基于蚂蚁算法和遗传算法的时序电路测试生成

许川佩[***] 李 智[*] 莫 玮[*]
[*](桂林电子工业学院电子工程系 桂林 541004)
[**](西安电子科技大学机电工程学院 西安 710071)

摘 要： 为提高时序电路的测试生成效率，该文提出一种新的基于蚂蚁算法和遗传算法的时序电路测试矢量生成算法。针对国际标准时序电路的实验结果表明，该交叉算法既充分发挥了两种算法的优点，又克服了各自的缺点，与其它同类测试生成算法相比，获得了较好的故障覆盖率和测试集。说明采用蚂蚁算法和遗传算法的交叉算法是成功的。

关键词： 时序电路， 测试生成， 蚂蚁算法， 遗传算法
中图分类号： TN407 **文献标识码：** A **文章编号：** 1009-5896(2005)07-1157-05

Test Generation of Sequential Circuits Based on Ant Algorithm and Genetic Algorithm

Xu Chuan-pei[***] Li Zhi[*] Mo Wei[*]
[*] (*Dept. of Electronic Engineering, Guilin University of Electronic Technology, Guilin* 541004, *China*)
[**](*School of Mechano-Electronic Engineering, Xidian University, Xi'an* 710071, *China*)

Abstract This paper presents a new test pattern generation technique based on ant algorithm and genetic algorithm for improving the test generation efficiency for sequential circuits. Experimental results for the sequential benchmark circuits show that the hybrid approach not only takes full advantage of utilizing both algorithms, but also overcomes their disadvantages. It can achieve higher fault coverages and more compact test sets when compared to other similar test generation algorithms, demonstrating the combined algorithm is a successful algorithm.

Key words Sequential circuit, Test generation, Ant algorithm, Genetic algorithm

1 引言

数字集成电路的发展对测试提出了日益紧迫的要求，测试技术的发展已跟不上集成电路的发展，测试已成为妨碍 LSI/VLSI 付诸应用的瓶颈问题。几十年来，各国学者在数字电路的测试生成上已经做了大量的工作，但电路的测试仍然是个公认的难题，尤其是时序电路的测试生成，由于状态预置的困难和竞争冒险的存在，理论上是个没有完全解决的问题。测试问题的时空复杂性一方面使某些算法失去实用价值，另一方面又促进测试技术的发展，那就意味着除要对原有算法作改进外，必须研究和探索新的测试生成方法。

集成电路的测试生成方法主要有 3 类，第 1 类是确定性生成算法，第 2 类是基于符号和状态表的算法，第 3 类则是基于模拟的算法。从已有的测试生成算法看，确定性算法往往需要进行极大量的回溯判决，而且不能处理局部优化问题，因此仅适于处理小规模电路。例如确定性算法测试生成器 HITEC[1]。

基于符号和状态表的算法仅仅针对无故障电路生成测试序列，这些序列可能在故障电路中就变得无效，其故障覆盖率相对较差，但能得到很简洁的测试集。例如文献[2]。

另一方面，基于模拟的算法，则避免了回溯的复杂性，只须作前向处理，而且也比较容易处理复杂的元件类型，针对各种规模的时序电路作测试生成都能得到较好的故障覆盖率。所面临的问题主要是缺乏激活难测故障和传播相应故障影响所必需的信息，因而往往造成测试生成的时间较长。例如 CRIS[3]，GATEST[4]。因此如何选择合适的算法克服已有算法的缺点，进行更有效的时序电路测试生成具有很大的现实意义和应用前景。

2004-02-09 收到，2004-08-10 改回
国家自然科学基金(60266001)资助课题

综合比较 3 类算法，基于模拟的测试生成方法占有优势。本文研究的基于蚂蚁算法和遗传算法的时序电路测试生成也属于此类方法。

蚂蚁算法是模拟自然界蚂蚁寻找食物时在所经过的路径上留下一种挥发性的物质(称为信息素)，从而引导后续蚂蚁走最短路径而提出的一种算法，并已应用于时序电路的初始化处理[5, 6]。但蚂蚁算法的缺点是容易陷入局部优化。遗传算法(Genetic Algorithm, GA)的主要特点是群体搜索策略和群体中个体之间的信息交换，搜索不依赖于梯度信息，并已应用在测试生成方面，例如 Illinois 大学 Patel 教授等在 VLSI 的自动测试生成方面已取得了较好成果[4,7]。但遗传算法也存在缺点，搜索具有一定的盲目性。本文将蚂蚁算法和遗传算法结合起来，应用于时序电路的测试生成。

