

# 钣金件数控激光切割割嘴路径的优化

刘会霞 王 霄 蔡 兰

(江苏大学机械工程学院 镇江 212013)

**摘 要** 提出了分级规划的三步算法来解决优化问题. 首先用改进的最近邻算法选择打孔点; 然后用智能仿生算法——蚁群算法来求解最佳割嘴路径; 最后根据 Z 轴随动功能约束情况, 用路径调整算法来调整前两步算法确定的最佳路径, 并获得最终割嘴最佳路径. 给出了运行实例并指出进一步的研究工作.

**关键词** 钣金件; 数控激光切割; 割嘴路径优化; 打孔点; 蚁群算法

**中图分类号** TP391; TH16

## Torch Path Optimization for NC Laser Cutting of Sheet Metal Part

Liu Huixia Wang Xiao Cai Lan

(School of Mechanical Engineering, Jiangsu University, Zhenjiang 212013)

**Abstract** An hierarchical three-step approach is presented to solve the optimization problem. First, an improved nearest neighbor algorithm is applied to select pierce points. Then ant system algorithm is used to solve the optimal cutting path. At last, the former path is adjusted to avoid the torch falling down into the finished cut cavity during its vacancy course. Example and the further research are discussed.

**Key words** sheet metal part; NC laser cutting; torch path optimization; pierce point; ant colony algorithm

## 1 问题的提出

CNC 数控激光切割机床(Laser Cutting Machine, LCM)是典型的机、光、电为一体的技术密集型高科技设备. 在工业发达国家, LCM 已被看作为“敏捷制造”模式下的一种快速响应生产的制造设备. 近年来, LCM 正在被越来越多的国内企业所采用, 并在经济发达地区, 如深圳、上海、江南一带发展十分迅猛. 在数控激光切割技术日趋成熟的情况下, 与之配套的专用 CAD/CAM 一体化自动编程系统的性能越来越引起用户的关注, 因为这种应用软件的性能严重制约了这些先进数控激光加工设备的加工

效率与质量. 通过我校 2.5KW(CO<sub>2</sub>)激光切割机床生产实践发现, 在这些性能要求中, 激光切割打孔点位置的合理选取与割嘴路径的有效规划, 对激光切割这样一个高投入、高消耗、更高产出的高新设备而言, 即使单件节省较少的时间, 其效益也是十分可观的. 然而从我们接触的国内外几个商品化激光切割自动编程系统中发现, 它们在打孔点位置的合理选取及割嘴路径的确定方面尚存在如下问题: (1)要求人工交互设置切割路径或在图形中补加有关信息; (2)对割嘴随动功能给切割路径带来的不良影响处理不当或没有考虑, 导致非切割时间增长或易产生切割故障; (3)缺乏较好的割嘴路径优化策略与算法. 国外学者已对此作过一些研究: Manber 等<sup>[1]</sup>解

决了一块板上排样多个零件在火焰切割机切割时打孔点的选取与切割顺序的优化(与激光切割类似),但没有考虑被排样的每个零件的内部又包含多个要加工的内部轮廓的情况;Jackson 等<sup>[2]</sup>考虑了内部轮廓情况且能从已有的 CAD 系统中自动地确定切割的路径,但需用户在图形上补加引入/引出线等信息确定打孔点,并且切割路径也并未优化;Han 等<sup>[3]</sup>考虑了排样零件在激光切割过程中切割温度对切割质量的影响,并纳入了切割路径优化的数学模型中.文献[1-3]皆未涉及割嘴随动功能对切割路径的影响.国内关于激光切割的自动编程系统的论文很少有系统地研究路径优化或随动功能对切割路径的影响.

在单件小批量生产中,尺寸较大且内部包含多个内轮廓特征的这类零件在实际生产中十分常见,而这类零件只能单件加工(如纺织机械设备、收获机械设备上的钣金零件).因此,开展这类零件打孔点的选取及割嘴路径的优化的研究,并将其补加到自动编程系统中是十分必要的.

