

文章编号:1006-2467(2021)05-0536-08

DOI: 10.16183/j.cnki.jsjtu.2020.208

基于轨迹线的临界多边形算法拓展

周 鑫¹, 赖晓阳¹, 孟祥群², 王 堃², 唐厚君¹

(1. 上海交通大学 电气工程系, 上海 200240; 2. 上海方菱计算机软件有限公司, 上海 200241)

摘 要: 排料问题是指如何在有限的空间内装下最多指定形状物体的问题, 在工业生产中有着重要的意义. 其求解需要频繁对物体进行相交校验以判断排料位置是否合法. 临界多边形算法可以用于加速相交校验过程, 但算法本身不能计算曲线, 限制了其应用. 一种基于移动碰撞法的临界多边形算法可以将工件轮廓拓展至圆弧, 但其计算速度较慢. 针对该问题, 在基于轨迹线的临界多边形算法的基础上, 分析并改进了该算法的轨迹生成策略以及外包络轮廓算法. 改进后的算法能够在较短的时间内计算出包含圆弧的临界多边形, 同时解决了效率和精度问题. 最后, 在实际的冲床上进行了加工测试, 测试结果验证了算法的正确性与效率.

关键词: 临界多边形; 轨迹线算法; 排料问题

中图分类号: TP 391

文献标志码: A

Promotion of a No Fit Polygon Algorithm Based on Trajectory

ZHOU Xin¹, LAI Xiaoyang¹, MENG Xiangqun², WANG Kun², TANG Houjun¹

(1. Department of Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China;

2. Shanghai Fangling Software Co., Ltd., Shanghai 200241, China)

Abstract: The nesting problem is how to nest objects of a specified shape in a two-dimensional space to obtain the maximum space utilization rate, which is of great significance in industrial production. The solution to the nesting problem requires frequent cross-checking of the objects to determine whether the nesting position is legal. The No Fit Polygon algorithm can be used to accelerate the procedure of cross-checking, but the algorithm cannot be used to calculate shapes containing curves, which limits its application. An algorithm based on orbit sliding can calculate No Fit Polygon between shapes which includes arc, but its effectiveness is not satisfactory. Aimed at this problem, and based on trajectory algorithm, the trajectory generation strategy and the profile algorithm are analyzed and improved. The improved algorithm can calculate the No Fit Polygon between shapes containing arc in a shorter time, solving both accuracy and effectiveness problems. Finally, the algorithm is tested in the real punch production process and the results of the test confirms the correctness and effectiveness of the algorithm.

Key words: No Fit Polygon; trajectory algorithm; nesting problem

临界多边形这一概念由 Adamowicz 等^[1]提出, 广泛应用于 CAD、图像图形学等领域. 临界多边形

最重要的应用为排料问题^[2]的相交校验. 排料问题可概括为如何在有限的空间内装下最多指定形状的

收稿日期: 2020-07-06

作者简介: 周 鑫(1995-), 男, 湖北省宜昌市人, 硕士生, 研究方向为机器视觉和自动化系统控制.

通信作者: 唐厚君, 男, 教授, 博士生导师, 电话(Tel.): 13564887658; E-mail: hjtang@sjtu.edu.cn.

物体^[3]. 排料问题需要测试待排料位置中能否放下指定形状的物体,此时需要判断待排料工件是否与已排料的工件相交,这个过程称为相交校验. 临界多边形是排料问题相交校验计算的重要工具. 研究临界多边形对工业生产具有重要的学术价值和实用价值.

国内外学者对临界多边形进行了许多研究,提出了多种临界多边形的计算方法^[4-5]. 这些算法包括蚁群算法^[6]、元启发算法^[7]、移动碰撞法^[8]、模拟退火算法^[9]、遗传算法^[10]、重力临界多边形算法^[11]及轨迹线算法^[12-13].

以上算法有一个共同的局限:算法仅能对多边形的排样进行求解. 实际工业环境中,待排样的物体并非严格的多边形. 一种方法是用多边形对其轮廓进行拟合,这样在精度上势必有所损失. 一种基于移动碰撞法的临界多边形算法^[14]可以计算由包含圆弧和线段的临界多边形,解决了精度损失问题,但其

计算速度有待提高.

本文在分析了临界多边形的多种计算方法后,研究了计算速度较快的轨迹线算法,并将其加以拓展,使其能够计算包含圆弧和线段的临界多边形,解决了精度损失问题,同时保证了计算的效率. 在冲床自动送料机的排样中对算法进行了测试和实验,并将其与基于移动碰撞法的临界多边形算法和普通相交校验算法加以比较,验证了算法的效率和精度.

