

基于蚁群算法的测试集优化

俞龙江, 彭喜源, 彭 宇

(哈尔滨工业大学自动化测试与控制系, 黑龙江哈尔滨 150001)

摘 要: 电路集成度和复杂度的不断增加使电路故障诊断变得愈加困难. 其中, 测试集优化问题是电路故障诊断的关键问题之一. 本文以新颖的蚁群算法为基础, 较好地解决了测试集的优化问题, 并通过实验证明了该算法的良好性能.

关键词: 蚁群算法; 测试集优化; 故障诊断

中图分类号: TN919 文献标识码: A 文章编号: 0372-2112 (2003) 08-1178-04

A Test Set Optimization Method Based on Ant Algorithm

YU Longjiang, PENG Xi2yuan, PENG Yu

(Dept. of Automatic Test and Control, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China)

Abstract: The increase of digital circuit integrity and complexity has made fault diagnosis of circuits more and more difficult. The scale of test set has become very large because of redundancy, which costs lots of time and memory unnecessarily. It is important to acquire optimal test set for test application. Test set optimization, which can eliminate the redundancy, is one of key problems in fault diagnosis of digital circuits. Ant colony optimization, a new kind of random optimization algorithm, has become a better alternative to genetic algorithm in some areas. That algorithm has such advantages as less parameters and simple operations, so it is easier to be adopted. We propose a method based on ant colony optimization that solves test set optimization better than classic algorithm or genetic algorithm. The better performance of the proposed method is demonstrated by experimental results.

Key words: ant algorithm; test optimization; fault diagnosis

1 引言

随着大规模集成电路的发展, 电路的复杂度不断增加, 这就使电路测试变得愈加困难. 特别是在电路测试生成后, 电路测试集相当庞大, 其中包含大量的冗余测试矢量, 必须进行测试集优化将冗余测试矢量去除才可对测试集进行后续操作. 因此, 寻求一种新的优化算法以提高优化效果成为迫切需要.

蚁群算法^[1]是近年来出现的一种新颖的仿生算法, 它根据蚂蚁觅食规律建立算法模型, 通过多次迭代获得满足要求的结果. 它是继遗传算法之后出现的又一种全局优化算法. 同遗传算法相比, 蚁群算法概念简单, 需要设置的参数少, 不需要交叉、变异等操作, 而优化结果甚至优于遗传算法. 蚁群算法一经提出便发展迅速, 短短几年就应用到各种组合优化领域, 已成为优化算法中不错的选择. 本文将蚁群算法应用到测试集优化问题, 获得了理想的优化效果, 具有良好的理论价值和现实意义.

2 测试集优化

测试集优化, 又叫做测试集压缩, 是数字电路测试的一个

基本问题, 它通过减少测试矢量, 压缩测试时间来降低 VLSI 测试成本. 因此测试集优化对电路测试相当重要, 尤其是对于基于扫描结构的电路测试, 在这种测试中, 测试时间与测试集尺寸和扫描链中使用的存储单元数目之积成正比. 优化后的测试集可显著减少测试存储空间和测试时间, 对于加快测试进程, 减少测试开销具有重要的现实意义. 目前, 测试集优化方法主要包括: 行列消去法、布尔函数约简法和试探法. 就这三种优化方法而言, 布尔函数约简法优化效果最佳, 因为应用该算法可以获得真正意义上的最优解, 但是随测试矢量数目的增加该算法运算量呈指数增长, 如此庞大的计算量使该算法的实用性大大降低; 试探法优化效果最差, 但由于该算法的运算量最少, 使其仍具有一定的实用性; 相比而言, 行列消去法综合性能最佳, 但随电路规模的增加难以收敛于最优解^[2].

如果一个故障测试集可以测试一个电路的全部故障, 则该故障测试集就叫做该电路的一个完全测试集. 在所有完全测试集中, 测试矢量数目最少的完全测试集叫做最小完全测试集. 显然, 任意一个故障都能在最小完全测试集中至少找到一个测试矢量; 而且最小完全测试集中的任何一个测试矢量能测试的故障中, 至少有一个其他测试矢量所不能测试的故

障. 测试集优化问题就是最小完全测试集的求解问题. 目前已证明测试集优化问题属于 NP 问题^[3].

