

# 基于蚁群优化的点胶控制系统路径规划<sup>①</sup>

马俊<sup>1,2,3</sup> 梅涛<sup>1,3</sup> 黄健<sup>1,3</sup>

(1. 中国科学院合肥智能机械研究所 安徽合肥 230031; 2. 中国科学技术大学自动化系 安徽合肥 230031; 3. 安徽省仿生感知与先进机器人技术重点实验室 安徽合肥 230031)

**摘要:** 点胶是电子封装中的一个重要环节。为了改进传统点胶工艺中胶枪移动轨迹通常按 X/Y 方向进行优化导致加工效率低下的缺点, 引入了基于蚁群优化的路径优化算法。大量的仿真结果表明该方法可以有效缩短胶枪的加工路径长度进而提高生产力, 尤其当待加工点接近随机分布的时候, 本优化算法的加工路径比传统 X/Y 方向优化所产生的路径短 50% 左右。

**关键词:** 点胶 电子封装 路径规划 蚁群优化 旅行商问题

## Path Planning of Fluid Dispensing System Based on Ant Colony Optimization

MA Jun<sup>1,2,3</sup>, MEI Tao<sup>1,3</sup>, HUANG Jian<sup>1,3</sup>

(1. Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China;

2. Department of Automation, University of Science and Technology of China, Hefei 230031, China;

3. The Key Laboratory of Biomimetic Sensing and Advanced Robot Technology, Hefei 230031, China)

**Abstract:** Fluid dispensing is an important process of Electronic Packaging. In order to improve the low efficiency of the commonly with X/Y-axis optimization algorithm in traditional dispensing process, this paper introduces the Ant Colony Optimization (ACO) algorithm to optimize the dispensing route in offline mode. Extensive simulations show that it greatly outperforms the commonly used X/Y-axis optimization and the dispensing route generated is roughly 50% shorter than X/Y-axis optimization when the holes are close to random distribution, which improves the productivity greatly, especially in batch processing cases.

**Keywords:** Fluid dispensing; path planning; ant colony optimization; traveling salesman problem

## 1 引言

微电子封装工业中, 表面贴装技术(SMT)由于具有提高电路的密度, 改善电子性能; 同时可以降低工艺成本、提高产品质量和可靠性等特点而得到了越来越广泛的应用。SMT 中通常采用点胶来固定元件, 对元件进行密封等。由于使用点胶技术对元器件进行涂敷加固的过程需要胶枪相对于待加工部件做大量的移动, 因此移动距离的长短在很大程度上影响了加工的效率, 所以减少胶枪的移动距离对提高加工效率具有

重要意义。传统点胶工艺中所采用的路径优化算法与二维数控钻床加工算法<sup>[1]</sup>类似, 具有简单、快速的特点但优化后的路径通常并不是接近最优的。而本文通过引入蚁群优化算法对点胶涂敷路径进行优化可以明显缩短胶枪的移动距离, 进而提高加工效率。

胶枪移动路径优化问题事实上就是一个组合优化问题, 该类问题的解空间具有爆炸性增长的特点, 使得采用传统的数学方法进行求解变得很困难。采用现代人工智能方法比如遗传算法(GA)<sup>[2]</sup>、人工免疫系统

① 收稿时间: 2009-04-05

(AIS)算法<sup>[3]</sup>、人工神经网络(ANN)<sup>[4]</sup>以及蚁群优化(ACO)算法<sup>[5]</sup>等可以给我们解决这类问题提供一种新思路。由于蚁群优化算法在解决组合优化问题上比其它方法更具优越性<sup>[6]</sup>，所以本文采用蚁群优化算法来对点胶涂敷路径进行优化。

本文组织如下：第2节描述点胶胶枪加工路径优化问题；第3节详细介绍蚁群优化算法；第4节通过仿真实验对蚁群优化和传统的针对X/Y方向优化的结果进行比较；最后给出结论和展望。