1 临界多边形的定义

定义 1 临界多边形. 给定多边形 A 与多边形 B , 固定多边形 A , 令多边形 B 作不旋转的刚体运动绕多边形 A 运动一周, 绕行过程中, 保证 A 与 B 边界上至少有一点互相靠接, 且 A 、 B 不重叠. 在 B 上任取一参考点 M , 在绕行过程中 M 的运动轨迹为一闭合的多边形, 此多边形就是 B 相对于 A 的临界多边形, 记作 NFP_{AB} , 如图 1 所示.

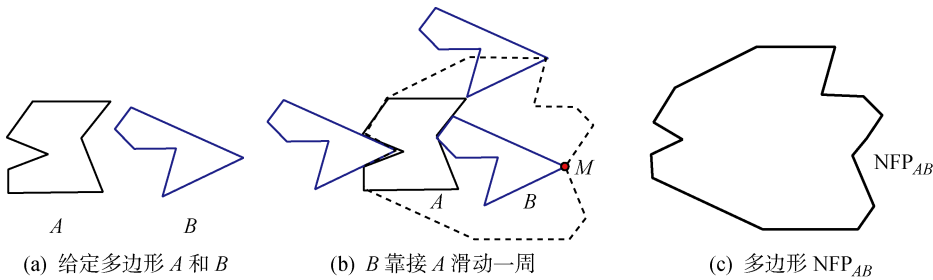


图 1 临界多边形 NFP_{AB} 的定义

Fig. 1 Definition of No Fit Polygon NFP_{AB}

按上述方式构造出来的临界多边形 NFP_{AB} 有着明确的物理意义, 当多边形 B 的参考点 M 落在 NFP_{AB} 以内时, 多边形 A 与多边形 B 互相重合. M 落在 NFP_{AB} 边界上时, A 与 B 刚好相互靠接. 当点 M 落在 NFP_{AB} 边界以外时, A 与 B 不重合. 基于这样的物理性质, 求解出临界多边形后, 可以方便地判断出任意相对位置下多边形 A 、 B 是否重叠.

在求解排料问题的过程中, 需要频繁地判断待排料的工件是否与边界或者已排料的物体重叠, 即待排料的物体位置是否合法, 此过程被称为相交校验. 由临界多边形的定义可知, 判断多边形 A 与 B 是否重叠的问题可以转化为判断 B 上的参考点 M 是否在 NFP_{AB} 内部的问题, 并且只需计算一次临界多边形, 即可对任意相对位置下多边形 A 与多边形 B 的相交校验进行计算. 理论和实践都证明, 在排料问题中使用临界多边形能够大大节省相交校验的用时.

2 算法整体流程和思想

计算临界多边形的轨迹线算法思想可以描述如下: 临界多边形是两个多边形靠接绕行过程中的轨迹, 绕行过程中都是多边形 A 与 B 仅有两种状态, A 的顶点与 B 的边靠接或者 B 的顶点与 A 的边靠接. 顶点在边上靠接并滑动的过程会生成一条轨迹线. 若枚举所有的顶点与边, 求解相应状态下的轨迹线, 则待求的临界多边形为这些轨迹线组成集合 S 的一个子集. 由临界多边形的定义可知, S 内轨迹线围成的最大多边形即为待求的临界多边形.

轨迹线算法的核心是如何快速计算轨迹线的集合 S , 并根据 S 求解对应临界多边形. 本文改进了原有轨迹线算法的轨迹生成策略以及临界多边形求解算法, 使其能够求解包含圆弧的物体间的临界多边形. 下文将详细介绍引入圆弧后轨迹线的求取以及如何从轨迹线集合中提取对应的临界多边形.

为了方便起见, 指定两物体的轮廓线方向, 轮廓

线首尾顺次连接,沿逆时针方向绕物体一周,并将大于 180° 的圆弧拆分成相连的两段等长圆弧,保证每段圆弧对应的圆心角小于 180° .

2.1 轨迹线的求取

由轨迹线的定义可知,所有物体可能的靠接方式都将形成轨迹线. 复杂物体形成的轨迹线的数量相当巨大,为了简化计算,需要对靠接方式进行简单的校验,排除部分不可能的靠接方式,减少轨迹线的数量.

下面给出顶点张角和轮廓线法向量的定义.

定义 2 顶点的张角. 对顶点 P ,求 P 点两侧轮廓线在顶点 P 处的切向量 α_1, α_2 . 其中 α_1 对应的轮廓线在前,由向量 α_1, α_2 构成的角 $\angle P_1 P_2 P_3$ 记作顶点 P 的张角,其中 $P_1 P_2$ 对应向量 $\alpha_1, P_2 P_3$ 对应向量 α_2 . 图 2、3 分别为 P 点两侧为线段、 P 点两侧有圆弧边时, P 点张角的示意图.