下面举例说明测试集优化问题. 表 1 中 t_i 表示测试矢量, f_i 表示故障. f_i 和 t_i 对应单元取 1 表示 t_i 可以测试 f_i , 取 0 表示 t_i 不能测试 f_i . 因此, t_1 可测 f_6, f_8 和 f_{10} , 但不能测 $f_1, f_2, f_3, f_4, f_5, f_7$ 和 f_9 . 测试集 $(t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8)$ 是故障集 $F = (f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10})$ 的一个完全测试集, 但不是最优测试集; (t_2, t_3, t_5, t_7) 是 F 的一个最优测试集, (t_3, t_4, t_5, t_7) 也是 F 的一个最优测试集.

表 1 故障测试集的 0-1 表示

测故障	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
t_1	0	0	0	0	0	1	0	1	0	1
t_2	0	0	0	0	1	0	0	0	1	0
t_3	0	1	0	0	0	1	0	1	0	1
t_4	0	0	0	0	1	0	0	0	1	0
t_5	0	0	0	1	0	1	0	1	0	1
t_6	0	0	0	0	1	0	0	0	1	0
t_7	1	0	1	0	0	0	1	0	1	0
t_8	0	0	0	0	0	0	0	0	1	0

3 蚁群算法

蚁群算法(ant algorithm)是一种源于大自然中生物世界的新的仿生类算法^[1], 作为与遗传算法同属一类的通用型随机优化方法, 它根据蚂蚁的集群觅食活动的规律, 建立了一个利用群体智能进行优化搜索的模型, 在一系列困难的组合优化问题求解中取得了卓有成效的结果. 与遗传算法类似, 蚁群算法不需要任何先验知识, 最初随机的选择搜索路径, 随着对解空间的了解, 搜索变得有规律, 并逐渐逼近直至最终达到全局最优解. 蚁群算法对搜索空间的了解是从观察蚁群觅食活动从中启发而建立的机制, 主要包括三个方面:

(1) 蚂蚁的记忆. 一只蚂蚁搜索过的路径在下次搜索时就不会被选择, 由此在蚁群算法中建立 tabu(禁忌)列表来进行模拟;

(2) 蚂蚁利用信息素(pheromone)进行相互通信. 蚂蚁在所选择的路径上会释放一种叫做信息素的物质, 当同伴进行路径选择时, 会根据路径上的信息素进行选择, 这样信息素就成为蚂蚁之间进行通讯的媒介.

(3) 蚂蚁的集群活动. 通过一只蚂蚁的运动很难到达食物源, 但整个蚁群进行搜索就完全不同. 当某些路径上通过的蚂蚁越来越多时, 在路径上留下的信息素数量也越来越多, 导致信息素强度增大, 蚂蚁选择该路径的概率随之增加, 从而进一步增加该路径的信息素强度, 而某些路径上通过的蚂蚁较少时, 路径上的信息素就会随时间的推移而蒸发. 因此, 模拟这种现象从而利用群体智能建立的路径选择机制, 使蚁群算法的搜索向最优解推进.

蚁群算法所利用的搜索机制呈现出一种自催化(autocatalytic behavior)或正反馈的特征, 因此, 可以将蚁群算法模型理解成增强型学习系统(reinforcement learning system)^[4]. 这种

选择不满足马尔可夫性: 某时刻采取的行动只与上一时刻的行动相关, 与前面所有时刻采取的行动无关. 这显而易见, 因为蚂蚁每次选择路径后, 就将该路径存到 tabu 列表中, 选择下一条路径时, 只能在 tabu 列表中不包括的路径中进行选择, 而 tabu 列表正是蚂蚁前面所有时刻采取的行动形成的. 因此, 蚁群算法模型就不是一个马尔可夫决策过程, 也不能理解为动态规划(dynamic program)和蒙特卡罗(Monte Carlo)这样的增强学习算法, 而应理解为一种 Q 学习^[5]. 信息素可理解为 Q 学习中的回报, 这方面还有待深入研究. 下面介绍蚁群算法的基本原理.

