

MCM 基板互连测试的单探针路径优化研究*

许川佩^{1,2}, 许君华², 莫玮², 李智²

(1. 西安电子科技大学 机电工程学院, 陕西 西安 710071; 2. 桂林电子工业学院 电子工程系, 广西 桂林 541004)

摘要: 针对 MCM 基板互连测试所采用的单探针技术, 本文提出一种基于蚁群算法的单探针路径优化算法, 通过设定合适的规则, 引导探针的移动, 缩短探针移动的距离, 达到减少测试时间提高 MCM 生产效率的目的。从基于 MCM 标准电路的仿真结果看, 采用蚁群算法得到的探针测试路径长度远远优于其它算法所得到的。

关键词: MCM 基板; 互连测试; 单探针; 蚁群算法

中图分类号: TP274; TN407 **文献标识码:** A

1 引言

目前多芯片组件 (Multi-Chip Module, MCM) 基板互连测试通常采用单探针或双探针法进行短路和断路故障测试。

这种探针测试是在飞针测试床上进行的, 测试的时间主要花费在探针的移动上, 为了减少移动定位时间, 除了提高探针移动速度外, 减小探针移动的总距离是一种有效的方法。对于 MCM 基板, 其表面焊区的布局及网络分布总存在一条使探针移动最短的路径。如何根据 MCM 基板焊区和网络的分布情况求解出探针运动的最短或较短路径是探针测试 (飞针测试) 技术的关键。本文将围绕单探针测试提出一种探针移动路径优化的新方法。

已有学者采用传统求解货郎担问题 (Traveling Salesman Problem, TSP) 的算法来解决探针测试路径的优化问题^[1~3], 但这些算法的求解过程只限于小规模模拟, 通常采用的实验对象是几十个网络的电路, 而 MCM 的发展速度很快, 高密度、大规模已经是 MCM 发展的趋势。此外, 已有的探针测试路径优化算法在寻找路径的时候, 主要是基于对局部优化的考虑而进行寻优工作的, 这就使得算法在全局优化方面表现出一定的局限性, 为了能够得到较优的探针测试路径, 算法必须进行遍历方式的试探工作, 以此来寻找较好的焊区序列, 最终减少测试路径的长度, 这就使得算法的运行时间会随着 MCM 规模的增大急剧增加。基于对算法运行时间和生成路径长度的考虑, 本文提出一种采用蚁群算法来解决探针测试路径优化问题的方法。

2 蚁群算法求解单探针路径的数学模型

求解探针测试的路径就是在待测焊区集合中寻找一个序列, 使得测试路径的总距离最短。虽然最短路径是必然存在的, 但实际寻找到的路径往往不是一条最短的路径, 而是一条较短路径。求解探针路径时首先由 CAD 软件提供的电路网表 (netlist) 得到 MCM 基板的网络互连以及各焊区的坐标数据, 这样就可以求得两个焊区间的欧几里距离 (Euclidean Distance)——探针在两个焊区间移动的距离。令: $I = \{I_1, I_2, \dots, I_N\}$ 是 MCM 基板互连网络的集合, 每个互连网络 I_i ($1 \leq i \leq N$) 包含一个焊区集合 $P_j = \{P_1, P_2, \dots, P_{n_j}\}$, 其中 n_j 是 I_i 的焊区数, N 是 MCM 基板网络的个数。探针路径优化问题的解就是一组焊区的序列 $T(t_1, t_2, \dots, t_N)$, 其中 $t_i \in P_j$, $i = 1 \dots N$, $j = 1 \dots n_j$ 。探针需要移动的距离是:

$$C = \sum_{i=1}^{N-1} \delta(t_i, t_{i+1}) \quad (1) \quad \text{其中 } \delta(t_i, t_{i+1}) \text{ 是 } t_i \text{ 到 } t_{i+1} \text{ 的欧几里距离。}$$

* 收稿日期: 2004-07-23 修订日期: 2004-10-10
基金项目: 国家自然科学基金资助项目 (60266001)

2.1 定义

本文为了叙述的需要,给出求解单探针路径所涉及的名词以及求解过程的定义:边是两个焊区间的连线,其长度就是两个焊区的欧几里距离;路径是一组边的有序集合;完整路径是蚂蚁在访问完所有焊区后得到的焊区结点序列;子路径是蚂蚁在没有得到完整路径之前访问过的焊区结点序列;待访问焊区集合是蚂蚁还没有访问过的焊区节点集合。