## 2 割嘴切割路径问题的定义

### 2.1 钣金零件的几何表达及定义

本文研究包含多个内轮廓特征的单个钣金零件,这类零件主要由首尾连接的直线、圆弧、圆等图元构成的多个封闭轮廓图形组成.我们定义有向有序首尾相连组成的封闭轮廓为环(Loop),组成环的基本元素称为边(E),边的端点称为顶点(V),环又分为内环与外环.显然,对单件钣金零件有且只有一个外环,内环可以有多个;边可以是直线、圆弧、圆等,定义直线的顶点是直线的首尾两 endpoint,圆弧的顶点是圆弧的起点与终点.对圆而言,定义一个顶点,这个顶点是圆上的某一点,这点对切割过程而言

既是切割的起点又是切割的终点.本文规定:外环记为  $Oloop$ ,外环顶点总数记为  $\omega$ ,任一顶点记为  $OV_i (i=1,2,\dots,\omega)$ ;任一内环记为  $loop_i$ ,其对应的顶点总数记为  $V(i)$ ,第  $k$  个内环  $loop_k$  对应的第  $j$  个顶点记为  $V_{kj} (k=1,2,\dots,m; j=1,2,\dots,V(k))$ ,所有内外环皆按整体逆时针方向从某一顶点开始排列构成,如图 1 所示.在图 1 中,环集  $loop = \{Oloop, loop_1, \dots, loop_m\}$ ;顶点集  $V = \{V_1, V_2, \dots, V_\omega, V_{11}, V_{12}, \dots, V_{1V(1)}, \dots, V_{ij}, \dots, V_{mV(m)}\}$ .

### 2.2 割嘴路径问题数学模型的建立

激光切割最终的目的是要切割出所有的内环与外环而形成工件,切割内环或外环时,首先以一个环上的某一顶点为打孔点,打穿再切割完该环.按实际切割工艺,切割顺序应遵循先内环后外环的原则,其整个切割过程是割嘴从编程零点(即对刀点)出发,快速行进到某一内环上的打孔点,并开光打孔;再以该打孔点所在的顶点为起点切割完该有向有序的封闭环,即回到开始打孔点;然后关光快速行进到下一个内环上的打孔点上,重复前面类似过程直到切割完所有内环;再关光快速行进到外环打孔点,开光打孔并切割外环;最后从外环打孔点关光快速返回程序零点.在图 1 中假定程序零点为  $P_0$ ,内环  $loop_1$  打孔点为  $V_{11}$ ,  $loop_2$  打孔点为  $V_{21}$ ,  $loop_3$  打孔点为  $V_{31}$ ,  $loop_4$  打孔点为  $V_{41}$ ,  $Oloop$  打孔点为  $OV_1$ ,则一个完整的切割路径可表示为  $P_0 \xrightarrow{\text{快速空移}} OV_1$  (开光打孔)  $\xrightarrow{\text{切割}} V_{12} \rightarrow V_{13} \rightarrow V_{14} \rightarrow V_{15} \rightarrow V_{16} \rightarrow V_{17} \rightarrow V_{18} \rightarrow V_{11}$  (关光)  $\xrightarrow{\text{快速空移}} V_{21}$  (开光打孔)  $\xrightarrow{\text{切割}} V_{22} \rightarrow V_{23} \rightarrow V_{24} \rightarrow V_{25} \rightarrow V_{21}$  (关光)  $\xrightarrow{\text{快速空移}} V_{31}$  (开光打孔)  $\xrightarrow{\text{切割}} V_{32} \rightarrow V_{33} \rightarrow V_{34} \rightarrow V_{35} \rightarrow V_{31}$  (关光)  $\xrightarrow{\text{快速空移}} V_{41}$  (开光打孔)  $\xrightarrow{\text{切割}} V_{42} \rightarrow V_{43} \rightarrow V_{44} \rightarrow V_{41}$  (关光)  $\xrightarrow{\text{快速空移}} OV_1$  (开光打孔)  $\xrightarrow{\text{切割}} OV_2 \rightarrow OV_3 \rightarrow OV_4 \rightarrow OV_5 \rightarrow OV_6 \rightarrow OV_7 \rightarrow OV_1$  (关光)  $\xrightarrow{\text{快速空移}} P_0$ . 由于封闭的内外环必须切割,因此无论从何点开始切割,内外环的长度不变,即不存在路径优化问题.这样,割嘴路径规划就是指如何安排切割顺序,使激光切割过程中快速空移的行程的时间最短.对图 1 而言,就是从  $P_0$  出发,如何选择  $V_{11}, V_{21}, V_{31}, V_{41}, OV_1$  及其加工顺序所构成的回路路径最短.很显然,当打孔点一定时,路径优化的问题就是最著名的旅行售货员问题(Traveling