图 1 用一个形象的图示说明了蚂蚁群体的路径搜索原理和机制. 假定障碍物的周围有两条道路可从蚁巢到达食物, 即(蚁巢 A → B → D → 食物)和(蚁巢 A → C → D → 食物), 路径的长度分别为 4 和 6. 每只蚂蚁在单位时间内移动一个单位长度的距离, 并在走过的路径上遗留一个单位的信息素. 开始时所有道路上未留有任何信息素. 在 $t = 0$ 时刻, 有 20 只蚂蚁从蚁巢出发移动到 A. 它们以相同概率选择左侧或右侧的路径, 因此有 10 只蚂蚁走左侧, 10 只走右侧. 在 $t = 4$ 时刻, 第一批找到食物的蚂蚁将返回. 在 $t = 5$ 时刻, 两批蚂蚁将在 D 点相遇. 此时 BD 上的信息素数量与 CD 上的相同, 因为各有 10 只蚂蚁选择了相应的道路. 从而有 5 只返回的蚂蚁将选择 BD 而另 5 只将选择 CD. 在 $t = 8$ 时刻, 前 5 只蚂蚁将返回蚁巢, 而 AC、CD 和 BD 上各有 5 只蚂蚁. 在 $t = 9$ 时刻, 前 5 只蚂蚁又返回 A 并且再次进行往左还是往右的选择. 这时, AB 上的信息素数量是 20 而 AC 上是 15, 因此将有较多的蚂蚁选择往左, 从而增加了该路径的信息素数量. 随着这种过程的不断进行, 两条路径上的信息素数量的差距将越来越大, 直到绝大多数蚂蚁都选择了最短的路径. 正是由于一条道路要比另一条道路短, 因此, 在相同的时间区间内, 短的路线会有更多的机会被选中.

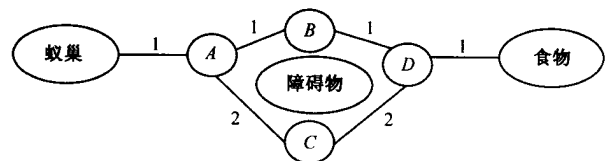


图 1 蚂蚁觅食过程演示

根据以上分析, 可以建立基本的蚁群算法模型. 算法由下面三个公式组成^[1]:

$$P_{ij}^k = \frac{S_{ij}^k \cdot \tau_{ij}^\alpha}{\sum_{j \in A} S_{ij}^k \cdot \tau_{ij}^\alpha} \quad (1)$$

$$S_{ij}(n+1) = Q^* S_{ij}(n) + \sum_{k=1}^m \Delta S_{ij}^k \quad (2)$$

$$\Delta S_{ij}^k = \frac{Q}{F_k} \quad (3)$$

式(1)、式(2)和式(3)中的参数意义为: m 为蚂蚁个数; n 为迭代次数; i 为蚂蚁所在位置; j 为蚂蚁可以到达的位置; A 为蚂蚁可以到达位置的集合; G_j 为启发性信息, 这里为由 i 到 j 的路径的能见度(visibility), 即 $1/d_{ij}$; F 为目标函数, 这里为欧式距离; S_{ij} 为由 i 到 j 的路径的信息素强度(intensity); ΔS_{ij}^k

为蚂蚁 k 由 i 到 j 的路径上留下的单位长度轨迹信息素数量; A 为路径权; B 为启发性信息的权; Q 为路径上信息素数量的蒸发系数; Q 为信息素质量系数; P_{ij}^k 为蚂蚁 k 从位置 i 移动到位置 j 的转移概率.

目前,除了业已得到公认的遗传算法、模拟退火算法、禁忌搜索算法等计算智能算法,新加入这个行列的蚁群算法正在开始崭露头角,为复杂困难的组合优化问题提供了新颖的具有竞争力的求解算法.

4 用于测试集优化的蚁群算法

蚁群算法应用到测试集优化问题时,很容易想到可以让蚂蚁随机分布到各个测试矢量上.每个蚂蚁从当前所在的测试矢量出发,选择还未走过的测试矢量,直到走过的所有测试矢量组成的测试集是完全测试集为止.每次迭代从每个蚂蚁完成的完全测试集中选择矢量数目最少的完全测试集,即最优测试集,经过多次迭代便求得测试集优化问题的最优解.这里需要对蚁群算法最初的模型进行修正:

(1) 蚂蚁的转移概率为:

$$P_{ij}^k = \frac{S_i^A G_{ij}^B}{\sum_{j \in A} S_j^A G_{ij}^B} \quad (4)$$

由于测试集优化问题不同于经典的旅行商问题(TSP)^[11],它的解具有顺序无关性,如得到的测试集 (t_1, t_2, t_3) 和 (t_3, t_1, t_2) 是相同的测试集.因此,放置信息素的对象由从 i 到 j 的路径变为点 j 对应的测试矢量,用 S_j 代替 S_{ij} 表示测试矢量 j 上的信息素轨迹强度.