单探针路径的求解过程定义如下:设 $V = \{c_1, c_2, \dots, c_n\}$ 是一组焊区的集合, $A = \{(r, s) : r, s \in V\}$ 是一组边的集合, $\delta(r, s)$ 是 r, s 两点的欧几里距离。该问题的已知条件是各个焊区的坐标,求解的目标是得到一条最短的路径序列,该序列中包含了每一个焊区。本文研究的对象属于对称性 TSP, 即 $\delta(r, s) = \delta(s, r)$ 。

蚂蚁算法求解探针路径时,除了从已知条件中求得的 $\delta(r, s)$ 外,还有一个重要的因子 $\tau(r, s)$ —信息素, $\tau(r, s)$ 在算法运行时被蚂蚁更新。对于对称性 TSP, 有 $\tau(r, s) = \tau(s, r)$ 。

2.2 蚂蚁算法 (AS) 求解单探针路径的数学模型

蚂蚁算法^[4]是模拟自然界蚂蚁寻找食物时在所经过的路径上留下一一种挥发性的物质(称为信息素),从而引导后续蚂蚁走最短路径提出的一种算法。

蚂蚁算法 (Ant System, AS) 求解单探针路径的过程^[5]如下:每只蚂蚁采用基于概率的状态转移规则来选择下一步要访问的焊区,直至走完一条完整路径。当所有的蚂蚁都走完了一条完整路径后,算法将采用全局信息素更新规则来更新现有路径上的信息素。在这个过程中,有一小部分信息素挥发,而有蚂蚁经过的路径上就会有新的信息素增加。在下次蚂蚁选择路径时,更新后的信息素将会影响蚂蚁选择该边的概率,使得全局较短路径上走过的蚂蚁增多。不断地重复这两个过程,最终蚂蚁将寻找到一条合适的路径。

状态转移规则如式 (2) 所示,等式给出了蚂蚁 k 在焊区 r 时选择焊区 s 的概率。

$$P_k(r, s) = \begin{cases} \frac{[\tau(r, s)] \cdot [\eta(r, s)]^\beta}{\sum_{u \in J_k(r)} [\tau(r, u)] \cdot [\eta(r, u)]^\beta}, & \text{如果 } s \in J_k(r) \\ 0, & \text{否则} \end{cases} \quad (2)$$

其中, τ 是信息素, $\eta(r, s) = 1/\delta(r, s)$, $\eta(r, u) = 1/\delta(r, u)$, $J_k(r)$ 是蚂蚁 k 在焊区 r 时待访问焊区集合, β 是决定信息素与距离两个因子重要程度的参数。通过 (2) 式,可以使得距离较短而信息素较高的路径被选择的机会增大,该状态转移规则兼顾了全局路径长度和当前要选择的边的长度。

在蚂蚁算法中,全局更新规则按照 (3) 式定义:

$$\tau(r, s) \leftarrow (1 - \alpha) \cdot \tau(r, s) + \sum_{k=1}^m \Delta \tau_k(r, s) \quad (3)$$

其中, $\Delta \tau_k(r, s) = \begin{cases} \frac{1}{L_k}, & \text{如果 } (r, s) \in J_k \\ 0, & \text{否则} \end{cases}$, J_k 是第 k 只蚂蚁走过的边的集合, L_k 是第 k 只蚂蚁访问过的

完整路径的长度, m 是蚂蚁数, $0 < \alpha < 1$, 是信息素挥发因子。

2.3 蚁群算法 (ACS) 求解单探针路径的数学模型

针对 MCM 基板互联的复杂程度,本文在求解探针路径时,借鉴了 Dorigo Marco^[6]提出的蚁群 (Ant Colony System, ACS) 算法。这是一种分布式算法,利用一组相互协作的主体(即蚂蚁)并行寻优,通过信息素实现间接协作和全局通讯。ACS 与 AS 的不同之处主要有三点:1) 状态转移规则为进行新路径搜索和依据已有的信息搜索之间提供了一种平衡方法;2) 全局更新规则仅用于最优路径的边;3) 当蚂蚁寻找路径时使用局部更新规则。

采用 ACS 算法求解探针路径的过程如下:首先初始化 m 只蚂蚁到 n 个焊区上,作为路径的起始点。

接下来每只蚂蚁重复使用状态转移规则, 寻找下一步要走过的焊区, 在已经走过的子路径上, 蚂蚁根据局部信息素更新规则释放信息素。当全部蚂蚁都走完了全部焊区后, 采用全局信息素更新规则来更新路径上的信息素。ACS算法所采用的状态转移规则、全局信息素更新规则以及局部信息素更新规则的定义如下。

