

基于改进蚁群算法的贴片机贴装过程优化

王 君, 罗家祥, 胡跃明

(华南理工大学自动化科学与工程学院精密电子装备制造教育部工程中心, 广州 510640)

摘 要: 为提高贴片机生产效率, 对贴片机贴装过程中的元器件拾取贴放顺序进行优化, 提出一种改进的蚁群算法。该算法将分散搜索算法中的参考解集引入到 cunning 蚁群系统中, 使其参与到蚁群对解的搜索, 从而提高算法的全局搜索能力。实验结果表明, 该算法在多数情况下能够搜索到优于传统 cunning 蚁群系统的解。

关键词: 表面贴装技术; 贴片机; cunning 蚁群系统; 分散搜索

Mount Process Optimization for Chip Mounter Based on Improved Ant Colony Algorithm

WANG Jun, LUO Jia-xiang, HU Yue-ming

(Engineering Research Center for Precision Electronic Manufacture Equipments, Ministry of Education,

College of Automatic Science and Engineering, South China University of Technology, Guangzhou 510640, China)

【Abstract】 In order to promote the production efficiency of chip mounter, the component mounting sequence is optimized. An improved Ant Colony Algorithm(ACA) is proposed to solve the problem. In this algorithm, the reference solution set in scatter search is introduced into cunning ant colony optimization algorithm to take part in the ants' route selection, aiming to improve the global search ability of the algorithm. Experimental results are compared with those of the traditional cunning ant colony system.

【Key words】 Surface Mount Technology(SMT); chip mounter; cunning ant colony system; scatter search

DOI: 10.3969/j.issn.1000-3428.2011.14.087

1 概述

随着电子制造业的发展和表面贴装技术(Surface Mount Technology, SMT)在 PCB 组装生产中的广泛应用, 如何提高现有 PCB 组装生产线的生产效率变得越来越重要。因此, 作为瓶颈设备的贴片机, 其贴装效率的提高对于提高整个生产线的生产效率有着重要意义。提高贴片机的贴装效率主要是对喂料器的位置分配和元器件的拾取贴放顺序两方面进行优化^[1]。在以往的研究中, 最近邻算法、2-opt 算法、遗传算法等多种启发式算法都曾被用于求解该问题, 并取得了良好的效果。但随着电子制造技术的发展, 电子元器件的种类和数量越来越大, 对贴装的速度要求越来越高。因此, 有必要研究更有效的优化算法来提高贴片机的生产效率。针对喂料器位置确定情况下元器件拾取贴放顺序优化的问题, 本文引入一种具有参考解集的 cunning 蚁群系统, 以优化贴片机。

2 问题描述

根据结构的不同, 贴片机分为多种类型。其中, 拱架式贴片机具有性能稳定、贴装精度高等优点, 是普遍应用的贴片机类型之一, 其结构如图 1 所示。

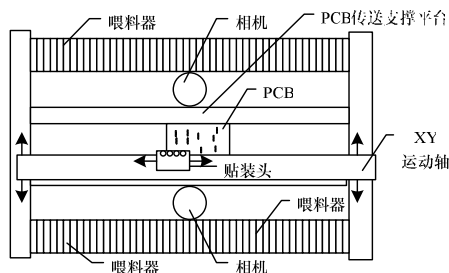


图 1 贴片机结构

完成一块 PCB 的贴装要经过多个贴装循环。在每个贴装循环中, 贴装头先移动到喂料槽拾取一组元器件, 经由相机对元器件进行识别和校正后, 移到 PCB 上方将元器件贴装到相应的贴装点上。由于贴装头每次移动的平均速度差别不大, 贴装时间的长短主要取决于贴装头走过的路径长度, 而路径及其长度又是由元器件的拾取贴放顺序决定的。

为了方便描述这一问题, 设置变量如下:

$J=\{1, 2, \dots, n\}$, 表示 PCB 上需要贴装的元器件集合。

$G=\{1, 2, \dots, m\}$, 表示贴装元器件类型/喂料器集合。

$P=\{1, 2, \dots, n\}$, 表示 PCB 上的贴装点集合。

$g(j)$, 表示元器件 j 对应的喂料器。

$p(j)$, 表示元器件 j 在 PCB 上对应的贴装点。

D_i , 表示第 i 个贴装组内元器件的数量。

F , 表示贴装头上吸嘴的数目, 为一定值。

C , 表示贴装组个数。

$A=\{A_1, A_2, \dots, A_C\}=\{A'_1, A'_2, \dots, A'_C\}=\{A''_1, A''_2, \dots, A''_C\}$,

表示贴装组集合。

A_i , 表示第 i 个贴装组的元器件集合, $A_i=\{A_i(1), A_i(2), \dots, A_i(D_i)\}$, $i=1, 2, \dots, C$ 。

A'_i , 表示第 i 个贴装组的元器件拾取顺序, $A'_i=(A'_i(1), A'_i(2), \dots, A'_i(D_i))$, $i=1, 2, \dots, C$ 。

A''_i , 表示第 i 个贴装组的元器件贴放顺序, $A''_i=(A''_i(1),$

基金项目: 国家自然科学基金资助项目(60835001, 60804053); 教育部博士点基金资助项目(200805611065)

作者简介: 王 君(1984—), 男, 硕士研究生, 主研方向: 调度优化; 罗家祥, 博士; 胡跃明, 教授、博士生导师

收稿日期: 2011-03-22 **E-mail:** wangjuncut@163.com

$A''_i(2), \dots, A''_i(D_i))$, $i=1, 2, \dots, C$ 。

l_i , 表示在第 i 个贴装循环中, 贴装头从 $p(A''_{i-1}(D_i))$ 运动到 $p(A''_i(D_i))$ 的距离。

l^a_i , 表示在第 i 个贴装循环中, 贴装头从 $p(A''_{i-1}(D_i))$ 移动到 $g(A'_i(I))$ 的距离。

l^b_i , 表示在第 i 个贴装循环中, 贴装头从 $g(A'_i(I))$ 运动到 $g(A'_i(D_i))$ 的距离。

l^c_i , 表示在第 i 个贴装循环中, 贴装头从 $g(A'_i(D_i))$ 运动到 $p(A''_i(I))$ 的距离。

一块有 n 个元器件的 PCB 需要通过 C 个贴装循环贴装完成, 则 $C=\lceil n/F \rceil$ 。则在每个贴装循环中贴装组的元器件数量 D_i 为:

$$D_i = \begin{cases} F & \text{if } i = \{1, 2, L, C-1\} \\ n-F \cdot (C-1) & \text{if } i = C \end{cases} \quad (1)$$

显然, 在第 i 贴装循环中, 贴装头走过的距离 l_i 为:

$$l_i = l^a_i + l^b_i + l^c_i + l^d_i \quad (2)$$

完成整个 PCB 的贴装, 贴装头所走过的总的距离 L 为:

$$L = \sum_{i=1}^C l_i = \sum_{i=1}^C (l^a_i + l^b_i + l^c_i + l^d_i) \quad (3)$$

在贴装过程中, 首先由 A 确定待贴装元器件的分组, 然后贴装头根据拾取顺序 A' 和贴放顺序 A'' , 循环完成整块 PCB 的贴装。则需要解决的问题就是找到最好的 A 、 A' 和 A'' , 使 L 最短。

3 算法求解

由上可知, 在一个贴装循环内, 只有当该贴装循环内的 A_i 确定后才能得到 A'_i 和 A''_i , 为此本文采用按贴装循环逐步求解的方式使用改进的 cunning 蚁群系统解决该问题。