(2) 启发性信息为:

$$G_{ij}^k = \frac{U_i^k H f_j}{\text{sum}} \quad (5)$$

测试集优化问题的启发性信息可以这样理解:蚂蚁 k 在测试矢量 i 上时,计算可以到达的测试矢量 j 能检测的故障,并从中去除已经过的测试矢量组成的测试集能检测的故障.也就是说,可选的测试矢量 j 包含的已有测试集不能检测的故障的数目越多,启发性信息就越大.其中, U_i^k 为蚂蚁 k 经过的测试矢量组成的测试集不能检测的故障集合, f_j 为测试矢量 j 能检测的故障集合, sum 为故障总数.

(3) 信息素轨迹强度的更新按式(2)和(3)进行,其中的目标函数 F 变为测试集所包含的测试矢量数目.

基于文献[1]的思路,本问题求解的主要步骤如下:

(1) $\text{count} \geq 0$ (count 为迭代次数),每条路径对应的 S_i 和 $v S_j$ 用常数 c 进行初始化,将 m 个蚂蚁随机分布到 n 个测试矢量上;

(2) 将每个蚂蚁的初始位置存放到目前的解集中,对每个蚂蚁 $k(k=1, \dots, m)$,按概率 P_{ij}^k 移至另一测试矢量 j ,将 j 置于当前解集;

(3) 计算每个蚂蚁的目标函数值 F_k ,本问题目标函数值就是测试集所包含的测试矢量数目,将当前的最优解即测试矢量数最少的测试集记录下来;

(4) 按式(2)和(3)更新路径的信息素强度;

(5) $\text{count} \geq \text{count} + 1$, 返回到步骤 1;

(6) 若 count 小于预定的迭代次数且无退化行为(即找到的都是相同解),则返回 2.

算法中的参数选取目前尚无定论,需要根据经验设置.最初选择的参数是通过求解旅行商问题(TSP)库 TSPLIB^[6]中的一些例子得到的: $0 \leq A \leq 5, 1 \leq B \leq 5, 0.1 \leq Q \leq 0.99$ (0.7 左右为佳), $10 \leq Q \leq 10000$. 本文选取 $A=1, B=1, Q=0.9, Q=10$.

算法中蚂蚁的移动可以按照赌轮法进行:

$\sum_{k=1}^{l-1} P_{ij}^k < \text{rand}[0, 1] < \sum_{k=1}^l P_{ij}^k$, 则选择 l . 其中, l 为蚂蚁 k 可选的测试矢量所对应的序号.

5 实验结果

实验利用蚁群算法,对若干已知最优解的测试集和随机生成的测试集进行测试优化.为便于进行比较,这里选取文献[8]中的测试集进行优化.测试集按照惯例,用 @@V@@@F 的形式表示,如 8V10F,对应的测试集包含 8 个测试矢量,检测 10 个故障.

表 2 中待优化的测试集(8V10F~40V70F)为最优解已知的测试集.用布尔函数法可以得到最优解.但是对测试集 30V32F 操作时,庞大的运算量已经使优化难以进行.通过比较蚁群算法和行列消去法的优化结果,发现对已知最优解的测试集,蚁群算法可以获得好于行列消去法的优化效果.

表 2 测试集优化实验结果(已知最优解的测试集)

测试集	蚁群算法(测试矢量数目)	行列消去法(测试矢量数目)
8V10F	3	3
18V20F	3	3
30V32F	5	7
40V70F	6	7

表 3 中待优化的测试集(80V100F~200V400F)为随机生成的测试集.在矢量数较多时,布尔函数约简法显然无能为力,而行列消去法和试探法获得的解不仅不是最优解,相反却与最优解相去甚远,这是因为:试探法的原理导致了测试集不同划分方式会产生不同的结果,当测试集增大时这种现象就会十分突出;行列消去法中对测试集不同的行消去(或列消去)以及消去的顺序都会对优化效果产生直接影响,当测试集增大时这种对操作和顺序的依赖性愈加明显.总之,二者共同之处是测试矢量数较多时无法收敛于最优解.应用蚁群算法对表 3 中的测试集优化后发现,不同的测试集均获得了满意的优化效果,且蚁群算法运行多次后结果相同,这充分证明了蚁群算法具有极强的鲁棒性.