2.3.1 ACS状态转移规则

ACS状态转移规则如式(4)所示:

$$s = \begin{cases} \max_{u \in J_k(r)} \{[\tau(r, s)] \cdot [\eta(r, s)]^{\beta}\}, & \text{如果 } q \leq q_0 \\ S, & \text{否则} \end{cases} \quad (4)$$

在式(4)中, s 是蚂蚁 k 在焊区 r 处将选择的焊区, q 是 $[0, 1]$ 中均匀分布的随机数, 而 q_0 是一常数 ($0 < q_0 < 1$), S 是根据等式(2)选择出的焊区。这种基于等式(2)和(4)的状态转移规则称为伪随机概率规则, 这种规则通常会让距离较短而且信息素较大的路径以更大的概率被选中。参数 q_0 决定了是搜索新路径还是依据信息作选择, 当 $q \leq q_0$ 时, 蚂蚁将直接选择信息素和距离因素之积最好的边, 否则仍按照式(2)来选择将要访问的焊区。这种做法在降低了算法求出要访问的边所花费的运算时间的同时, 与AS相比, 也可以避免因局部优化而影响全局优化的结果。

2.3.2 ACS全局信息素更新规则

在ACS中, 只有获得全局最优路径的蚂蚁释放信息素, 再与状态转移规则结合, 使得搜索更加直接: 蚁群在算法当前的迭代中是在最好路径的相邻范围内进行搜索。全局信息素更新规则如式(5):

$$\tau(r, s) \leftarrow (1 - \alpha) \cdot \tau(r, s) + \alpha \cdot \Delta\tau(r, s) \quad (5)$$

其中, $\Delta\tau(r, s) = \begin{cases} (L_{gb})^{-1}, & \text{如果 } (r, s) \in \text{全局最好路径} \\ 0, & \text{否则} \end{cases}$, $0 < \alpha < 1$, 是信息素挥发因子, 它决定信息素挥发的速度, L_{gb} 是蚂蚁在一次完整的路径寻找过程中寻找到的最短路径的距离, 即全局最优路径的长度。

在AS中, 全局更新规则倾向于给较短的路径与更多的信息素, 而式(5)只允许走出最好路径的蚂蚁留下信息素, 这使得蚂蚁能排除其它信息素的干扰而更快地找到距离较短的路径。

2.3.3 ACS局部更新规则

更新规则如式(6)所示:

$$\tau(r, s) \leftarrow (1 - \rho) \cdot \tau(r, s) + \rho \cdot \Delta\tau(r, s) \quad (6)$$

其中 $0 < \rho < 1$, 是信息素衰减因子。Dorigo证明当 $\Delta\tau(r, s) = \tau_0$ 时的结果较好, τ_0 是初始时各条边上的信息素。

经过大量的实验表明, 在对MCM基板焊区进行路径优化时, 各种规则中参数的取值如下: $\beta = 2$, $q_0 = 0.99$, $\alpha = \rho = 0.1$, $\tau_0 = (n \cdot L_m)^{-1}$, 其中 n 是网络数, 而 L_m 是采用最临近法则求得的最短完整路径长度。对于网络数小于1000的探针测试路径优化, 蚂蚁数 m 取10时的结果较好, 而当网络数大于1000时, 考虑到算法的复杂度, m 取5比较合适。

3 参数选取仿真试验

在设定单探针的优化规则后, 单探针测试路径优化包括下面三个步骤: 首先从CAD软件中取得电路网表(netlist), 并对网表进行编译。接着计算出每个焊区的坐标及相互间的距离。最后采用蚁群算法来寻找并优化单探针的测试路径。

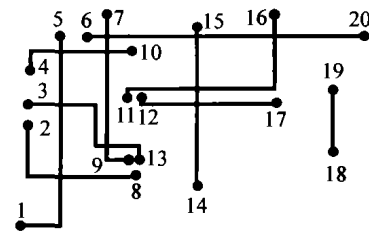


图1 单探针测试的路径

针对图1所示电路, 采用ACS算法来进行寻找和优化基于单探针的测试路径实验, 在实验过程中, 程序每次运行时都连续进行10次路径搜索, 设定 $\alpha = 0.1$, 调整蚂蚁的数量 m 进行了若干次实验, 实验结果如表1所示。