## 2 问题描述

点胶胶枪加工路径优化问题可以抽象为以下问题：给出一系列待涂敷加工的孔  $V=v_1, v_2, v_3, \dots, v_n$  定义孔  $v_i$  和  $v_j$  之间的欧几里德距离为  $d_{ij}(d_{ij} \geq 0)$ ，其中  $1 \leq i, j \leq n$  并且  $i \neq j$ ； $S$  被定义为代表问题解的孔的组合，要求找到一个最小代价可行解  $S^*$  使得：

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \quad (1 \leq i, j \leq n \text{ and } i \neq j) \quad (1)$$

其中：
$$x_{ij} = \begin{cases} 1, & \text{如果边}(v_i, v_j) \text{属于优化后的路径} \\ 0, & \text{否则} \end{cases}$$

要求每个孔被加工且仅加工一次最后回到初始位置。

这实质上是一个旅行商问题(TSP)。TSP问题最优解的精确解法的计算代价很高，因为它们必须(或者隐式地)考察所有解并从中找出最优者。幸运的是我们可以通过启发式算法(heuristics)在合理的时间内以高概率得到很接近最优解的近似解。与其它启发式算法相比ACS具有更好的性能<sup>[6]</sup>，因此本文采用了基于蚁群系统的算法来对点胶胶枪加工路径进行优化。

## 3 蚁群优化

蚁群优化(ACO)算法是基于蚂蚁觅食行为的一类算法，它的第一个成员为蚂蚁系统是由Coloni, Dorigo等在文献[7,8]中提出的，后来在文献[9,10]中正式提出为一个新的元启发式算法术语：蚁群优化(Ant Colony Optimization, ACO)。该算法的特点是它显式使用之前构造的解成分，进而加速求解过程。

### 3.1 ACO的数学模型

蚁群优化的数学模型首先被应用于TSP问题。在TSP问题中给定一组城市以及各城市间的路径长度，而问题即为找到访问且仅访问每个城市一次的最短闭

合路径。

在TSP问题中我们可以设定各顶点间边的长度与各顶点所代表的城市间的距离成比例并将信息素值以及启发因子与图的边相关联。信息素量在计算过程中会被及时更新，它代表了对蚁群过去经验的累积，而启发因子则是与问题相关联的值，在TSP问题中它通常被设置为各城市间距离的倒数。

我们定义第  $k$  只蚂蚁从城市  $i$  到城市  $j$  的转移概率如下：

$$P_{ij}^k = \begin{cases} \frac{([\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta)}{\sum_{l \in J_k} [\tau_{il}]^\alpha \cdot [\eta_{il}]^\beta} & \text{if } j \in J_k \\ 0 & \text{否则} \end{cases} \quad (2)$$

其中， $\text{Tabu}_k$  是一个禁忌列表存储着第  $k$  只蚂蚁已经访问过的城市，存在于禁忌列表中的城市不允许蚂蚁再次访问，而  $J_k = \{N - \text{Tabu}_k\}$  则代表处在城市  $i$  位置的蚂蚁  $k$  还需要访问的城市。 $\alpha$  和  $\beta$  则是控制信息素和可见度相对重要性的参数， $\alpha > 0$  且  $2 \leq \beta \leq 5$ 。 $\alpha$  值过大将会增强系统的初始随机扰动，因此  $\alpha$  的典型值是  $1^{[11]}$ 。 $\eta_{ij} = 1/d_{ij}$  为启发因子，其中  $d_{ij}$  为城市  $i$  和城市  $j$  之间的距离。 $\tau_{ij}$  为边  $(i, j)$  上的信息素量。当算法中的蚂蚁结束它们的旅程之后每一个边  $(i, j)$  上的信息素量将会被通过如下公式进行更新：

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho \Delta \tau_{ij} \quad (3)$$

其中  $\rho \in (0, 1]$  被称作蒸发速率参数。信息素蒸发可以避免信息素的过量积累进而导致算法陷于停滞，它有助于蚁群部分“遗忘”过去所走过的路径，从而促使蚁群在搜索空间中进行新的探索。被定义如下：

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k \quad (4)$$

其中代表第  $k$  只蚂蚁在边  $(i, j)$  每单位长度上所释放的信息素量。在比较流行的 ant-cycle 模型中它由下式给出：

$$\Delta \tau_{ij}^k = \begin{cases} Q/L_k, & \text{如果蚂蚁 } k \text{ 选择边}(i, j) \\ 0, & \text{否则} \end{cases} \quad (5)$$