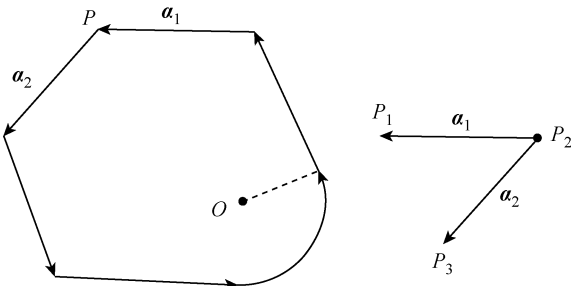


图 2 P 两侧均为线段时顶点 P 的张角
Fig. 2 Expanding angle of P when both sides of P are segments

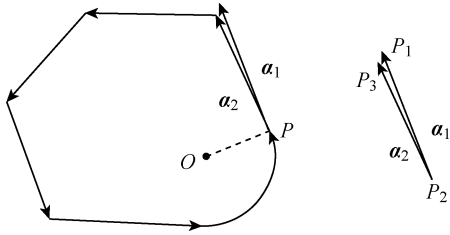


图 3 P 两侧有圆弧时顶点 P 的张角
Fig. 3 Expanding angle of P when either side of P is arc

定义 3 角的凹凸性. 对组成几何图形的某一顶点 P ,求取 P 点的张角 $\angle P_1 P_2 P_3$,若 P_1 位于射线 $P_2 P_3$ 的顺时针方向,则点 P_2 处顶角为凸角,否则点 P_2 处顶角为凹角,如图 4 所示.

定义 4 轮廓线的法向量. 以图 5 所示的多边形 $V_1 V_2 V_3$ 为例,对于线段边 $V_1 V_2$,将其方向向量逆时针旋转 90° 即可得到其法向量 n . 对圆弧边 $V_2 V_3$,圆弧上任意一点 P' 均有相应的法向量,为该点切线方向向量逆时针旋转 90° 的向量,这些法向量的集合为圆弧的法向量. 因此,线段的法向量方向为一确定值,圆弧的法向量方向为一个角度区间.

圆弧线可以分为外凸圆弧和内凹圆弧. 由于轮廓线是按照逆时针方向排列的,则轮廓内部始终位于轮廓线方向左侧,根据此性质可以简单求出圆弧的凹凸性. 下面给出圆弧的外凸与内凹的判定方法:若圆弧 A_1 相对于圆心 O 以逆时针方向旋转,则 A_1 为外凸圆弧,否则 A_1 为内凹圆弧.

引入圆弧后,物体轮廓将包含线段与圆弧两种

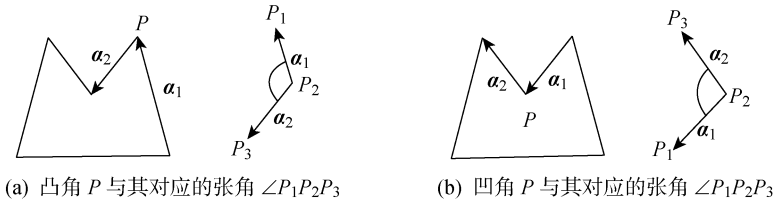


图 4 凸角凹角的示意图
Fig. 4 Diagram of convex angle and concave angle

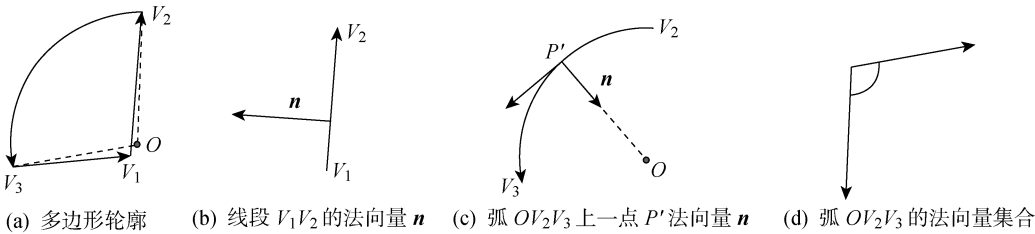


图 5 轮廓线的法向量定义
Fig. 5 Definition of normal vector of contour

边. 此时的靠接状态可以分为三种:一多边形的顶点与另一多边形的边相互靠接、一多边形的圆弧边与另一多边形的线段边相互靠接以及两多边形的圆弧边相互靠接. 以图 6 所示的两个物体为例,分析物体相互靠接的情况.

