2016 数学建模校赛

题目 B

印制电路板焊锡

学　　校: 山西大学

队　　员: 李恩宾

张红若

张彪

指导教师: 李顺勇

2016 年 9 月 3 日

印制电路板焊锡

摘　要

针对电路板焊锡过程中点胶机的涂锡膏顺序问题，本文将电路板中所有的位点抽象为图中的顶点，将位点间的连边抽象成顶点间的连边，位点间的距离抽象为边上的权值，构建赋权完全图。通过遗传算法和贪婪算法，求解赋权完全图中的最小哈密尔顿圈，即求得最优焊锡顺序和相应的路径长度。[1]

关键词：遗传算法 旅行商问题 贪婪算法 蚁群算法

目录

1 问题的提出

焊锡是印制电路板生产过程中的一步后续处理，目的是把贴片元件焊接到电路板表面。人们使用一台装有锡膏点胶机的数控（计算机数字控制，简称 CNC）机械，将锡膏涂布于电路板表面的特定位置。现有一个电路板，大小为 $300mm \times 180mm$，共有 280 个位点需要点涂锡膏。

问题一：求解点胶机从 1 号位点开始对这 280 个位点进行焊锡，最后回到 1 号位点的最优焊锡顺序，并给出相应路径长。

问题二：不预先设定起点和终点，求解对 1 万张这种类型的电路板进行焊锡的最优顺序，并给出相应路径长，这里假设每张电路板安装点胶机所需时间和所处位置都是相同的。

2 模型假设

1. 假设点胶机的初始位置位于电路板起始位点的正上方；
2. 假设电路板上的位点大小均一致；

3 符号说明

4 模型的建立

4.1 问题分析

通过对电路板焊锡问题的分析，我们将焊锡过程转化为赋权完全图中求最小哈密尔顿圈问题，即 TSP 旅行售货员问题。TSP 问题是典型的 NP-hard 问题，不存在多项式时间算法求其最优解。我们通过建立贪婪算法和遗传算法，对电路板上的位点进行遍历，最终回到起点，得到最优焊锡顺序，求得相应的路径长。

将问题看做定义在图 $G=(V,E)$ 上的 TSP 问题,其中 $V=1,2,\ldots,n$，$E=(i,j),i,j\in V$ 图中节点之间的距离矩阵为

$$D = \begin{bmatrix} 0 & d_{1,2} & \dots & d_{1,n} \\ d_{2,1} & 0 & \dots & d_{2,n} \\ \dots & \dots & 0 & \dots \\ d_{n,1} & \dots & \dots & 0 \end{bmatrix} \tag{1}$$

4.2 问题一

4.2.1 蚁群算法

蚁群算法是通过模拟蚂蚁群体觅食行径而提出来的一种模拟进化算法[3]。蚁群算法结果如图 1，总长度为 2946.44724482

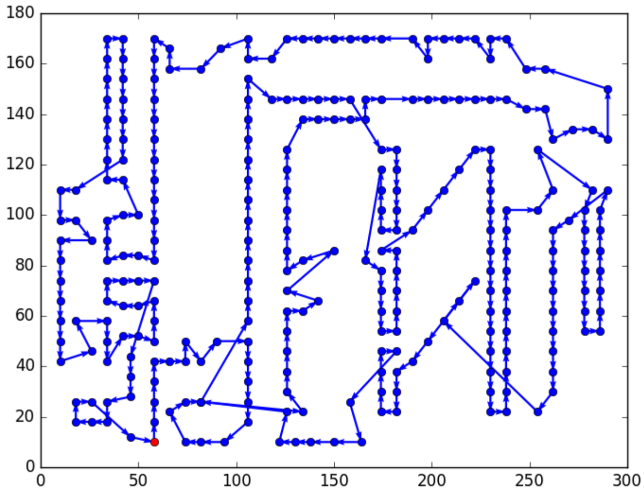

图 1: 改进贪婪算法得到的结果

4.2.2 遗传算法

遗传算法是模仿生物进化过程的一种算法，它以自然选择和遗传学中的复制、变异和交叉等自然规律为理论依据。利用 GA 算法求解问题，首先需要将模型的解以适当的方式进行编码，并根据模型的目标建立个体适应度函数，然后随机生成初始种群（即一组初始的可行解），再重复对种群进行选择、交叉、变异等遗传操作直到种群成熟，停止优化[?]。对