3.1 传统的 cunning 蚁群系统

在传统 cunning 蚁群系统(cunning Ant colony System, cAS)^[2]中, 一只蚂蚁第 t 次迭代产生的解称为 c-ant, 第 $t-1$ 次迭代产生的解被称为 d-ant。在蚂蚁构造 c-ant 时, 根据 cut-point 方法截取 d-ant 中的一部分给 c-ant, 其他部分则根据基于信息素的解构造方法来求取。

c-ant 构造完成后, 取两解中较优者作为该只蚂蚁在第 t 次迭代中的解。这一克隆特性的引入对防止算法过早停滞, 搜索更好的解起到了很好的效果。

3.2 引入参考解集的 cunning 蚁群系统

为进一步提高 cunning 蚁群系统的优化效果, 将分散搜索(Scatter Search, SS)算法^[3]中的参考解集(Reference Set, RefSet)引入到 cunning 算法中。使解的搜索具有导向性, 一方面对优秀的解空间进行更深入的局部寻优, 另一方面 RefSet 中的多样性解集又引导蚂蚁对其他可能的解空间进行探索, 避免了出现过早收敛。

本文算法中引入的 RefSet 由 b 个解组成, 分为最好解集(解中优化结果最好的前 b_1 个解)和多样性解集(与最好解集中所有解的欧几里德距离之和最大的 b_2 个解)($b=b_1+b_2$)。

3.3 求解过程

该算法的总体求解过程如下:

Step1 构造网络图并初始化参数; 令迭代次数 $t=1$ 。

Step2 初始化参 RefSet。

Step3 对于蚁群中的每只蚂蚁完成如下过程:

Step3.1 取 RefSet 中的第 i 个解作为 d-ant, 使用具有克隆特性的蚂蚁搜索算法构造 c-ant。

Step3.2 用求得的解 c-ant 更新参考解集。

Step4 信息素更新。

Step5 如果迭代结束 $t > \text{MAXLOOP}$ (MAXLOOP 为最大迭代次数)或算法停滞, 算法结束。否则令 $t=t+1$, 转到 Step3 继续搜索新解。

3.3.1 蚂蚁搜索网络图的构造

将 n 个元器件作为节点, 连接这些节点, 由此组成了一个网络图。信息素分布在各节点之间的路径上。

网络图中的路径距离与循环内的元器件及其位置相关, 在不确定最终解的情况下, 其长度是动态变化的估算值。设 $\Psi = \{\psi_1, \psi_2, \dots, \psi_{\text{index}}\}$ 为未选择的元器件集合, 其中 index 为集合中元器件的个数。则对于评估元器件 ψ_j 与之前已确定的循环中元器件的距离估算方法如下:

当选取 A_i 中第 $h(h=1)$ 个元器件时, 待评估元器件 ψ_j 与已确定循环组中元器件的距离评估值 $d(\psi_j, A_i, h)$ 为:

$$d(\psi_j, A_i, h) = \begin{cases} \text{distance}(0, g(\psi_j)) & \text{if } i=1 \\ \text{distance}(p(A''_{i-1}(D_i)), g(\psi_j)) & \text{if } i>1 \end{cases} \quad (4)$$

即当 $i=1$ 时, $d(\psi_j, A_i, h)$ 为设定原点与 ψ_j 对应的喂料器 $g(\psi_j)$ 之间的距离; 当 $i>1$ 时, $d(\psi_j, A_i, h)$ 为第 $i-1$ 次循环中的最后一个贴装点 $p(A''_{i-1}(D_i))$ 与 ψ_j 对应的喂料器 $g(\psi_j)$ 之间的距离。

当选取 A_i 中第 $h(h=2, 3, \dots, D_i)$ 个元器件时, ψ_j 与已确定循环组中元器件的距离评估值 $d(\psi_j, A_i)$ 为:

$$d(\psi_j, A_i, h) = \frac{1}{h-1} \sum_{k=1}^{h-1} (\text{distance}(g(A_i(k)), g(\psi_j)) + \text{distance}(p(A_i(k)), p(\psi_j))) \quad (5)$$

其中, $\text{distance}(g(A_i(k)), g(\psi_j))$ 指元器件 ψ_j 对应的喂料器与 A_i 中已选定的第 k 个元器件 $A_i(k)$ 对应喂料器之间的距离; $\text{distance}(p(A_i(k)), p(\psi_j))$ 指元器件 ψ_j 在 PCB 上的贴装点与 $A_i(k)$ 在 PCB 上的贴装点之间的距离。