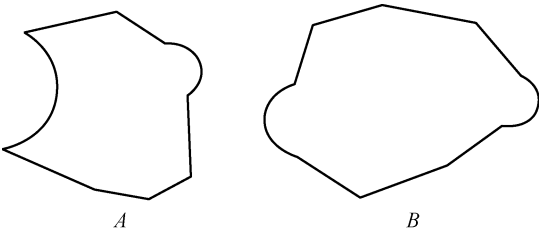


图 6 轮廓示意图
Fig. 6 Diagram of contour

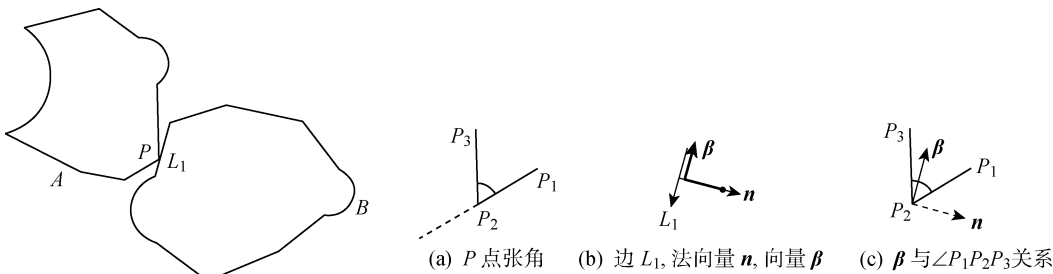


图 7 顶点与线段靠接
Fig. 7 Vertex touch segment

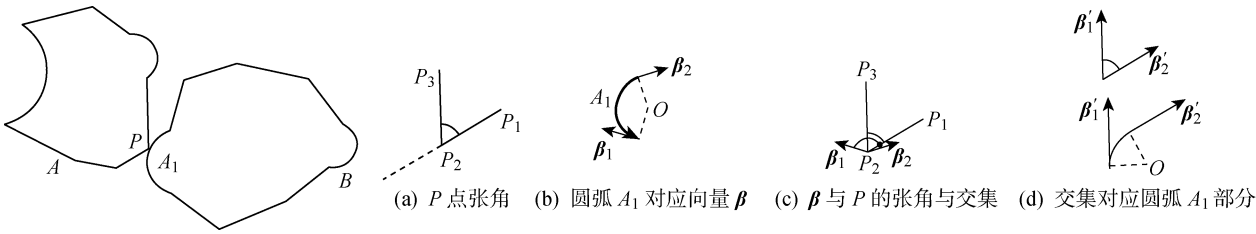


图 8 顶点与圆弧靠接
Fig. 8 Vertex touch arc

2.1.2 线段与圆弧边相互靠接 若边 L_1 和圆弧 A_1 靠接,此时 L_1 必定与圆弧 A_1 相切,且 A_1 为外凸圆弧. 如图 9 所示,求取 A_1 上一点 P' ,使 P' 的法向量 n_2 和 L_1 的法向量方向 n_1 相反. 若点 P' 存在且 A_1 外凸,边 L_1 和圆弧 A_1 可以靠接,靠接点为 P' .

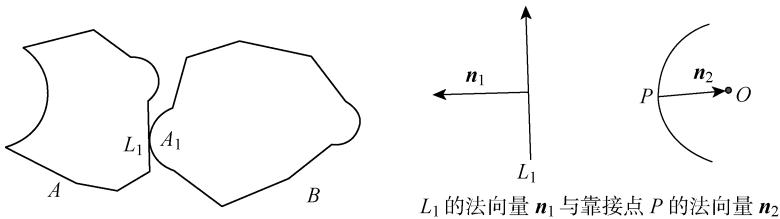


图 9 线段与圆弧靠接
Fig. 9 Segment touch arc

2.1.1 顶点与边的靠接 对于外接 NFP 而言,若顶点 P_2 为凹角,则该点不可能与边靠接.

若顶点 P_2 不为凹角:

情况 1 当边为线段时,以 A 的顶点 P 靠接 B 的 L_1 边为例,将边 L_1 的法向量 n 逆时针旋转 90° 后得到向量 β ,则且仅当 β 位于张角 $\angle P_1 P_2 P_3$ 内部时,顶点 P 与边 L_1 可能靠接,如图 7 所示.

情况 2 当靠接边为圆弧时,如图 8 所示. 当 P 与圆弧 A_1 靠接时,将 A_1 的所有法向量 n 逆时针旋转 90° 得到对应的向量 β . A_1 的 β 向量是一个集合 $\{\beta_1, \beta_2\}$,求该集合与 P 的张角的交集. 若交集不为空,则点 P 可能与弧 A_1 靠接. 设交集为 $\{\beta'_1, \beta'_1\}$,点 P 可能与 A_1 靠接的弧部分对应的向量 β 落在此交集中.

2.1.3 圆弧与圆弧相互靠接 此时的靠接情况分为 3 种:外凸圆弧与外凸圆弧相互靠接、外凸圆弧与内凹圆弧相互靠接以及两段内凹圆弧相互靠接. 两段内凹圆弧不可能相互靠接,不纳入讨论范围,故实际的靠接情况仅有 2 种.

情况 1 当两段圆弧都是外凸圆弧时,如图 10 所示,两段圆弧 A_1 、 A_2 均为外凸圆弧,分别求取 A_1 与 A_2 的法向量,并将 A_2 的法向量逆时针旋转 180° . 若这两段法向量的角度存在交集, A_1 与 A_2 可能相互靠接.