表 3 测试集优化实验结果(随机生成的测试集)

测试集	蚁群算法(测试矢量数目)	遗传算法 ^[8] (测试矢量数目)
80V100F	5	26
100V200F	6	16
200V400F	6	15
80V1000F	7	10

实验中将蚁群算法同文献[8]中使用的遗传算法进行比较,从结果看来,蚁群算法优化效果要明显好于遗传算法。分析表明,遗传算法优化效果比蚁群算法差的原因主要是由于其参数设置,如种群大小、变异概率、交叉概率的变化,会对优化效果产生不同的影响,而适应度函数和罚函数的选取方法也会使遗传算法对同一问题产生不同的结果。文献[8]中就讨论了两种不同的染色体编码方案,结果证明其中一种要比另外一种好得多。此外,遗传算法在求解过程中,有时会出现得到问题的近似最优解而非最优解的现象。与此相比,蚁群算法需设置参数较少,且参数稍作调整或无需做出调整即可从一个问题移植到另一个问题上,算法中不包括交叉、变异等操作,使其应用起来较遗传算法容易得多。

在实际测试集优化应用中往往需优化数千乃至数万的测试集,通过算法可行性分析可知蚁群算法可以在多项式时间内完成求解,但优化效率与具体问题的复杂性及算法实现的软硬件环境有直接关系。

6 结论

蚁群算法虽然在理论上较遗传算法相比尚未成熟,但在实际应用中却展现出较好的优化能力。通过在理论上建立模型和实验证明,蚁群算法能够较好地解决电路故障的测试集优化问题。蚁群算法自身理论体系尚待完善,诸如 A、B、Q、Q 等参数的设置和优化问题将是今后要继续开展的研究工作。可以预见,蚁群算法的理论研究和应用技术的进步将极大地促进蚁群算法的发展,扩展其应用领域。

参考文献:

- [1] Colomi A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies[A]. Proc. 1st European Conf. On artificial life [C]. Paris, France: Elsevier, 1991: 134- 142.
- [2] 杨士元. 数字系统的故障诊断与可靠性设计[M]. 北京: 清华大学出版社, 2000.
- [3] Hideo Fujiwara, Shunichi Toda. The complexity of fault detection problems for combinational logic circuits[J]. IEEE Trans. Comput. June

1982, C- 31(6).

- [4] <http://reinforcementlearning.a2depot.com/Intro.html#OL2>.
- [5] Gambardella M, Dorigo M. AntQ: A reinforcement learning approach to the traveling salesman problem [A]. Proc. of ML295, Twelfth Intern. Conf. on Machine Learning [C]. Morgan Kaufmann, 1995. 252- 260.
- [6] Reinelt G. TSPLIB: a traveling salesman problem library[J]. ORSAJ. On Computing, 1991, 3(4): 376- 385.
- [7] Mark Allen Weiss. Data Structures and Algorithm Analysis (2nd Edition) [M]. Addison Wesley&Pearson, 1993.
- [8] 王小港. 遗传算法在 VLSI 设计自动化中的应用研究[D]. 2001.
- [9] Duivivier D, Preux P, Talbi E G. Parallel genetic algorithms for optimization and application to NP-complete problem solving [A]. Int. Workshop on Combinatorics and Computer Science [C]. Brest, France, 1995.
- [10] Elghazali Talbi, Olivier Roux, Cyril Fonlupt, Denis Robillard. Parallel ant colonies for combinatorial optimization problems [A]. Job Scheduling Strategies for Parallel Processing: IPPS '95 Workshop [C]. Springer LNCS 949, 1995.

作者简介:



俞龙江 男, 1976 年 10 月生于哈尔滨, 哈尔滨工业大学自动化测试与控制系, 博士研究生, 主要研究方向为故障诊断, 计算智能, 数字电路测试。



彭喜源 男, 1961 年 12 月生于内蒙古四子王旗, 哈尔滨工业大学自动化测试与控制系教授, 博士研究生, 主要研究方向为自动测试技术和智能故障诊断。