表 1 当 $\alpha=0.1$ 时的优化结果

m	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$	$i=7$	$i=8$	$i=9$	$i=10$
1	248.86	234.69	234.69	257.12	244.28	234.69	297.06	228.37	297.06	244.28
2	203.79	265.21	265.21	237.53	248.86	187.35	214.84	185.28	243.87	232.23
5	216.55	184.22	237.85	202.14	202.14	181.43	210.64	202.14	181.43	181.43
10	187.35	185.28	214.5	185.28	178.03	185.28	185.28	190.54	184.22	184.22
20	184.22	179.37	181.43	179.37	179.37	179.37	181.43	184.22	184.22	179.37

mm

从表 1 看到, 蚂蚁数与优化结果有着明显的因果关系, 当蚂蚁数增多时, 优化结果就会向着更好的趋势发展, 且收敛速度加快, 这种变化过程如图 2 所示。

图 2 中分别给出了当蚂蚁数量为 1、5、10、20 时的路径优化过程的曲线, 曲线的横轴是算法取得完整路径的次数, 而纵轴是所得到优化路径的长度。比较 $m=10$ 和 $m=20$ 时的结果发现, 并不是增多蚂蚁数就一定能提高优化结果。这表明, 蚂蚁留下的信息素可以传递距离信息, 同时也存在一定的干扰性, 使得算法难以寻找到较好的路径, 这种干扰性随着蚂蚁数的增大而加剧。另外, 不管蚂蚁的数量为多少, 每条优化曲线都出现了一定的波动性, 这种波动性随着蚂蚁数的增多而降低。由于干扰性和波动性的相互制约使得蚂蚁数量的选取有了一定的限制, 本文将蚂蚁数设定为 5 到 10 只, 具体数量的设定根据 MCM 基板互联的复杂度来确定。在此次实验中, 蚁群算法求得的最优路径如图 3 虚线所示, 矢量

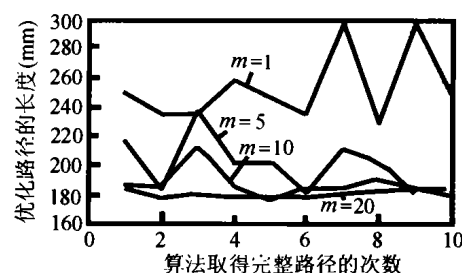


图 2 蚂蚁数量与优化结果

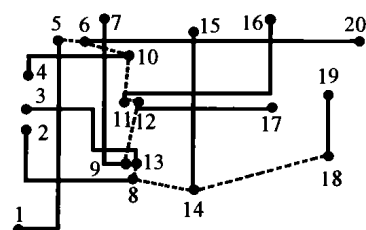


图 3 蚂蚁算法优化的测试路径

表示为 [5, 6, 10, 11, 12, 9, 13, 8, 14, 18], 这也是该问题的最优解。

下面讨论优化结果与 α 的关系, 作为信息素的挥发系数, α 是蚂蚁算法中最重要的参数之一, 它将影响蚁群算法的收敛、波动性与干扰性, 直至影响最后的优化结果。由于 α 所产生的影响是和其他参数共同作用产生的, 而且随着优化目标的不同而不同, 所以很难确定 α 的最终取值。

下面进行一组实验来说明 α 的取值与优化结果的关系。在本次实验中, $m=10$ 固定不变, 而通过改变 α 来求得优化结果的曲线, 如图 4 所示。虽然无法真正地确定 α 对优化结果的影响函数, 但从图中仍能看出各次测量曲线受 α 影响的变化趋势,

当 α 增大时, 优化结果就出现较大的波动, 这是由于蚂蚁在寻找路径时更注重局部优化而产生的, 而当 $0.05 < \alpha < 0.1$ 时, 优化结果的波动性减小, 并且收敛趋势明显。 α 所引起的波动性是和其他参数共同作用产生的, 所以本文将蚁群算法的波动性函数定义为:

$$\lambda = f(\alpha, m, n, Pn) \quad (7)$$

其中, λ 是波动性函数, m 是蚂蚁数, 而 n 和 Pn 分别是网络数与焊区数, 它们是标识优化对象复杂度的变量。虽然目前还没有文献给出 λ 的详细内容, 但在对其性质了解的基础上, 人们已经可以取得一些经验值来设定 α , 并且取得了较好的效果。