情况 2 当两段圆弧一个为外凸圆弧,另一个

为内凹圆弧时,如图 11 所示,圆弧 A_1 为外凸圆弧,圆弧 A_2 为内凹圆弧. 先判断 A_2 的半径是否小于 A_1 ,若是,两段圆弧不可能靠接. 否则计算 A_1 与 A_2 的法向量范围,并将 A_2 的法向量旋转 180° ,得到对应法向量范围 $[n'_{21}, n'_{22}]$,若两者交集不为空,则 A_1 与 A_2 可能靠接.

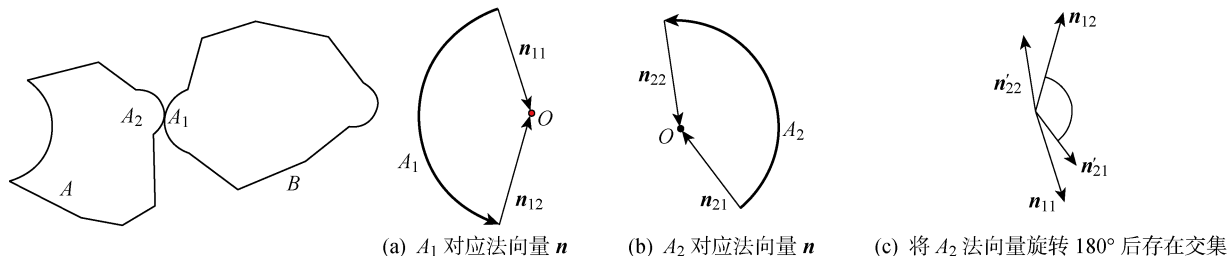


图 10 两段外凸圆弧靠接

Fig. 10 Two convex arc touch

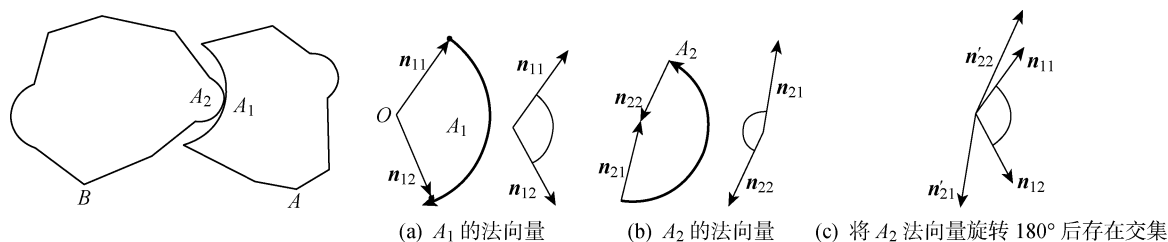


图 11 外凸圆弧与内凹圆弧靠接

Fig. 11 Convex arc touch concave arc

以上是所有可能靠接的轨迹线情况. 综上,轨迹线集合 S 可以用以下算法求取:

对轮廓 A 、 B 的所有顶点,计算其张角,对所有的边,计算其法向量方向(对于圆弧线求取其法向量范围).

(1) 遍历轮廓 A 、 B 的所有顶点,对每个顶点 P ,求其张角 $\angle P_1 P_2 P_3$,并判断其凹凸性,舍弃其中的凹角. 再求取 A 、 B 中所有边的法向量.

(2) 对于(1)中求取的每个张角 $\angle P_1 P_2 P_3$,遍历轮廓 B 的所有边 L ,将(1)中求取的法向量逆时针旋转 90° ,得到向量或向量集合 β . 若 β 的方向落入 $\angle P_1 P_2 P_3$ 内,说明轮廓 B 的边 L 和轮廓 A 的顶点 P 可能靠接,将靠接得到的轨迹线加入集合 S .

(3) 对于轮廓 B 的所有顶点 P ,重复步骤(1)、(2),计算轮廓 B 顶点靠接轮廓 A 的边形成的轨迹线集合.

(4) 轮廓 A 任取一条边 L_1 ,轮廓 B 任取一条边 L_2 ,遍历所有这样的边组合 (L_1, L_2) . 对每一个组合进行如下操作:

(a) 若 L_1 、 L_2 均为线段,不作任何操作,计算下一个组合.

(b) 若 L_1 、 L_2 其中有一条边为线段,另一条边为圆弧,将线段边的法向量 n_1 旋转 180° 后得到向量 n_2 ,若 n_2 落在圆弧的法向量范围内,说明两条边可能靠接. 靠接点的法向量与 n_2 重合. 计算 L_1 、 L_2 相互靠接的轨迹线,加入集合 S .