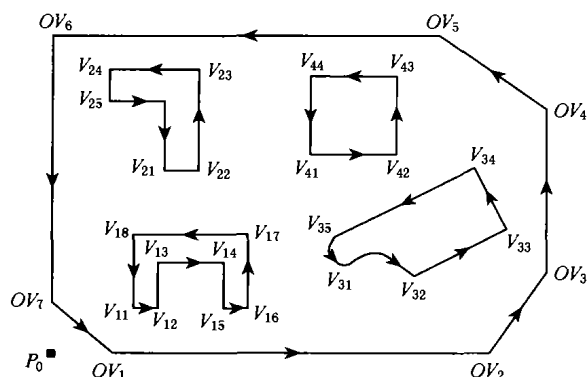


图 1 钣金零件轮廓的定义

Salesman Problem, TSP). 然而, 内外环上的打孔点从切割工艺上讲并不惟一, 可以是环上的任一顶点, 打孔点在切割路径的优化算法中应被考虑的总顶点数为外环总顶点数  $\omega$  加上所有内环总顶点数  $\sum_{k=1}^M V(k)$  个. 为了给出切割路径的数学表达式, 本文中把按某种方法确定的内环  $loop_i$  上的打孔点记为  $P_i$ ,  $Oloop$  上的打孔点记为  $P_E$ . 显然,  $P_i \in \{V_{i1}, V_{i2}, \dots, V_{iV(i)}\}$ ;  $P_E \in \{OV_1, OV_2, \dots, OV_\omega\}$ , 并设对所有打孔点  $P = \{P_0, P_1, P_2, \dots, P_M, P_E\}$  的一个访问顺序为  $T = \{P_0, t_1, t_2, t_M, P_E, P_0\}$ , 其中  $t_i \in \{P_0, P_1, P_2, \dots, P_M, P_E\}$ , 则切割路径的数学模型为

$$\min L = \{d(p_0, t_1) + \sum_{i=1}^{M-1} d(t_i, t_{i+1}) + d(t_M, p_E) + d(p_E, p_0)\}.$$

在实际切割过程中, 切割路径的优化还受到  $Z$  轴随动功能的制约.  $Z$  轴随动功能实质上是一种激光焦点自动跟踪功能, 它能保证在切割过程中使激光切割焦点始终保持在工件表面内, 且与工件表面的距离恒定, 从而避免了因工件变形、表面粘附物等因素影响引起焦点位置变化.  $Z$  轴随动功能会引起下列问题: 当一个环切割完后, 割嘴快速空移到下一个环打孔点的过程中, 随动功能有可能导致割嘴掉入已切割的板材空洞中而造成割嘴损坏, 使机床无法正常切割. 如图 2 所示, 当打孔点为  $t_{i-1}, t_i, t_{i+1}$  时, 切割完  $loop_i$  后,  $loop_i$  区域内钢板会因重力下落, 则割嘴由  $t_i$  快速空移到  $t_{i+1}$ , 会掉入  $loop_i$  形成的空洞中. 而当打孔点  $t_i$  改为  $t'_i$  时, 将不存在此问题. 因为加工完  $loop_i$  后, 割嘴由  $t_{i-1}$  快速空移到  $t'_i$  点, 此时  $loop_i$  还未加工.