于 NP 困难问题智能算法是有效的的选择，我们选用遗传算法对问题一进行建模求解，并且根据实际题目的要求对遗传算法进行微调。

在一般的遗传算法中，种群个体 j 结构如图 2所示，变异算子是指让种群中某个个体的某些基因突变的算子，由于本题限定了初始位置与

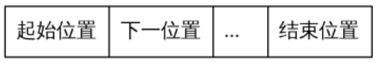

图 2: 算法中的个体

结束位置所以个体基因中的起始位与结束位是固定的不可以被突变算子改变，在种群迭代中其过程如如图 3所示。

同样的对于算法中的交叉算子，它是使有效基因遗传下去的主要算子，在交叉过程中同样应该遵守第一位与最后一位不可算在交叉过程中的原则，如图 4所示。

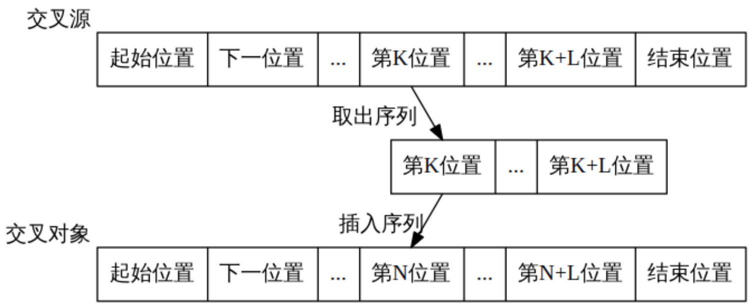

图 3: 交叉算子

图 4: 变异算子

使用上述算法结构对于本问题所给的数据进行种群演化，经过长时间算法运行，得到结果如图 5，总长度为 2775.273329

4.2.3 改进贪婪算法[2]

贪婪算法是一种构建型启发式算法，其原理是求出任意两位点间欧式距离，建立图 G 的邻接矩阵 D 并为其赋值，其中 $d_{i,j}$ 表示位点 x_i 与

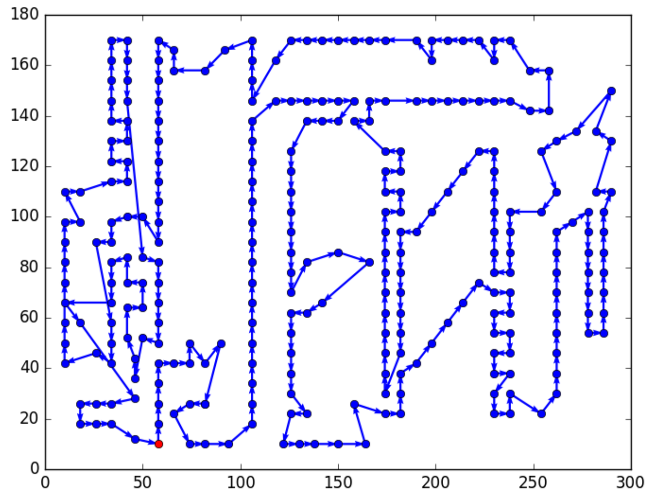

图 5: 遗传算法结果

x_j 间的距离。将距离矩阵中两位点间距离按从小到大的顺序排列，从最短边开始向图 G 中添加，直至添加到图 G 成为一条回路为止。在添加过程中算法遵循以下原则：

1. 该节点的链接边数小于 2 。

2. 链接的边不会形成边数小于 n 的回路。

但贪婪算法在寻找过程中并没有主动使起点与终点回合，所以会造成最后返回路径非常长,如图 6所示,这时所得到的总长度为 2861.03928238。

分析贪婪算法过程可知，算法最后产生的跨度主要由于算法本身没有确定边的拼接方向，依照这种思路对贪婪算法进行改进。根据 Held 与 Karp 提出的 Held-Karp 模型[4]，我们使用一个向量 $\pi = (\pi_1, \pi_2, \ldots, \pi_k)$ 对贪婪算法的距离矩阵 D 进行改造，如式 2。

$$D' = \begin{cases} d_{i,j} = d_{i,j} - \pi_i - \pi_j & i \neq \mathrm{j} \\ d_{i,j} = M & i = \mathrm{j} \end{cases} \tag{2}$$