利用遗传算法的群体寻优特征及全局收敛特性，将蚂蚁算法生成的待测测试矢量作为启发性信息注入遗传算法中，作为 GA 的个体，引导 GA 的搜索，使 GA 达到快速收敛的目的，经过遗传算子操作的结果才加入测试集中，从而获得更优的测试生成序列。

2 基于蚂蚁算法和遗传算法的自动测试生成

2.1 算法描述

在测试生成中，一个测试矢量是由电路的主输入(Primary Input，PI)“0”，“1”组成的二进制字符串，我们所要寻找的是能检测到一个或一组故障(本文仅研究固定型故障)的测试矢量，将其作为激励加入到电路中，通过故障模拟得到测试的结果。

根据时序电路的特点，其前时帧的输入会对后时帧产生影响，而蚂蚁算法的特点是后续蚂蚁可依据前面经过蚂蚁留下的信息素选择路径，并最终可以发现最佳路径，因此本文作如下描述。

相关定义：将测试矢量定义为蚂蚁要经过的路径，PI 值“0”，“1”作为路径节点。测试序列作为 GA 的个体。每一个路径节点或个体中的字符对应时帧中的一个 PI 值。将电路的故障作为蚂蚁所要获得的食物。

用并行设计的思想，将电路按主输入扇出锥的划分法进行划分：从一个主输入 PI_i 出发，沿电路向主输出(Primary Output，PO)正向搜索，凡与 PI_i 逻辑相关的电路，都属于以 PI_i 为锥顶的输入扇出锥。这样就把逻辑相关性强的电路划分成 N 个子电路，各锥内的故障受所在锥 PI 值影响的可能性最大，因此把 PI_i 锥内的故障作为施加 PI_i 值后可能检测到的故障，在蚂蚁算法中作为蚂蚁经过 PI_i 节点后可能取得的食物，而将确定 PI_i 值后该锥内检测到的故障数作为信息素反映在此节点上，作为下一时帧 PI_i 取值的依据。但各个子电路之间并不是独立工作的，采用蚂蚁算法实现时，在同一时帧中，路径上各 PI 节点的值是同时进行处理的，即各扇出锥的测试生成以并行方式进行，使得各个子电路间的通信实现无缝连接，因此可大大提高加速比。

采用蚂蚁算法实现时，各扇出锥间的逻辑相关性不易分析处理，使得 PI 节点的取值可能只是局部最优值，因此引入另一种并行启发式搜索算法—GA 加以弥补，用于将多个待测序列进行组合以获取更有效的值。此外，经验指出，占电路故障绝大多数的易测故障对启发式的选择并不是那么严格，但难测故障则不同，它们往往只对某种特定的启发式才敏感，此时采用不同的启发式共同处理可提高加速比。

算法描述：先进行基于蚂蚁算法的待测矢量生成，一次发出 M 只蚂蚁(M 等于 GA 群体尺寸)，根据状态转移规则分别选择 M 条路径，之后将蚂蚁算法的生成结果作为 GA 种群的个体。经过 GA 操作得到的最优结果才作为测试矢量加入到测试集中，并用信息素更新规则更新各节点信息素。整个生成过程是通过蚂蚁算法和 GA 进行交叉反复循环实现的。本交叉算法所使用的有关规则定义如下。

2.2 基于蚂蚁算法的规则定义

在基于蚂蚁算法的测试生成中，遵循如下两种规则：

(1)信息素更新规则 假设共有 N 个 PI 节点，在 j+1 时帧第 i 个 PI 节点的信息素的更新规则为

$$P_i(j+1)=d_i/ud_i+(1-\alpha)\cdot P_i(j-1) \tag{1}$$

其中，$P_i(j+1)$ 是第 j+1 时帧 PI_i 节点的信息素，d_i 是 PI_i 输入扇出锥内在第 j 时帧检测到的故障数，ud_i 是 PI_i 输入扇出锥内所有未检测到的故障数，α=0.8，$i\in[1, N]$，j：表示当前时帧序号。