其中  $Q$  为一个常量而  $L_k$  则代表第  $k$  只蚂蚁所走过的路径的长度。在 ant-cycle 算法中信息素在所有蚂蚁完成它们的路径探索之后被更新，由于其采用全局信息，在求解 TSP 问题时可以防止残留信息素的无限积累等问题。

ACO 算法比较耗时，因此我们需要为算法设置一

些终止条件以使得算法在一段合理的时间内结束。ACO 的典型终止条件可以是达到设定的迭代次数、解质量满足需要或者 CPU 完成设定的运算时间,当然也可以在算法停滞时间超过一定限度的时候将其终止。

### 3.2 Ant Colony System

蚂蚁系统(Ant system)对于小规模 TSP 问题(30 个城市左右)的求解是比较有效的,但是对于更大规模的问题的求解将会因为太过耗时而变得不可行<sup>[6]</sup>。因而出现了一个性能更好的算法: Ant Colony System(ACS)。在 ACS 中状态转移规则如下:

$$j = \begin{cases} \operatorname{argmax}_{\mu \in J_k} \left\{ [\tau_{i\mu}]^\alpha \cdot [\eta_{i\mu}]^\beta \right\}, & \text{if } q \leq q_0 \\ J, & \text{otherwise} \end{cases} \quad (6)$$

其中  $q$  是一个均匀分布于  $[0, 1]$  范围内的随机数,  $q_0$  为满足  $0 \leq q_0 \leq 1$  的参数,而  $J$  是一个根据(2)式中的概率分布规则所产生的随机变量。此状态转移规则倾向于选择由较短的边连接以及边上信息素量较大的节点。参数  $q_0$  决定了开发最优解或探寻搜索空间的相对重要性,可以通过调节  $q_0$  的值以在二者之间取得平衡。

在 ACS 中只有全局最优蚂蚁(即构造最短路径的蚂蚁)被允许释放信息素。信息素量通过采用以下全局更新规则进行更新:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij} \quad (7)$$

其中:

$$\Delta\tau_{ij} = \begin{cases} Q/L_g, & \text{if } (i, j) \in \text{global-best-tour} \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

而  $L_g$  为全局最优路径的长度。全局更新规则的目的在于向较短的路径释放较多的信息素。

在 ACS 中还有一条局部更新规则其目的在于防止某条边被所有的蚂蚁访问而不利于探索新路径: 每当一条边被某一蚂蚁选择后它的信息素量将被如下局部信息素更新规则所改变:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\tau_0 \quad (9)$$

其中  $\tau_0$  为参数。局部信息素更新规则是受真实蚂蚁所释放的信息素的挥发而产生的。

## 4 仿真结果

### 4.1 X/Y 方向优化

在点胶工艺中比较常用的路径优化方法是将待加

工点按照 X 轴或者 Y 轴方向进行优化。按 X 向路径法优化即将所有待加工的孔按照其坐标值进行排序,排序原则是按 X 值从小到大的顺序,当 X 值相等时,按 Y 值排序, Y 值的排序原则取决于上一点的 Y 值,当上一点的 Y 值大于等于当前点的 Y 值时,则按从大到小的顺序,否则按从小到大的顺序<sup>[1]</sup>。Y 方向路径法优化算法同 X 方向路径优化算法类似。

这种路径优化算法的思路比较简单,而且优化过程可以在很短的时间内完成,不过它更适用于当大多数待加工点都排列在同一条直线上的时候。而当待加工点呈接近随机分布或者很不规则的时候, X/Y 方向优化的效果则不明显。因此在这种情况下引入了蚁群优化算法对其进行改进。

为了将 X/Y 方向优化结果和基于蚁群优化算法的结果进行对比,我们随机产生 30 个点,并将这两种算法分别应用于这些待加工点。随机生成的点分布如图 1 所示:

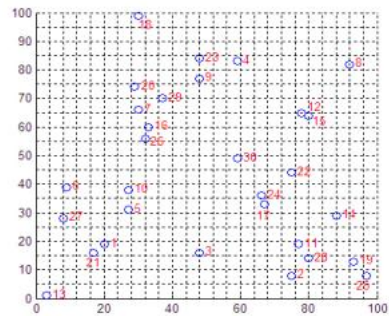


图 1 随机生成的 30 个点的分布图

为了简单起见,在图 1 的点分布图中每个点的横纵坐标都是介于 0 和 100 之间的随机整数。

通过对其进行 X 方向优化后我们得到如下点加工顺序,如图 2 所示:

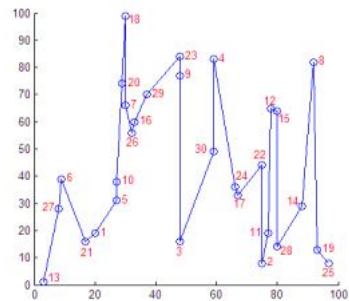


图 2 X 方向优化后加工顺序

事实上，当最后一个点 25 被加工后胶枪需要回到初始位置即点 13 所在处，为了清晰起见该加工路线在上图中并未标出。计算得出总加工路线长度为 811.2784。

通过对其进行 Y 方向优化后我们得到如下点加工顺序图，类似地，为了清晰起见，点 18 和点 13 之间的加工路线未被画出，如图 3 所示。计算可得 Y 方向优化后总加工路径长度为：1011.1。

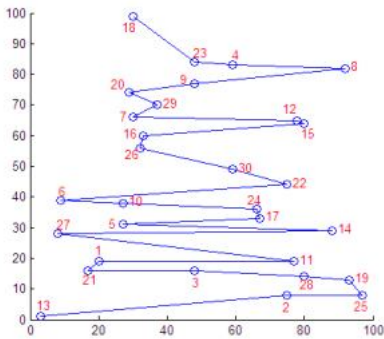


图 3 Y 方向优化后加工顺序

#### 4.2 采用 ACS 算法进行优化

按照 X/Y 方向对待加工点进行优化的效果并不理想，因为这两种算法产生的路径长度仍然较长，由于算法比较简单、快速，对于处理器性能较低、速度要求较高的场合比较实用。由于点胶过程通常都是一次优化然后大批量加工，因此即便采用较为复杂的算法在上位机上进行一次离线优化然后将优化结果发送给下位机进行批量点胶加工，仍然可以大幅度提高加工效率。所以可以通过采用 ACS 算法来进行优化，采用 ACS 进行优化后得到如下加工顺序如图 4 所示：

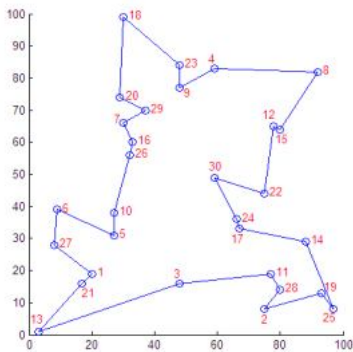


图 4 采用 ACS 算法进行优化所得加工路线

(NC\_Max=200, Ant\_Num =10,  $\alpha=1, \beta=5, \rho=0.01, Q=100$ )

其全长为 464.3316，长度为 X 方向优化长度的 57.23% 是 Y 方向优化路径长度的 45.92%。

#### 4.3 更进一步比较

前面仅仅对一个例子的优化结果进行比较，下面我们又随机产生了 15 组样本，每个样本包含 30 个待加工孔，而且每个孔的坐标都是位于 0 和 100 之间的随机数。通过对上述样本分别进行 X/Y 方向优化和 ACS 优化后产生的结果如下图 5 所示，其中 X 轴代表样本编号，Y 轴代表优化后的路径长度：