当贴装组的元器件确定时, 该循环内元器件的拾取贴放顺序问题可以作为一个以组内元器件及其对应拾取点为城市节点的带约束的 TSP 问题来求解, 约束条件为, 当拾取点对应的城市节点完全选取后才开始选取贴装点对应的城市节点。该问题可以用穷举法获得一个贴装循环内部的最好解。

3.3.2 参考解集的初始化与更新

(1) 参考解集的初始化

将 RefSet 中解的数量 b 设置为蚁群中蚂蚁的数量 $\text{NumAnt}(b=\text{NumAnt})$, 并将所有路径的信息素设定为一个任意的值。蚁群首先迭代一次。这次迭代中蚂蚁使用传统的基于信息素的方法构造解。

(2) 参考解集的更新。

完成 RefSet 的初始化以后, 蚁群在后面的迭代中, 每只蚂蚁搜索完成后, 将根据解 c-ant 的评价值和与最优解的欧几里德距离对 RefSet 进行更新。

3.3.3 信息素的初始化与更新

(1) 信息素的初始化,

在完成参考集的初始化后, 得到历史最好解的路径长度 L^{sb} 。采用最大最小蚁群算法^[4]中的方法求解最大信息素 τ_{max} 并将其作为初始值。

(2) 信息素的更新

采用 MMAS 中的信息素更新方法: 对全部路径进行信息素的衰减, 但仅对历史最好解所在的路径增加信息素。从而强化最好的路线, 保证在求解大规模的问题时能够更好搜索到有效的解。

3.3.4 具有克隆特性的蚂蚁搜索算法

参考解集和信息素初始化以后, 蚁群将借鉴 cAS 中解的

构造方法来构造解。对于蚁群中的第 i 只蚂蚁 ant_i , ($i=1,2,\dots, \text{NumAnt}$), 其构造新解 c-ant 的步骤如下:

Step1 取 RefSet 中的第 i 个解作为 d-ant。

Step2 新解 c-ant 的构造与 cAS 中基本相同, c-ant 从 d-ant 中保留部分解, 保留的部分将被放在 c-ant 的前端。保留部分在选取时, 起始点为随机选取, 保留段的长度 len 为参数 γ 控制的可调值, 满足: $len=\lceil \gamma \times n \rceil$, 其中 $0<\gamma<1$ 。

Step3 c-ant 的余下的部分通过传统蚁群算法中的基于信息素和路径长度的选择方法。

在该问题中, 由于元器件是成组拾取和贴放的, 截取部分的起始点应从 C 个元器件组的第 1 个元器件中随机选取。

4 实验

以实验室研制的 4 头拱架式贴片机为原型, 通过生产中使用的一批元器件数量和种类不同的 PCB 的优化结果验证算法的有效性。

求解过程中的参数统一设定为: 信息启发式因子 $\alpha=1$; 期望启发式因子 $\beta=2$; 信息素消逝因子 $\rho=0.5$; 选择阈值 $q_0=0.9$; $\text{NumAnt}=n$, $b=n$, $b1=b2=b/2$; 信息素更新参数 $p_{\text{best}}=0.005$; $\text{MAXLOOP}=800$; $\gamma=0.4$ 。将算法与传统的 cunning 蚁群系统进行了比较, 部分结果如表 1 所示。

由表 1 可知, 引入参考解集的 cunning 蚁群系统优化结果在不同程度上优于传统 cunning 蚁群系统。同时, 对于元器件数量和种类不同、贴装点布局不同的 PCB, 引入参考解集的 cunning 蚁群系统都能够在不调节参数的情况下取得较好的解, 表明该算法具有较强的参数鲁棒性和适应性。

