定相应的弧度区间 (用圆弧表示可行的弧度区间, 如图 4 所示).

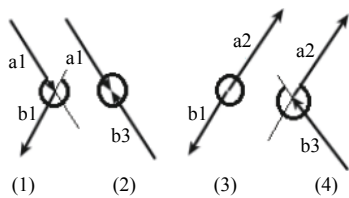


图 4 接触边对的可行弧度区间

在 1.2.2 中, 找到的平移向量有 $-\vec{b}_1$ 和 \vec{a}_2 . 对照 4 组接触边对的弧度区间和 $-\vec{b}_1$ (或 \vec{a}_2) 的弧度方向, 发现 $-\vec{b}_1$ (或 \vec{a}_2) 的弧度方向均在每一组的弧度区间内, 故 $-\vec{b}_1$ 和 \vec{a}_2 都是可行的平移向量.

1.2.4 修剪可行的平移向量

通过 1.2.3 的过滤, 找到了全部可行的平移向量. 那么接下来, 需要判断每一个可行的平移向量是否能全部应用.

在图 5 中, A 是一个不规则多边形, B 是一个矩形. 红色的是一个可行的平移向量, 浅绿色是红色的平移向量平移到 B 的每个顶点之后的情况.

此时 B 沿着红色的平移向量平移, A 与 B 必然交叉. 从而, 必须对红色向量进行修剪. 图中得到修剪后

的是绿色的平移向量. 看到有两个绿色向量, 但取得的是较短的那个. 同时, 需要注意的是, B 按修剪后的平移向量平移, 一定不会与 A 相交?

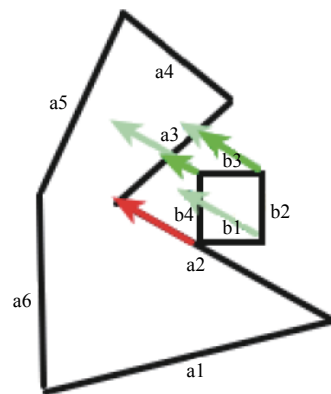


图 5 多边形沿平移向量平移, 交叉

在图 6 中, A 不变, B 是较大的矩形, 红色向量仍是可行的平移向量. 将红色向量平移到 B 的每个顶点, 发现并不需要修剪平移向量. 但是按红色向量平移后, 如图 7 所示, A 与 B 相交. 这种情况下, 就需要考虑将可行的平移向量平移到 A 的每个顶点, 然后再修剪. 如图 8 所示.

在图 8 中, 将平移向量的每个终点平移到 A 的每个顶点, 判断与 B 的交叉性, 最终得到修剪后的平移向量 (绿色). 然后将两种情况的最小值作为最终的可行平移向量. 对每一个可行的平移向量执行上述过程之后, 就能得到全部修剪后的平移向量, 然后按从长到短排序. 接下来只需要按最长的修剪后的可行平移向量平移 B 就可以了.