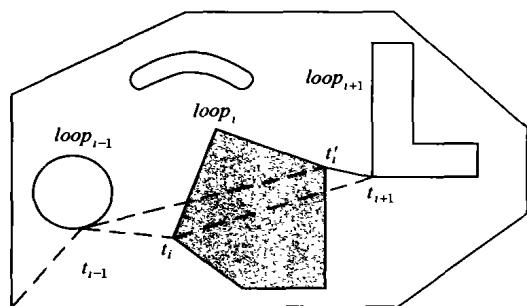


图 2 割嘴掉入切割区域示意图

### 3 打孔点的选择及路径优化

当打孔点确定后, 路径优化问题相当于一个

TSP 问题, 此时切割路径共有  $M!$  条, 如考虑到打孔点的可变化性, 切割路径将为  $M! \times \omega \times \prod_{i=1}^M V(k)$  条. 其数目是十分巨大的, 即使对于如图 1 所示这样简单的零件, 切割路径也有  $4! \times 7 \times 8 \times 5 \times 4 \times 5 = 134\,400$  条. 从形式上看, 打孔点选择及路径优化可被看作一个具有约束的离散变量的优化问题, 然而解决这一问题十分复杂. 由于 TSP 问题本身就属于著名的 NP 问题, 因此若再同时考虑打孔点变化及  $Z$  轴随动功能的影响, 将使这一问题变得更加复杂. 早期人们为了简化这一难题, 采用固定打孔点且不考虑  $Z$  轴随动功能的影响, 这显然很不合理. 本文针对这一问题, 提出了分级规划的三步方法: 先按改进最近邻算法合理选择打孔点; 再按 TSP 问题进行路径优化; 最后再判定空移时是否掉入空洞中, 并按一定算法调整路径使之避开空洞, 有效地解决了这一难题.

#### 3.1 打孔点的确定

打孔点确定的原则是一个内环或外环仅有一个打孔点, 本文按切割工艺采用最近邻算法的一个改进算法确定打孔点. 该算法充分考虑了打孔点对割嘴路径优化的影响.

##### 算法 1. 打孔点的确定

Step1. 从编程零点  $P_0$  出发, 令  $P_k = P_0, P = \{P_0\}$ .

Step2. 遍历所有待加工内环  $loop = \{loop_1, loop_2, \dots, loop_M\}$  对应的顶点集  $V = \{V_{11}, V_{12}, \dots, V_{1V(1)}, \dots, V_{ij}, \dots, V_{MV(k)}\}$ . 找到距  $P_k$  最近的顶点  $P_i = V_{ij}$  (对应内环为  $loop_i$ ), 并将  $P_i$  加入到打孔点集  $Q$  中,  $Q = P + \{P_i\}$ ; 令  $P_k = P_i$ , 然后在  $loop$  集中删除  $loop_i$  及  $loop_i$  对应的顶点  $\{V_{i1}, V_{i2}, \dots, V_{iV(i)}\}$ .

Step3. 依次遍历所有的未访问的内轮廓  $loop$  对应的顶点集, 找到打孔点, 并加入打孔点集  $Q$  中, 令  $P = Q$ .

Step4. 从  $P_k$  出发, 再遍历外环上的所有顶点集  $\{OV_1, OV_2, \dots, OV_\omega\}$ , 找到距离最近的一个顶点即打孔点  $P_E$ , 则所有打孔点  $P = \{P_0, P_1, P_2, \dots, P_M, P_E\}$ .

注意: 当环是圆时, 按最近邻算法, 前一个切割起点与圆上最近点即切割起点, 它是前一个切割起点与该圆圆心连线与圆的交点. 本文为求解方便, 把圆心定义为虚顶点, 以便标识及方便求解圆上的切割起点.