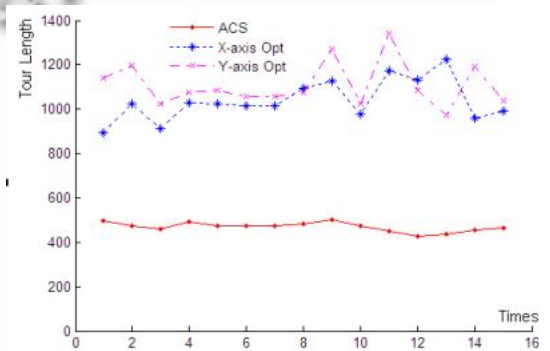


图 5 15 个随机样本的优化结果比较

从上图我们可以看出 ACS 算法的优化效果与 X/Y 方向优化效果相比具有明显优越性，尤其是当待加工点接近随机分布时，通过 ACS 算法优化后的路径长度普遍为 X/Y 方向优化长度的 40%~50%，优化效果十分明显。

#### 4.4 研究展望

在前面的例子中每个例子只包含 30 个随机点，不过对于大型的 TSP 问题而言 30 个点远远不够的。通常情况下点胶装置需要加工的点远不止 30 个，在这种情况下采用标准的蚁群优化算法效率仍较低，而采用 ACS 以及 3-Opt 路径优化算法来求解则高效得多。还可以采用类似候选列表的数据结构，对于一个给定的城市 i，在该列表中包含 cl 个最近邻的城市且以距离递增形式排列，在这种情况下，蚂蚁应当首先向存在于候选列表中的城市移动，当且仅当候选城市列表中的每一个城市都无法访问时才选择其它城市。文献[6]显示这种改进的 ACS 算法可以较好地解决城市数目多至 1500 的大型 TSP 问题。



我们还可以将 ACS 算法与传统 X/Y 方向优化算法进行结合,在待加工孔较多的情况下用 X/Y 方向优化算法以提高速度而在待加工孔少的情况下采用 ACS 算法以生成更短的路径来提高加工效率。也可以在待加工点呈线性规律分布时采用 X/Y 方向优化算法,而当点接近随机分布或很不规则的时候采用 ACS 算法进行优化。

## 5 结论

在点胶工艺中胶枪加工的路径长度很大程度上影响了生产效率,因此减少胶枪移动距离具有重要意义,而通常采用的 X/Y 方向路径优化算法虽然具有快速、简单的特点但只实用于大多数加工点都分布在同一直线上时的情况,而对于待加工点接近随机分布时的优化效率则不明显,这种情况下,通过采用蚁群优化的算法对待加工点的加工顺序进行离线优化可以明显缩短加工路径长度至传统算法所生成路径长度的 50% 左右,进而明显提高生产效率。尽管 ACS 算法比较相对耗时,但对于在上位机上离线优化一次然后发送到下位机进行批量加工的情况仍然是十分有效的。

### 参考文献

- 1 王恒.二维多孔数控钻床加工路径的生成及优化.机械科学与技术, 2002,21(3):463—464.
- 2 周明,孙树栋.遗传算法原理及应用.北京:国防工业出版社, 1999.
- 3 肖人彬,王磊.人工免疫系统:原理、模型、分析及展望.计算机学报, 2002,25(12):1281—1293.
- 4 周济,查建中,肖人彬.智能设计.北京:高等教育出版社, 1998.
- 5 Bonabeau E, Dorigo M, Theraulaz G. Inspiration for optimization from social insect behavior. Nature, 2000,406(6791):39—42.
- 6 Dorigo M, Gambardella LM. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation, 1997,1(1).
- 7 Dorigo CM, Maniezzo V. Distributed optimization by ant colonies. Proc. of the ECAL'91, European Conference on Artificial Life. Elsevier Publishing, Amsterdam, 1991.
- 8 Dorigo M, Maniezzo V, Colomi A. The ant system: an autocatalytic optimizing process, Technical Report, TR 91—016, Politecnico di Milano, 1991.
- 9 Dorigo M, DC. Gianni. Ant colony optimization: a new meta-heuristic. Proc. of the Congress on Evolutionary Computation. Piscataway, 1999. 1470—1477.
- 10 Dorigo M, Caro GD, Gambardella LM. Ant algorithms for discrete optimization. Artificial Life, 1999, 5(2):137—172.
- 11 Dorigo M, Stutzle T. Ant Colony Optimization. Peking: Tsinghua University Press, 2007. 18—70.