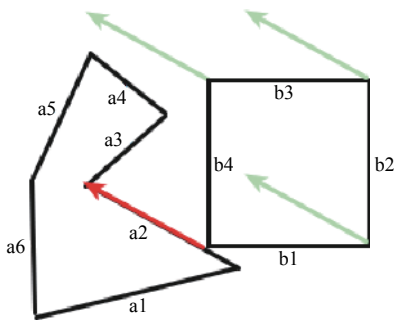


图6 平移向量的起点放置在B的每个顶点

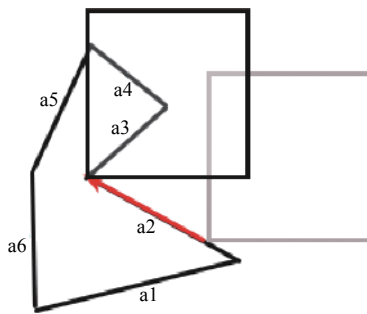


图7 多边形B沿平移向量平移

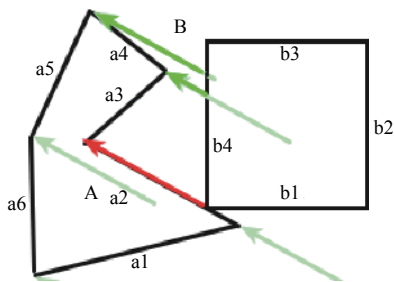


图8 平移向量的修剪

值得指出的是,在文献[5]中,直接使用修剪后的最长平移向量进行平移.但是,在实验中发现该平移向量不一定是能用的.也就是说,按此平移向量平移后,B在下一次平移之前就不一定与A接触了.

在图9中,找到了两个可行的平移向量 $\vec{p1}$ 和 $\vec{p2}$, $\vec{p2}$ 的长度比 $\vec{p1}$ 的长,并且 $\vec{p1}$ 和 $\vec{p2}$ 均是不需要修剪的.当B按 $\vec{p2}$ 平移之后,A与B就不在接触,那么在下一轮找接触点时,就不可能找到,也就无法继续执行.因此,在这种情况下,需要抛弃 $\vec{p2}$,选择 $\vec{p1}$.

1.2.5 移动多边形B

通过前面的步骤,得到了修剪后最长的可行平移向量.接下来,B按这个向量平移即可.但需要注意两点:(1)判断B的参考点是否回到了起点.若回到起点,

得到一个NFP.(2)判断B的参考点是否越过起点.这个是对文献[5]的一个补充.也就是此时B的平移距离过长,超过了起点,如图10所示.

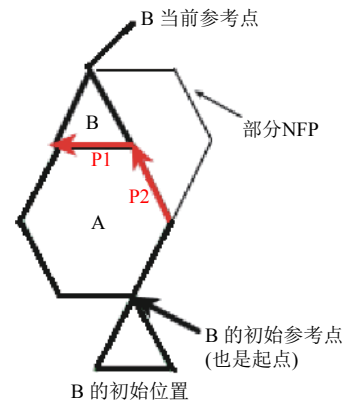


图9 B沿平移向量 $\vec{p1}$ (或 $\vec{p2}$)平移情况分析

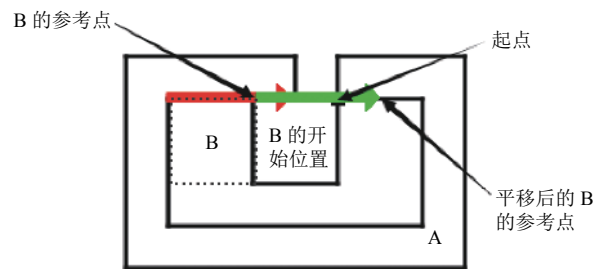


图10 B的参考点沿平移向量平移越过起点

在图10中,红色的是可行的平移向量,绿色的表示B的参考点的平移.由图知,当B的参考点沿平移向量平移后,越过了起点.本来可以生成一个NFP,现在就会无限循环下去,因为B的参考点不会移东到起点了.在这种情况下,需要缩短平移向量.

1.3 找到下一个起点

在1.2节中,生成了一个完整的外部NFP,是B对A的外部环绕.是否存在其他NFP呢?也就是B在A的内部平移存不存在,从而生成内部的NFP.这里涉及到起点搜索.在1.2节中,我们假定的起点是A的最低点.而在这里,我们需要找到其他起点,以便B能从这些起点出发,得到其他的NFP.

1.3.1 起点搜索算法

在寻找外部NFP过程中,A中的有些边可能没有遍历到.故可以从这些边中寻找起点.假设a是A中一条未遍历的边,试着在a中找到一个可行的起点.算法过程如下:依次让B的每个顶点平移到a的起点,判

断: (1) 如果 A 此时与 B 没有相交, 那么 a 的起点是一个可行的起点. 可再次运用 1.2 节中介绍的方法. (2) 如果 A 此时与 B 有交叉, 那么需要让 B 沿着 a 移动, 直到平移到一个的不相交位置, 或者到达 a 的终点 (这就说明在 a 上不可能找到起点).