(c) 若 L_1 、 L_2 均为圆弧,判断 L_1 、 L_2 的凹凸性,若 L_1 、 L_2 均内凹,计算下一个组合. 若 L_1 、 L_2 均外凸,将 L_2 的法向量旋转 180° . 若旋转后 L_1 与 L_2 法向量范围有所重合,说明 L_1 、 L_2 可能相交,计算 L_1 、 L_2 相互靠接的轨迹线,加入集合 S . 若 L_1 、 L_2 一个外凸一个内凹,当外凸的圆弧半径小于内凹圆弧时,将 L_2 的法向量旋转 180° . 若旋转后 L_1 与 L_2 法向量范围有所重合,说明 L_1 、 L_2 可能相互靠接,计算 L_1 、 L_2 相互靠接的轨迹线,加入集合 S .

2.2 从轨迹线集合中求取临界多边形

由临界多边形的定义可知,当参考点 P 落在临界多边形上时,待校验的两物体刚好相互靠接. 由轮廓线的定义可知,当参考点 P 落在轮廓线上时,待校验的两个物体至少有一个交点,故而轮廓线集合中所有点均在临界多边形内部或边上. 故只需要寻找轮廓线集合的最小包络轮廓,此轮廓即为待求的

临界多边形.

临界多边形 N 求取的算法步骤如下:

- (1) 将轮廓线集合置于平面直角坐标系中, 计算轮廓线集合 S 内部所有交点, 以及圆弧 Y 值最小的点, 组成点集 V .
- (2) 取 V 中 Y 值最小的点, 作为轮廓起始点 P_s , 选取过 P_s 且与 X 轴夹角最小的边作为起始边.
- (3) 计算与当前边可能相交的所有点, 取离当前点最近的交点作为截断点, 将当前点与截断点之间的边 L 部分加入外围轮廓 N 中.
- (4) 计算当前边在截断点处的切线向量 α , 并计算过截断点的其他边在截断点处的切线向量, 取与 α 右侧夹角最小的向量所在边为下一个当前边.
- (5) 重复步骤 (3) ~ (4), 直至截断点与 P_s 重合. 此时外围轮廓 N 即为临界多边形.

3 算法复杂度分析

假定多边形 A 是由 m 条边构成, 多边形 B 是由 n 条边构成, 采用轨迹线算法可以生成 k 条轨迹线, 最终生成的临界多边形有 l 条边. 对于某个排料问题, 需要在 p 个不同的相对位置上对两多边形作相交校验.

对比移动碰撞法和轨迹线算法两种临界多边形算法, 根据文献[12], 移动碰撞法的时间复杂度为 $O(lmn)$, 轨迹线算法的时间复杂度为 $O(lk)$, 由轨迹线的求取方式可知 $k \leq mn$, 故而两算法最大时间复杂度相同. 但轨迹线算法根据法向量方向事先排除了大量轨迹线, 实际的轨迹线算法时间复杂度远小于 $O(lmn)$. 故而轨迹线算法的效率会明显高于移动碰撞法.

对比临界多边形算法和常规相交校验算法. 常

规的相交校验是判断多边形 B 内任意一点是否落在多边形 A 内. 若是, 则两多边形重叠; 若不是, 则将多边形 A 的每一条边与多边形 B 的每一条边分别作相交校验. 如有相交, 说明 A 、 B 互相重叠, 若均无相交, 则两多边形不重合. 采用常规相交校验算法, 进行一次套料的时间复杂度为 $O(pmn)$.

若使用临界多边形进行校验, 一次套料的时间复杂度为 $O(pl + lmn)$. 两个校验时间相比较, 可以发现使用临界多边形进行校验, 会在生成临界多边形本身产生额外的开销, 但在多个位置进行重叠校验时由于可以直接重复利用 NFP 的信息, 节省重复计算的时间. 当 $p \gg mn$ 时, 使用临界多边形进行校验能够极大地节省装载问题的计算开销.

实际生产过程中, 工件与板材形状往往不会过度复杂, 但为了最大程度地提高利用效率, 装载问题算法会计算大量的排料位置以寻求最优解. 以冲床自动送料机的加工为例, 一次套料中, p 的数量级在 10^4 左右, 而工件边数一般少于 60, 满足 $p \gg mn$ 这个条件. 因此采用临界多边形算法可以极大地提高运算速度.

4 实验结果与分析

本文对冲床送料机常用零件库中的 60 种不同工件进行了临界多边形计算, 总共测试了 330 种不同的组合. 其运行平台为 PC 机, 其处理器为 i7-7700HQ, 2.81 GHz, 8 GB 内存, Windows 10 操作系统. 经过测试, 算法能够生成正确的临界多边形, 通过临界多边形进行的相交校验计算与传统的相交校验计算结果一致, 验证了算法的正确性.