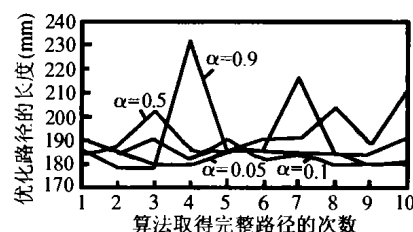
4 基于蚁群算法的单探针测试路径优化仿真

采用以 ACS 算法为核心的探针测试路径优化方法对 mcc1-75 MCM 基板互联的探针测试路径进行优化试验, mcc1-75 是 MCNC 组织建立的 MCM 基准电路, 该电路共有 806 个网络, 1946 个焊区。试验所用的计算机硬件配置为: PIV 1.8G/512M DRAM。

采用 ACS 算法生成 mcc1-75 的单探针测试路径后, 将其与 Pendurkar R^[1] 等人采用线性方法进行试验的数据进行比较, 结果详见表 2。Pendurkar R 等人的算法是在 SUN Ultra-2 SPARC 工作站上进行运算的。从运行结果可以看到采用蚁群算法得到的测试路径长度

表 2 优化结果比较

算法	mcc1-75.net	
	长度(mil)	CPU 时间(s)
插入法	6569.22	6.43
改进插入法	3919.30	18.75
移动法	3910.30	2366.92
改进移动法	3886.85	5881.81
ACS	2718	3844.10

图 4 α 与优化结果的关系

要远远短于其它算法所得到的, 所得到的测试距离比移动法^[1]的短30%, 比插入法^[1]的短59%。

5 结论

本文提出了基于蚁群算法的MCM基板测试单探针路径优化模型。当ACS算法用于MCM基板互联测试的探针路径优化时, 蚂蚁之间通过一种简单的传递信息的方式—信息素来完成寻优工作, 本文将该算法与单探针测试的规则相结合, 针对MCM基板互联测试的特点, 得到了一种新颖的路径优化算法, 并针对MCM基准电路mcc1-75 等作了仿真实验。从得到的结果看, 探针测试路径长度远小于其它传统算法得到的结果。

参考文献:

- [1] Rajesh Pendurkar, *et al.* Single-Probe Traversal Optimization for Testing of MCM Substrate Interconnections [J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1999, 8(18): 1178-1191.
- [2] Kim Bruce C, *et al.* Probe Scheduling Algorithm for MCM substrates [A]. *IEEE International Test Conference (TC)* [C]. 1999. 31-37.
- [3] Chou Nan-Chi, *et al.* Dynamic Probe Scheduling Optimization for MCM Substrate Test [J]. *IEEE Trans. Components*, 1994, 5(17): 182-189.
- [4] Goss S, Aron S, Deneubourg J L, *et al.* Self-organized shortcuts in the argentine ant [J]. *Naturwissenschaften*, 1989, 76: 579-581.
- [5] Dorigo M, Maniezzo V, Colomi A. The ant system: Optimization by a colony of cooperating agents [J]. *IEEE Trans. Syst, Man, Cybern. B*, 1996, 26: 29-41.
- [6] Dorigo M, *et al.* Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem [J]. *IEEE Transactions on Evolutionary Computation*, 1997, 4(1): 53-66.

作者简介: 许川佩 (1968-), 女, 副教授, 博士生, 主要研究方向为集成电路测试理论与技术; 许君华 (1978-), 男, 硕士, 主要研究方向为计算机辅助测试; 莫玮 (1956-), 男, 教授, 博士生导师, 现为中国电子技术标准化研究所所长, 主要研究方向为自动测试总线与系统、IP 核测试技术、集成电路测试理论与技术; 李智 (1965-), 男, 教授, 主要研究方向为自动测试总线与系统、集成电路测试理论与技术。

Research on single-probe traversal optimization for testing of MCM substrate interconnections

XU Chuan-pei^{1,2}, XU Jun-hua², MO Wei², LI Zhi²

(1. School of Mechano-Electronic Engineering, Xidian University, Xi'an 710071, China;

2. Dept. of Electronic Engineering Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: For the single-probe technology used for testing of MCM substrate interconnections, this paper presents a single-probe Traversal Optimization approach based on ant colony system algorithm. By finding suitable rules to pilot the probe to move, it can minimize the total distance traveled, so as to decrease the test time and to increase the MCM test throughput finally. Experiments on benchmark MCM netlists with single probe traversal confirm that our traversal optimization algorithm based on ACS has an increasing advantage over other algorithms in the total distance traveled.

Key words: MCM substrate; interconnection test; single-probe; ant colony system algorithm