表 1 算法有效性验证结果

PCB 标号	元器件数量/个	元器件类型数/种	传统算法	改进算法
			路径长度/mm	路径长度/mm
1	49	19	11 924.1	11 780.4
2	50	24	10 499.5	10 466.0
3	60	4	14 223.8	13 949.9
4	72	6	16 478.9	16 192.7
5	117	24	25 896.9	25 361.0
6	162	70	43 483.5	41 805.6
7	171	83	44 048.9	42 310.9
8	189	16	39 747.0	38 820.9
9	223	27	61 515.5	60 570.0
10	276	44	63 556.9	59 456.8

5 结束语

针对贴片机生产效率优化问题, 本文提出一种引入参考解集的 cunning 蚁群算法。通过将其嵌入到 SMT 生产线的优化软件中, 参与完成 SMT 生产线的前期调度工作。在不改变贴片机机械特性的条件下, 有效缓解了贴片机这一瓶颈设备对整个生产线贴装效率的影响。

参考文献

[1] Ball M. Sequencing of Insertions in Printed Circuit Board Assembly[J]. Operations Research, 1988, 36(2): 192-201.

[2] Tsutsui S. Ant Colony Optimization with Cunning Ants[C]//Proc. of the 9th Int'l Conf. on Parallel Problem Solving. [S. l.]: IEEE Press, 2006.

[3] Glover F. A Template for Scatter Search and Path Relinking[M]. Berlin, Germany: [s. n.], 1998.

[4] Stutzle T, Hoos H H. Max-min Ant System[J]. Future Generation Computer Systems, 2000, 16(8): 889-914.

编辑 陈 文

(上接第 255 页)

5 实验结果与分析

本实验使用部分 ISCAS'89 基准电路。实验结果如表 1 所示, 其中, 第 1 列表示电路名称, 第 2 列表示触发器的个

数。第 3 列至第 5 列给出在划分扫描子链条数 N 分别为 2、3、4 时本文方法的测试应用时间, 单位是时钟周期。这里用测试向量扫描移入时间近似表示测试应用时间。

表 1 测试应用时间的降低

电路名	FFs	本文方法(时钟周期)			相对文献[3]降低的比例/(%)			相对文献[4]降低的比例/(%)		
		N=2	N=3	N=4	N=2	N=3	N=4	N=2	N=3	N=4
S838	32	3 938	3 207	2 426	37.173	39.639	51.246	42.977	25.245	42.675
S953	29	1 853	1 477	972	30.936	39.591	62.090	51.694	43.017	56.742
S1196	18	1 719	1 322	1 052	19.371	16.646	28.775	13.182	15.903	8.362
S1238	18	1 763	1 417	1 104	16.287	37.467	9.508	-22.943	12.260	20.690
S1423	74	3 464	3 562	2 605	37.041	36.472	60.089	59.954	37.901	28.825
平均	-	-	-	-	28.162	33.963	42.342	28.973	26.865	31.459

由实验结果可以看出, 经过应用本文提出的改进方法, 在一定程度上降低了测试应用时间, 在 $N=2$, $N=3$, $N=4$ 的情况下, 相对文献[3]分别减少了 28.162%, 33.963%, 42.342%。相对文献[4]分别减少了 28.973%, 26.865%, 31.459%。

6 结束语

本文提出一种在扫描链阻塞结构下, 通过对不相容测试向量进行压缩, 实现降低测试应用时间的方法。实验结果表明, 该方法简单, 且不需要增加任何硬件, 并能进一步地降低测试应用时间。

参考文献

[1] 易东严, 尤志强. 基于树形压缩器的低测试数据量方法[J].

计算机工程, 2010, 36(9): 249-251.

[2] Whetse1 L. Adapting Scan Architectures for Low Power Operation[C]//Proc. of International Test Conference. Trento, Italy: [s. n.], 2000.

[3] You Zhiqiang, Iwagaki T, Inoue M, et al. A Low Power Deterministic Test Using Scan Chain Disable Technique[J]. IEICE Transactions on Information and Systems, 2006, E89-D(6): 1931-1939.

[4] 江招生, 尤志强, 张大方, 等. 一种基于 TSP 问题的扫描链阻塞技术低费用测试方法[J]. 哈尔滨工业大学学报, 2009, 41(1): 176-179.

编辑 陈 文