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，

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。图1展示了使用的数控机械。现有一个电路板，大小为300毫米C:\Users\Lenovo\AppData\Local\Temp\ksohtml\wpsD7DF.tmp.pngC:\Users\Lenovo\AppData\Local\Temp\ksohtml\wpsD7E0.tmp.jpg180毫米，共有280个位点需要点涂锡膏，这280个位点的横纵坐标在表2中给出。

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。

模型的假设

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；

符号说明

顶点

边

距离

总权值

模型的建立

问题分析

题目中针对电路板焊锡问题，给出了焊锡位点及过程，要求我们建立适当的数学模型并求解，得到最优焊锡顺序及路径长度，使得电路板上的280个位点都被点涂锡膏。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，并通过计算最优焊接路径的总权值，求得相应的路径长。

针对问题一，

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。

针对问题二，

模型建立

画个图

贪婪算法（GR算法）求解原理

由公式( )求出任意两位点间距离，建立距离矩阵Dij=dij，其中dij表示位点xi与xj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。

遗传算法求解原理

问题一

以电路板上的位点为一个图的顶点，对任意两个位点进行连边，两位点之间的距离作为边上的的权值，构造一个赋权完全图G(V,E)。

设题目中所给的位点看作一个图中的顶点，其坐标分别用X1,X2,,,X280表示,任意位点两两连边，且设两位点Xi，Xj间的距离用dij表示，则

dij=

即两位点连线的边上权值为dij。

模型的求解

结果分析和检验

模型展望

焊锡是在焊接线路中连接电子元器件的重要加工工艺，广泛应用于电子工业、家电制造业、汽车制造业、维修业和日常生活中。焊锡是印制电路板生产过程中的一项重要处理步骤，即将锡膏涂布于电路板表面的特定位置的操作工艺。焊锡的具体操作如下：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（即电路板上的特定位点）。点胶机以电路板的1号位点为起点，对电路板上的280个位点进行焊锡，最后回到1号位点。以电路板上的位点为一个图的顶点，对任意两个位点进行连边，两位点之间的距离作为边上的的权值，构造一个赋权完全图。对电路板焊锡，即经过电路板上280个位点至少一次，最终回到起点，求最优焊锡顺序，并求出相应路径长的过程，就可以转化为图论问题:在赋权完全图中寻找从1号位点出发，找出一个最小权的哈密尔顿圈，使得遍历所有顶点至少一次再回到1号位点，使得总权值最小，此即TSP旅行售货员问题。

TSP问题是一个组合优化问题，也是NP-hard问题，即不存在多项式时间算法，不能用[精确算法](http://baike.baidu.com/view/5136411.htm)求解。 事实上，TSP问题的求解算法有很多，比如：回溯法、线性规划法、分支限界法等，除此之外，还可以用近似算法求解，但是由近似算法求得问题的解一般为问题的非最优解、近似解。对此，我们采取如下几种策略：(1)用启发式方法求解；(2) 寻求行之有效的[近似算法](http://baike.baidu.com/view/2489123.htm)。因此，我们采用启发式算法—贪婪算法和仿生算法--遗传算法解决此问题。

模型评价及改进方向

优点：

1.采用图论模型，将问题转化为TSP旅行售货员问题，贪婪算法和遗传算法

缺点：

* + - 1. 贪婪算法是问题在某种意义上的局部最优解算法；

不能保证求得的最后解是最佳的。由于 贪心策略总是采用从局部看来是最优的选择，因此 并不从整体上加以考虑。 (2)贪心算法只能用来求某些最大或最小解 的问题。从前面的讨论中，找零钱问题要求得到最 小数量采用贪心算法是可行的，但是在另外一个求 解权值最小路径时采用贪心算法得到的结果并不 是最佳。 (3)贪心算法只能确定某些问题的可行性范

采用近似算法

模型展望

参考文献

[1]戴三,陈恭洋,周云才等.旅行商问题(TSP)算法比较[J].计算机与数字工程,2013,41(9):1445-1447.DOI:10.3969/j.issn1672-9722.2013.09.011.

该问题可以定义在图 G ＝（V，E）上，其中 V ＝｛１，２，…，n｝ （n≥３）， E ＝（i，j），i，j∈V。 １，２，…，n 代表城市或称为点；（i，j）、dij分别代表城市 i 至城市 j 的路和距离。 显然，图 G 为 具有 n 个点且任何点之间均有边的全图，距离矩阵 D 是 n 阶方阵可表示为 D ＝（dij）n ×n， dij （i≠j） dij ＝M （i ＝j） ， ＴＳＰ 问题就是在全图 G 中找一个过所有点且总边长最短的一个简单圈（或称 Ｈａｍｉｌｔｏｎ 回路）。 一个 ＴＳＰ 问题的解（路径）可表示为 n 个城市｛１，２，…，n｝的环状排列。 令 S 为存放了一个解的一维 数组，那么此解的旅行总距离（路径总长度）可由公式（１）计算： L（S） ＝（∑ n －２ i ＝０ dS［ i］ S［ i＋１］ ） ＋dS［ n－１］ S［０］ （１） 显然，使得 L（S） 取得最小值的一维数组 S 中存放的排列为 ＴＳＰ 问题的最优解［１４］