其中 M 为一个足够大的数，π_k 由式 3构造，其中 α 为比例系数，比

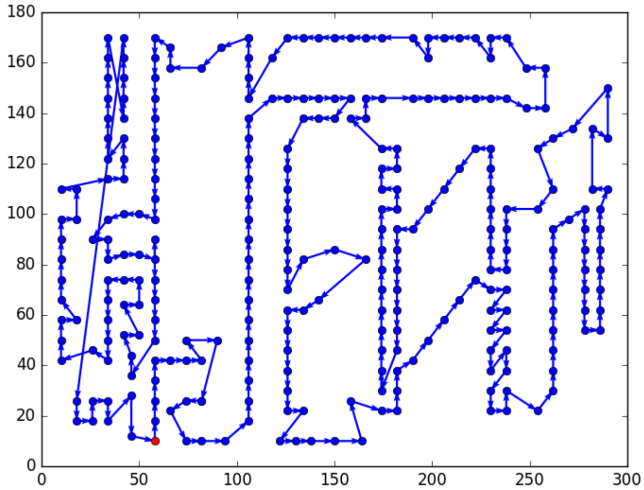

图 6: 贪婪算法得到的结果

例系数越大算法就越倾向于先添加外围的点。

$$\begin{cases} \pi = (\pi_1, \pi_2, \ldots, \pi_k), \pi_k = \alpha \cdot (len_k - \overline{len}) \\ len_k = \sqrt{(x_0 - x_k)^2 + (y_0 - y_k)^2} \\ \overline{len} = \frac{\sum_{k=0}^{k=n} len_k}{n} \\ x_0 = \frac{\sum_{k=0}^{k=n} x_k}{n}, y_0 = \frac{\sum_{k=0}^{k=n} y_k}{n} \end{cases} \tag{3}$$

根据改进贪婪算法得到最终结果如图 7所示,总长度为 2791.27883397。

4.3 问题二

4.3.1 贪婪算法

遗传算法可以获得问题的满意解但是其运行过程十分漫长不适用于题目要求，所以我们利用贪婪算法设计了一个较快获得满意解的算法。

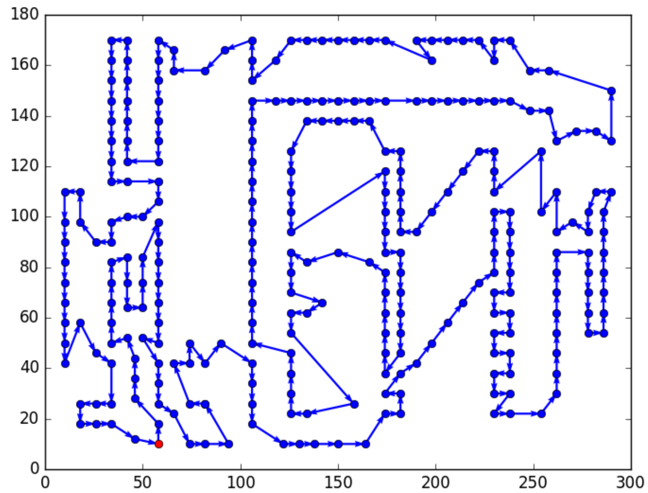

图 7: 改进贪婪算法得到的结果

4.3.2 原理

参考文献

[1] 楼建华. 数学建模与数学实验 [J]. 黑龙江高教研究, 2003(3): 126–127.

[2] 饶卫振, 金淳. 基于求解 TSP 问题的改进贪婪算法 [J]. 运筹与管理, 2012, 21(6): 1–9.

[3] 士勇, 永强, 研. 蚁群算法及其应用 [M]. [S.l.]: 哈尔滨工业大学出版社, 2004.

[4] HELD M, KARP R M. The traveling-salesman problem and minimum spanning trees[J]. Operations Research, 1970, 18(6): 1138–1162.