现在, 假定 A 与 B 相交, 然后 B 沿 a 平移寻找可行的起点. 首先, 可以快速判断一下, B 沿着 a 移动是否一定存在交叉. 方法如下: 因为此时的接触点是 a 的起点, 并且 B 中有两条边也接触到这个顶点. 那只要判断一下, B 的两条接触边是否至少有一条边在 a 的左侧. 若成立, 那么 B 沿着 a 移动一定会与 A 相交. 这样, 就需要判断 A 中其他未遍历的边了. 如果通过上述判断, 就需要修剪 a 向量. 过程如下: 当前的平移向量 $\vec{T} = \vec{a}$ 的起点 \rightarrow a 的终点. 同样, 利用 1.2.4 中介绍的修剪方法修剪 \vec{T} , 得到 \vec{T}' . 那么 B 按 \vec{T}' 平移, 同时更新接触点和 B 的参考点. 再次运用 1.2 节中介绍的方法, 找出剩余的平移向量. 这里要注意, 就是对 a 进行标记, 表示 a 被遍历过了. 那么在下一轮的搜索中就不会遍历边 a 了.

1.4 平移向量合并生成 NFP

通过 1.1 节到 1.3 节的计算, 找到所需要全部平移向量. 现在, 就需要对他们合并生成 NFP. 每一组平移向量都对应一个起点和 B 的参考点. 那么对 B 的参考点执行一组平移操作, 就可以生成一个 NFP.

2 案例介绍

下面举一些例子, 如图 11 至图 15, 表明算法的运行情况. 图 11 至图 15 中左图是多边形的初始位置, 右图是平移的开始位置、起点和生成的 NFP.

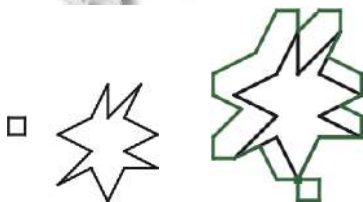


图 11 生成一个外部的 NFP

3 测试

根据文献[5]和来自 the Association of the European

Operational Research Societies 的板材排样工作小组的数据集 (<https://www.euro-online.org/websites/esicup/data-sets/>), 执行如表 2 所列出的测试. 实验中使用的机器为 Intel Core i5-3230M@2.6 GHz, 8 GB 内存. 表格的形式参考了文献[5].

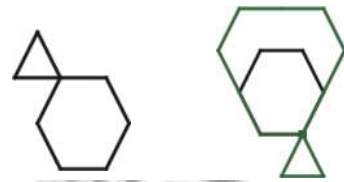


图 12 生成一个外部的 NFP

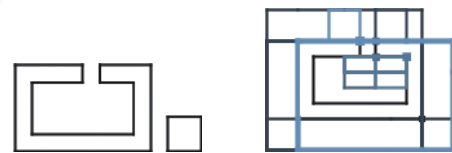


图 13 生成内部和外部的 NFP

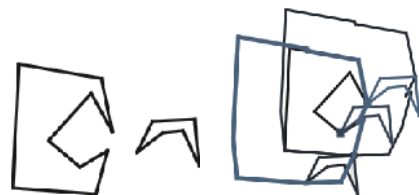


图 14 生成内部和外部的 NFP



图 15 互锁和外部的 NFP

4 结论与展望

生成了完整的 NFP 结构, 给出算法实现的具体细节和一些要点. 通过测试, 表明该方法具有一定的高效性. 与先前的一些借助于三角函数的方法相比, 该算法的实现简单, 高效, 且不需要对每一种特殊情况做特殊的考虑. 利用起点搜索过程, 对某些传统方法无法解决的互锁, 洞等情况, 能够成功解决. 对板材排样系统的设计与实现有一定的借鉴意义.