#### 3.2 基于打孔点的路径优化

对所有已知的打孔点  $P = \{P_0, P_1, P_2, \dots, P_M, P_E\}$ , 由第 2.2 节可知路径优化问题等同于著名的 TSP 问题. TSP 问题是一个典型的、易于描述却难以处理的 NP 完全问题. 人们针对 TSP 问题提出了

多种解决办法,典型的优化算法有局部优化、遗传算法、模拟退火法、禁忌搜索、人工神经网络算法和蚁群系统等启发式搜索算法。

20 世纪 90 年代初,意大利学者 Dorigo 等从生物进化的机理中受到启发,通过模拟自然界蚂蚁寻径的行为,提出一种全新的智能仿生算法——蚁群算法<sup>[4]</sup>。蚁群算法不仅能够智能搜索、全局优化,而且具有稳健性(鲁棒性)、正反馈、分布式计算、易于与其他算法结合等特点,在一系列复杂困难的系统优化问题求解中取得了成效,显示出该算法在求解复杂优化问题,特别是离散优化问题方面的一些优越性,如蚁群算法求解 TSP 明显优于其他算法<sup>[5-8]</sup>。本文采用蚁群算法求解激光切割中的 TSP 问题。

### 3.2.1 蚁群算法原理

蚁群算法是对自然界蚂蚁的寻径方式进行模拟而得出的一种仿生算法。虽然单个蚂蚁的行为极其简单,但它们所组成的蚁群群体却表现出极其复杂的行为。它们在找到食物时,总有能力找到一条从食物源到窝巢之间的最优路径,并能随环境的变化而变化,适应地搜索新的路径,产生新的选择,这是因为蚂蚁在寻找路径时会在路径上释放一种特殊的信息素。当他们遇到一个还没有走过的路口时,就随机地挑选一条路径前行,同时释放与路径长度有关的信息素。路径越长,释放的信息素浓度越低。当后来的蚂蚁再次遇到这个路口时选择信息素浓度较高的路径概率就会相对较大,这样大量蚂蚁组成的蚁群集体行为便形成了一种信息正反馈现象。最优路径上的激素浓度越来越大,而其他路径上的信息素浓度却会随时间的流逝而消减,最终整个蚁群会找到最优路径。不仅如此,蚂蚁还能够适应环境变化,当蚁群运动路线上突然出现障碍物时,蚂蚁能够很快地重新找到最优路径。

### 3.2.2 蚁群算法模型及算法实现

**定义.** 切割中打孔点即 TSP 问题中的城市; $d_{ij}$  表示两个打孔点之间的欧氏距离,即 TSP 问题中两个城市间的花费; $b_{i(t)}$  表示  $t$  时刻位于城市  $i$  的蚂蚁的个数,  $m = \sum_{i=1}^n b_{i(t)}$ ;  $\tau_{ij}$  表示  $t$  时刻边  $(i, j)$  上的信息激素浓度,且设  $\tau_{ij}(0) = C$  ( $C$  为常数),  $i, j = 0, 1, \dots, n-1$ ;  $\eta_{ij} = 1/d_{ij}$  表示  $t$  时间边弧  $(i, j)$  的能见度,反映由城市  $i$  转移到城市  $j$  的期望程度。

由蚁群算法原理可知,蚂蚁  $k$  ( $k = 1, 2, \dots, m$ ) 在运动过程中根据各条路径上的信息决定转移方向,随着时间的推移,从前留下的信息逐渐消逝;经

过  $n$  个时刻,蚂蚁完成一次循环,各路径上的信息量要作调整。人工蚁群系统模型<sup>[5-7]</sup>如下:

(1) 设蚁群在并行搜索 TSP 的解。通过一种信息激素作媒体相互通信,在每个结点上且和该结点相连的边的长度上,以信息激素量作为搜索下一结点的试探依据,直到找到一个 TSP 可行解。