上式中，α 的取值决定前 $j-1$ 时帧检测到的故障数对 j+1 时帧 PI_i 取值的影响，若 $P_i(j-1)$ 较大，考虑到锥内的故障数有限，将 α 取 0.8，使得 $P_i(j+1)$ 不会太大，不过于引导 j+1 时帧的 PI_i 值保持。

(2)状态转移规则 整条路径由各个 PI 节点的值“0”或“1”构成，蚂蚁的选择就必定是这 N 个节点的“0”或“1”，即要么选“0”，要么选“1”，直到选完 N 个节点，即完成一条路径的选择。蚂蚁在第 i 个 PI 节点，选择 0 或 1 的规则如下：

$$\rho_i(j)=\begin{cases}1, & P_i(j)>\beta\\ 0, & \text{其它}\end{cases} \tag{2}$$

其中 β=γ 本电路中所有的故障数目，根据经验，取 γ=5；$i\in[1,N]$；j 表示当前时帧序号；1：表示当 $P_i(j)>\beta$ 时第 i 个 PI 节点的值保持不变；0：表示当 $P_i(j)\le\beta$ 时第 i 个 PI 节点的值取反，即由“0”变为“1”，或由“1”变为“0”。

实验表明，针对很多电路的测试生成，前后时帧使用相

同的测试矢量取得了不错的效果。实验中，也考虑加入随机信息，在连续 10 个时帧检测不到故障的情况下，对该矢量的各个节点值作随机保持或取反后，再作为新的待测矢量。

2.3 遗传算法的组成因子

遗传算法的组成因子，包括目标函数的构造，选择算子、交叉算子、变异算子的确定。

(1)目标函数的构造　有一个精确的目标函数才能获得高效的测试集。本文作如下定义：

fitness=本次检测到的故障数+已传播到 flip-flop 的故障数/((总故障数)(flip-flop 数))

目标函数的目的是寻找能检测最多故障的测试矢量，为了区分检测故障数相同的测试矢量，在函数中加进了传播到触发器的故障影响，因为这些故障影响很可能在下一时帧会传播到主输出。

(2)GA 的组成因子　首先是群体大小的设定。针对电路的易测故障对很多测试矢量敏感，本文将 GA 进化分 3 个阶段，在不同阶段使用不同的群体尺寸，第 1 阶段的群体尺寸见表 1，第 2 阶段的群体尺寸为第 1 阶段的两倍，第 3 阶段又为第 2 阶段的两倍，这样既能保证故障覆盖率，又减少了运行时间。第 2 个关键的参数是进化的代数，足够的进化代数才能保证将几个染色体中的有用信息组合到一个染色体中，本文将每 1 阶段进化的代数定为 8 代以减少运行时间。

选择因子、交叉因子和变异因子的选取同文献[6]，这种保留历代最优个体的遗传算法其收敛性已经为众多学者所证明[8]，在此不再赘述。本文中使用的群体尺寸和变异因子见表 1，其中矢量长度 L=PI 个数×电路结构时序深度，结构时序深度指的是 PI 和最远器件门之间的最少触发器数目，这样设置便于寻找到能将故障激活并传播到主输出的测试序列。

表 1　群体尺寸和变异因子

矢量长度(L)	群体尺寸	变异因子
< 4	8	1/8
4~16	16	1/16
> 16	16	1/L

3　测试生成实现

3.1 测试生成步骤

将基于蚂蚁算法和遗传算法的测试生成主要分成两个阶段，第 1 阶段是基于蚂蚁算法的测试生成，第 2 阶段是基于遗传算法的测试生成。

第 1 阶段　基于蚂蚁算法的测试生成。发出 M 只蚂蚁根据状态转移规则选择路径，一个蚂蚁选出的路径与 GA 一个个体对应。时帧 0，随机选取待测矢量，由于初始时信息素均为 0，根据状态转移规则将各 PI 节点值取反后，加入电路作故障模拟；考虑到电路的初态始于未知态，所以初始测试序列一方面用于检测故障，另一方面用于电路初始化，侧重于检测故障。其它时帧，使用状态转移规则时 PI 取值参考的是上一时帧 GA 个体序列的最后一个测试矢量。