图 12 为几组不规则工件生成的临界多边形. 表 1 对比了基于轨迹线的临界多边形算法和基于移动

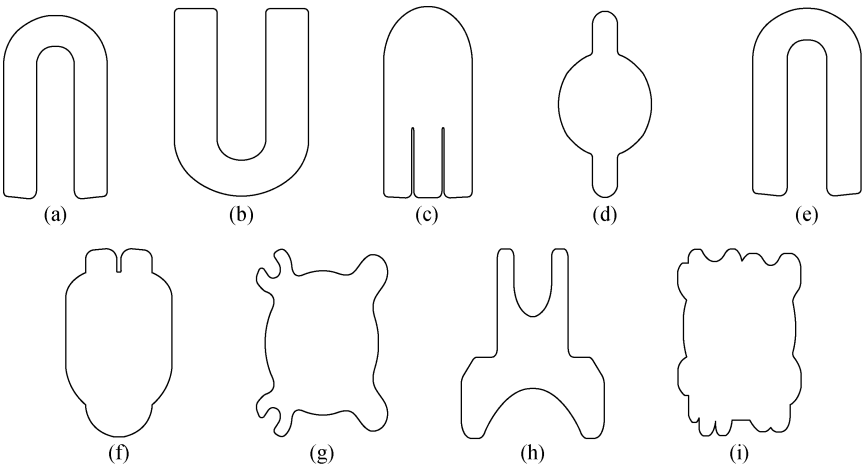



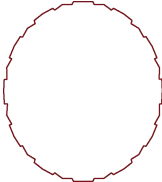


图 12 几组不规则工件生成的临界多边形

Fig. 12 Critical polygons generated by several groups of irregular workpieces

表 1 两种临界多边形算法计算效率的对比

Tab. 1 Comparisons of computational efficiency of two critical polygon algorithms

多边形类别	临界多边形	运行时间/ms		优化效率/%
		轨迹线算法	移动碰撞法	
简单凸多边形		0.47	0.55	15.5
凹多边形		9.15	11.21	18.4
扳手		50.91	67.25	24.3
齿轮		167.03	267.68	37.6

碰撞的临界多边形算法的效率.

由表 1 可知,基于轨迹线的临界多边形算法在计算效率上优于基于移动碰撞的临界多边形算法,且工件形状越复杂,其计算效率提高的效果越明显.

下面以冲床自动送料机为实际场景,对算法进

行测试,将基于轨迹线的临界多边形算法和常规相交校验算法效率进行对比分析. 冲床自动送料机使用的工控机处理器为 i7-4900MQ, 2.8 GHz, 8 GB 内存. 图 13 为一份典型的冲床自动送料机加工样例,加工的工件为灰色的五角星形工件.

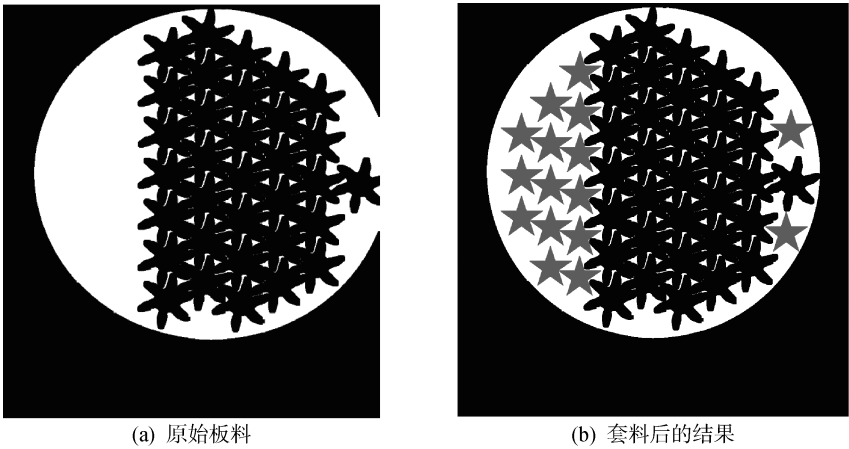


图 13 冲床自动送料机加工样例

Fig. 13 Processing example of punch automatic feeding machine

表 2 给出了冲床自动送料机在加工不同形状板材时,采用两种算法的效率对比,测试的板材为眼子料(即常规钢板被加工后剩余的边角料)和板料(分为窄与宽两种),测试工件相同,均为扳手形工件(见表 1 序号 3).

由表 2 可知,与原有的相交校验算法相比,使用临界多边形算法显著可以提高算法效率,测试用例中,算法效率提高了约 30%~55%左右. 效率提升

表 2 采用临界多边形算法前后冲床套料用时对比

Tab. 2 Comparisons of nesting time by using critical polygon algorithm and normal algorithm

序号	板料类型	套料用时/ms		效率提升/%
		临界多边形相交校验	常规相交校验	
1	眼子料	145.32	208.92	30.5
2	板料(窄)	208.73	391.41	46.7
3	板料(宽)	448.27	1013.15	55.8

的幅度随着排料区域的扩大而提升. 由理论分析可知, 临界多边形的生成相当耗时, 但生成临界多边形之后进行相交判断相比常规算法更为迅速. 当排料区域变大时, 排料算法对相同工件的相交判断会更为频繁, 使得临界多边形算法的优势得以扩大, 效率提升更为明显. 表 2 结果与理论分析相符.