(2) 根据路径上的激素浓度,以相应的转移概率来选取下一步路径。在  $t$  时刻蚂蚁  $k$  由一个结点位置  $i$  转移至下一个位置  $j$  的转移概率为

$$P_{ij} = \begin{cases} \frac{\tau_{ij}^{\alpha}(t) \eta_{ij}^{\beta}(t)}{\sum_{s \in S} \tau_{is}^{\alpha}(t) \eta_{is}^{\beta}(t)}, & S \in s \\ 0, & s \notin S \end{cases} \quad (1)$$

其中,参数  $\alpha$  表示信息的相对重要性 ( $\alpha \geq 0$ );  $\beta$  表示能见度的相对重要性 ( $\beta \geq 0$ );  $S$  表示可行结点集,即蚂蚁  $k$  下一步允许选择的的城市。 $\alpha, \beta$  分别反映了蚂蚁在运动过程中所积集的信息以及启发式因子在蚂蚁选择路径中所起的不同作用。

(3) 不再选取上次循环已经走过的路径为下一步路径,而由禁忌表控制。

(4) 当  $m$  个蚂蚁按式(1)找到可行解后,则修改各边的信息量,即调整信息激素强度的更新方程为

$$\tau_{ij}(t+n) = \rho \times \tau_{ij}(t) + \Delta \tau_{ij}, \rho \in (0, 1) \quad (2)$$

其中  $\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k$ ,  $\Delta \tau_{ij}^k$  表示第  $k$  只蚂蚁在本次循环中留在路径  $(i, j)$  上的信息激素量。

本文采用能反映整体信息的 Ant-Cycle System 模型,故

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{如果在时刻 } t \text{ 和 } t+n \text{ 之间} \\ & \text{第 } k \text{ 只蚂蚁使用边 } e(t, j) \\ 0, & \text{其他} \end{cases}$$

其中,  $Q$  是一个常数,  $L_k$  是第  $k$  只蚂蚁周游的路程长度;  $\Delta \tau_{ij}$  表示本次循环中路径  $(i, j)$  上的信息激素量的增量;参数  $\rho$  表示信息的特征性;  $1-\rho$  表示信息衰减度,即信息消逝程度。

对上述系统模型,其求解算法步骤可归结为算法 2。

#### 算法 2. TSP 模型求解

Step1. 初始化。

$NC \leftarrow 0$  ( $NC$  为迭代步数或搜索次数);  $\tau_{ij}(t) = C$ ;  $\Delta \tau_{ij} = 0$ , 将  $m$  只蚂蚁置于  $n$  个顶点上。

Step2. 把  $k$  个蚂蚁的初始城市号放置到  $tabu_k(s)$  中。

Step3. 根据式(1)概率来选择下一步应该到达的城市,

将第  $k$  个蚂蚁移到城市  $j$ , 并将  $j$  插入到  $tabu_k(s)$  中.

Step4. 计算第  $k$  个蚂蚁的总路径长度  $L_k$ , 更新找到的最短路径.

Step5. 按更新方程(2)更新边上的信息激素浓度.

Step6. 对各边弧  $(i, j)$ , 置  $\Delta\tau_{ij} \leftarrow 0$ ;  $NC \leftarrow NC + 1$ .

Step7. 若  $NC <$  预定的迭代次数且无退化行为, 则转 Step2; 否则, 输出当前最好解, 终止程序.

蚁群算法的核心有三条: (1) 选择机制. 信息激素越多, 被选中的概率越大; (2) 信息激素更新机制. 路径越短, 信息激素增加越快; (3) 协作机制. 个体之间通过信息激素进行信息交流.

### 3.3 割嘴切割路径的调整算法

过去, 人们为避免割嘴掉入已切割区域形成的空洞问题, 一般通过小功率试切割的办法或通过自动编程中的加工模拟来检查这一情况, 然后人工干预来调整打孔点位置或直接修改生成的数控指令, 即存在问题的快速空移空行程指令前补加取消  $Z$  轴随动指令, 让  $Z$  轴抬起某一高度, 再快速空移到下一打孔点, 并补加启动随动指令使割嘴下落打孔并切割. 但由于  $Z$  轴随动每次取消与启动系统响应速度较慢, 严重影响切割效率, 因此应尽可能调整打孔点, 仅当无法通过打孔点调整时, 才用  $Z$  轴随动功能.