表2 对不同数据集的测试结果

A	B	C	D	E	F	G	H
数据集	不同形状数	旋转度	有旋转生成的形状数	总共的逻辑形状数	NFP 个数	生成时间 (s)	每秒的 NFP 数
Terashima1	540	0	1	540	291 600	386.67	754
Terashima2	480	0	1	480	230 400	320.00	720
Albano180	8	180	2	16	256	0.21	1219
Albano90	8	90	4	32	1024	0.53	1932
Dagli	10	90	4	40	1600	0.70	2286
Dighe1	16	90	4	64	4096	1.06	3864
Dighe2	10	90	4	40	1600	0.45	3556
Fu	12	90	4	48	2304	0.15	15 360
Jakobs1	25	90	4	100	10 000	3.52	2841
Jakobs2	25	90	4	100	10 000	3.47	2882
Mao	9	90	4	36	1296	1.32	982
Marques	8	90	4	32	1024	0.65	1575
Poly1a	15	90	4	60	3600	1.00	3600
Poly2a	15	90	4	60	3600	0.99	3636
Poly3a	15	90	4	60	3600	0.8	4500
Poly4a	15	90	4	60	3600	1.09	3303
Poly5a	15	90	4	60	14 400	1.02	3529
Poly2b	30	90	4	120	32 400	5.64	2553
Poly3b	45	90	4	180	57 600	20.50	1580
Poly4b	60	90	4	240	90 000	55.48	1038
Poly5b	75	90	4	300	256	125.32	718
Shapes	4	90	4	16	16	0.10	2560
Shapes0	4	0	1	4	4	0.01	1600
Shapes1	4	180	2	8	8	0.01	6400
Shirts	8	180	2	16	16	0.1	2560
Swim	10	180	2	20	20	4.97	80
Trousers	17	180	2	34	34	0.53	2181

参考文献

- 贾志欣, 殷国富, 罗阳. 二维不规则零件排样问题的遗传算法求解. 计算机辅助设计与图形学学报, 2002, 14(5): 467-670. [doi: 10.3321/j.issn:1003-9775.2002.05.020]
- 李建涛, 黄星梅, 钟志华. 二维矩形件切割的路径优化. 机械设计与制造, 2005, (4): 86-87. [doi: 10.3969/j.issn.1001-3997.2005.04.040]
- 王书文, 黄星梅, 李建涛. 二维矩形件组块优化切割的研究与实现. 苏州大学学报(工科版), 2007, 27(6): 49-52.
- 曹德列. 不规则图形数控切割关键技术研究[硕士学位论文]. 武汉: 华中科技大学, 2012.
- O'Rourke J. Computational Geometry in C. 2nd ed. New York: Cambridge University Press, 1998.
- Burke EK, Hellier RSR, Kendall G, et al. Complete and robust no-fit polygon generation for the irregular stock cutting problem. European Journal of Operational Research, 2007, 179(1): 27-49. [doi: 10.1016/j.ejor.2006.03.011]
- Bennell JA, Dowland KA, Dowland WB. The irregular cutting-stock problem - A new procedure for deriving the no-fit polygon. Computers & Operations Research, 2001, 28(3): 271-287.
- Dowland K A, Dowland WB. Packing problems. European Journal of Operational Research, 1992, 56(1): 2-14. [doi: 10.1016/0377-2217(92)90288-K]
- Agarwal PK, Flato E, Halperin D. Polygon decomposition for efficient construction of Minkowski sums. Computational Geometry, 2002, 21(1-2): 39-61. [doi: 10.1016/S0925-7721(01)00041-4]
- Art Jr RC. An approach to the two dimensional irregular cutting stock problem. Massachusetts: IBM Cambridge Scientific Centre, 1966.