5 结语

本文在基于轨迹线的临界多边形算法基础上, 将算法加以改进, 拓展了算法的应用范围. 改进后的算法能够正确计算包含圆弧的物体所形成的临界多边形, 解决了使用多边形拟合曲线带来的精度损失问题. 对比现有的基于移动碰撞法的临界多边形算法, 该算法显著提高了计算速度, 在计算复杂物体临界多边形的情况下, 该算法平均能够提升 40% 左右的效率. 在冲床自动送料机上的测试结果验证了本文算法的可行性和有效性.

参考文献:

[1] ADAMOWICZ M, ALBANO A. Nesting two-dimensional shapes in rectangular modules[J]. **Computer-Aided Design**, 1976, 8(1): 27-33.

[2] TERRY W, AMICO M D, IORI M. Bin packing problem with general precedence constraints[J]. **IF-AC-PapersOnLine**, 2015, 48(3): 2027-2029.

[3] MARTELLO S, PISINGER D, VIGO D. The three-dimensional bin packing problem[J]. **Operations Research**, 2000, 48(2): 256-267.

[4] 孙佳正. 基于不完整临界多边形的二维排样问题的研究[D]. 上海: 华东师范大学, 2018.

SUN Jiazheng. Research on 2D Layout problem based on incomplete No-Fit Polygon[D]. Shanghai: East China Normal University, 2018.

[5] XU J J. An optimization algorithm based on no fit polygon method and hybrid heuristic strategy for irregular nesting problem[C]//**The 36th China Control Memories Collection**. Dalian: Technical Committee on Control Theory, Chinese Association of Automation, 2017: 1234-1239.

[6] YANG Q. No Fit Polygon for nesting problem solving with hybridizing ant algorithms[J]. **Journal of Software Engineering and Applications**, 2014, 7(5): 433-439.

[7] VALVO E L. Meta-heuristic algorithms for nesting

problem of rectangular pieces[J]. **Procedia Engineering**, 2017, 183: 291-296.

[8] 杨卫波, 王万良. 改进临界多边形生成算法[J]. **计算机工程与应用**, 2013, 49(1): 32-35.

YANG Weibo, WANG Wanliang. Improved algorithm for No-Fit Polygon calculation[J]. **Computer Engineering and Applications**, 2013, 49(1): 32-35.

[9] 杨卫波, 王万良, 张景玲, 等. 基于遗传模拟退火算法的矩形件优化排样[J]. **计算机工程与应用**, 2016, 52(7): 259-263.

YANG Weibo, WANG Wanliang, ZHANG Jingling, et al. Packing optimization of rectangles based on improved genetic annealing algorithm[J]. **Computer Engineering and Applications**, 2016, 52(7): 259-263.

[10] 刘海明, 周炯, 吴忻生. 应用临界多边形方法与小生境遗传算法求解不规则排样问题[J]. **小型微型计算机系统**, 2016, 37(5): 1002-1007.

LIU Haiming, ZHOU Jiong, WU Xinsheng. Using No Fit Polygon method and niche genetic algorithm to solve irregular layout problems[J]. **Small Microcomputer System**, 2016, 37(5): 1002-1007.

[11] 周炯. 基于临界多边形方法的二维不规则件排样问题及其算法研究[D]. 广州: 华南理工大学, 2015.

ZHOU Jiong. Two-dimensional irregular parts layout problem based on No Fit Polygon method and its algorithm research[D]. Guangzhou: South China University of Technology, 2015.

[12] 汤德佑, 周子琳. 基于临界多边形的不规则件启发式排样算法[J]. **计算机应用**, 2016, 36(9): 2540-2544.

TANG Deyou, ZHOU Zilin. No-Fit-Polygon-based heuristic nesting algorithm for irregular shapes[J]. **Journal of Computer Applications**, 2016, 36(9): 2540-2544.

[13] 刘胡瑶, 何援军. 基于轨迹计算的临界多边形求解算法[J]. **计算机辅助设计与图形学学报**, 2006, 18(8): 1123-1129.

LIU Huyao, HE Yuanjun. New algorithm for No Fit Polygon calculation[J]. **Journal of Computer-Aided Design & Computer Graphics**, 2006, 18(8): 1123-1129.

[14] BURKE E K, HELLIER R S R, KENDALL G, et al. Irregular packing using the line and arc No-Fit Polygon[J]. **Operations Research**, 2010, 58(4): 948-970.

(本文编辑:陈晓燕)