由于人工干涉严重影响了数控编程效率, 并易引起差错, 因此本文研究了一个有效地解决这一问题的算法. 假设经过 3.1、3.2 算法优化后的打孔点顺序 (即切割顺序) 为  $\{P_0, t_1, t_2, \dots, t_M, P_E, P_0\}$ , 显然切割路径调整是通过调整环上打孔点的位置实现的. 如图 2 所示.

#### 算法 3. 割嘴路径调整算法

Step1. 根据打孔点顺序建立待切割环集  $WP$  及切割环集  $PP$ , 每切割完一个环, 则从待切割环集  $WP$  中转入已切割环集  $PP$  中.

Step2. 判定打孔点  $t_i$  到  $t_{i+1}$  快速空行程构成的直线是否与已切割环集  $PP$  的某一边有交点 (除  $t_i$  外) (如图 2 所示), 完成所有路径判定后则程序停止.

a. 若无交点, 不需调整.

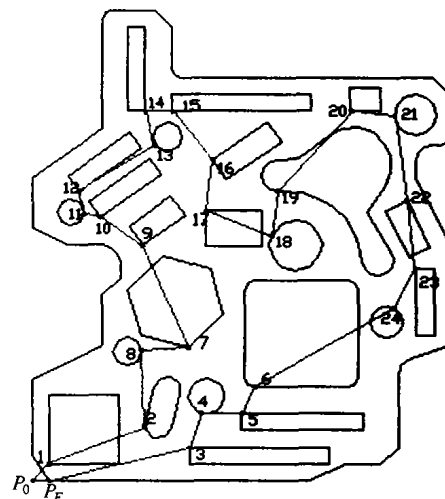
b. 若有交点, 则将打孔点  $t_i$  调整为该交点所在环上的边的最近端点  $t'_i$ , 再返回 Step2 重新判定; 若无交点则调整成功.

c. 若还有交点, 则依次选环上各边端点继续按 Step2 判定; 若环上所有边选完后都存在交点, 则转 Step3.

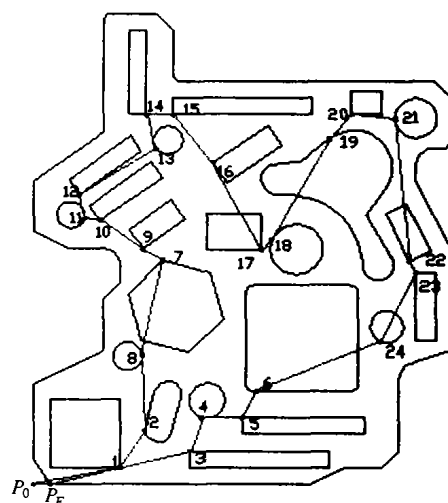
Step3. 标识  $t_i, t_{i+1}$  点, 以便在数控指令生成时在  $t_i$  到  $t_{i+1}$  快速移动指令前补加取消  $Z$  轴随动指令, 在  $t_i$  到  $t_{i+1}$  快速移动指令后补加启动  $Z$  轴随动指令. 转 Step2.

### 3.4 实例

本文算法用 Visual C++ 6.0 编写, 在 P IV 1.7GHz 高档微机上调试并通过运行. 人工蚁群算法中  $\alpha, \beta, \varphi, \rho, m$  等参数对算法性能有较大影响:  $\alpha$  值的大小表明留在每个结点上的信息量受重视的程度,  $\alpha$  值越大, 蚂蚁选择以前选过的点的可能性越大, 但过大会使搜索过早满足于局部最小点;  $\beta$  的大小表明启发式信息受重视的程度;  $\varphi$  值会影响算法的收敛速度,  $\varphi$  过大会使算法收敛于局部最小值, 过小又会影响算法的收敛速度, 随着问题规模的增大,  $\varphi$  值也需随之变化;  $\rho$  表示信息的持久性, 减少  $\rho$  虽可以提高算法的全局搜索能力, 但又会使算法的收敛速度降低. 蚂蚁数目越多, 算法的全局搜索能力越强, 但算法的收敛速度减慢. 蚁群算法中,  $\alpha, \beta, \varphi, \rho$  等参数的设定目前尚无理论上的依据, 只有经



a 算法 1, 2 求得的初始切割路径



b 算法 3 求得的最终切割路径

图 3 打孔点及最佳割嘴路径

验结果<sup>[7]</sup>: (1)  $\cos \alpha \leq 5$ ; (2)  $1 \leq \beta \leq 5$ ; (3)  $0.1 \leq \rho \leq 0.99$ , 一般  $\rho$  取 0.7 左右; (4)  $1 \leq \varphi \leq 10\,000$ . 图 3 所示为一个实例运行结果. 在图 3a 中, 由算法 1 确定的打孔点为  $P_0, 1, 2, 3, \dots, 24, P_E$ , 由算法 2 求得的最佳割嘴路径为图中各打孔点的连线; 图 3b 所示为由算法 3 所获得的最终割嘴最佳路径.

## 4 结束语

本文对钣金件切割中割嘴路径优化问题进行了数学建模, 这是一个极其复杂的、具有约束的离散变量的优化难题. 文中提出了分级规划的三步方法并给出了三个实现算法, 有效地解决了这一难题. 其算法实现程序已被加入我校自行开发的激光切割自动编程系统中, 并取得了较好的应用效果. 然而在实际应用中, 我们又发现本文算法对薄板切割还不能令人满意. 薄板切割中热影响将严重影响切割质量, 而切割路径不同引起的热影响也不同. 因此, 当切割参数优化完成后, 还必须在切割路径优化中进一步考虑热影响约束, 这正是我们今后要做的工作.

## 参 考 文 献

- [1] Manber Udi, Israni Sharat. Pierce point minimization and optimal torch path determination in flame cutting [J]. Journal of Manufacturing Systems, 1984, 3(1): 81~88
- [2] Jackson Steven D, Mittal Ravi O. Automatic generation of 2-axis laser-cutter NC machine-program and path planning from CAD [J]. Computer in Industry 1993, 21(2): 223~231
- [3] Han GUK-CHAN, Na SUCK-Joo. A study on torch path planning in laser cutting processes part2: Cutting path optimization using simulated annealing [J]. Journal of Manufacturing Process, 1999, 1(1): 62~70
- [4] Dorigo M, Maniezzo V, Colomi A. The ant system optimization by a colony of cooperating agents [J]. IEEE Transactions on Systems, Man, and Cybernetics Part B, 1996, 26(1): 1~13
- [5] Gambardella L M, Dorigo M. Solving symmetric and asymmetric TSPs by ant colonies [A]. In: Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC'96)[C]. Nagoya: IEEE Press, 1996. 622~627
- [6] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem [J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53~66
- [7] Ma Liang. Ant colony optimization for bottleneck TSP [J]. Computer Engineering, 2001, 27(9): 24~25(in Chinese)  
(马 良. 瓶颈 TSP 的蚂蚁系统优化[J]. 计算机工程, 2001, 27(9): 24~25)
- [8] Li Suoping, Zhang Xiuyuan, *et al.* Theory on artificial ant algorithm and its application in TSP-Problem [J]. Journal of Transportation Systems Engineering and Information Technology, 2002, 2(1): 54~57(in Chinese)  
(黎锁平, 张秀媛, 等. 人工蚁群算法理论及其在经典 TSP 问题中的实现[J]. 交通运输系统工程与信息, 2002, 2(1): 54~57)