检测到故障的待测矢量作为 GA 的个体，转入 GA 的调用。当一个蚂蚁根据规则选定待测矢量后，也有可能检测不到故障，但只要使得电路的状态发生变化，也会将此待测矢量加入 GA 种群中，否则就认为此蚂蚁已经死亡(即放弃此待测矢量)，需要放出另一只蚂蚁重新进行选择。

第 2 阶段　基于遗传算法的测试生成。由于 GA 的每个个体包含了多个测试矢量，但由蚂蚁算法仅得到一个，所以需要进行扩充。时帧 0，通过随机生成补足；其它时帧，将蚂蚁算法生成的待测矢量列在 GA 个体序列的最后，将序列中最前的一个待测矢量删去，如图 1 所示。然后进入选择、交叉、变异的进化操作过程，利用故障模拟器对每个个体实施模拟(注意对待测序列作模拟时要保存和恢复电路的正常和故障状态)。群体的进化要经过几代，直到群体中个体的适值不再提高，或者进化的代数已到了允许的最大值为止，才选出所有各代中最好的一个个体作为测试矢量，加入到测试集中。并将此测试序列检测到的故障返回，根据信息素更新规则标注在各 PI 节点上，重新转入第 1 阶段，进入测试生成的下一时帧处理。

如此循环反复，直到检测完所有故障或不能再检测到故障为止。

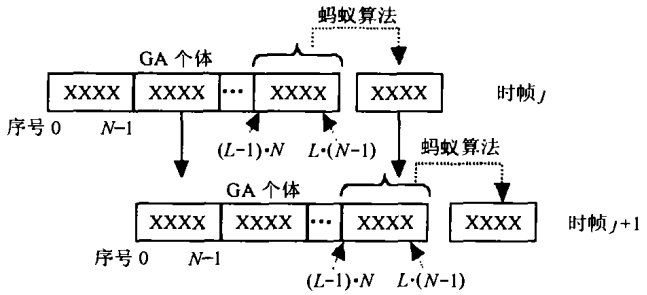


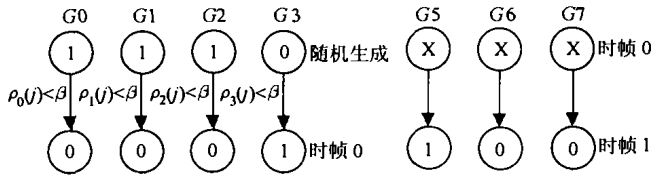
图 1　GA 个体与蚂蚁算法的路径之间转换关系

3.2 基于蚂蚁算法的测试生成示例

假设某电路包含 4 个主输入门 G0，G1，G2 和 G3；3 个 D 触发器 G5，G6 和 G7；以及其它器件门等。仅以时帧 0 为例说明矢量生成和状态转换过程，其它时帧的类似。

时帧 0，随机选取一个待测矢量“0111”，初始时 $P_i(j)=0(j=0$：表示时帧 0，$i=0\sim3$：分别表示主输入门 G0，G1，G2 和 G3)，$\beta>0$(具体计算省略)，调用蚂蚁算法后得到

“1110”，故障模拟的结果：成功实现了对 3 个触发器的初始化，并检测到 2 个故障，可作为待测矢量。矢量生成和状态转换如图 2 所示。同理，可获得其它 M-1 个待测矢量。



(a) 测试矢量生成图 (b) 状态转换图

图 2

4 实验结果

本文仿真对象选用国际标准时序电路：iscas’89 和几个同步时序电路。测试生成结果见表 2，其中包括各标准电路的总故障数，各生成器检测到的故障数。测试结果都是假设电路的初态为未知态下得到的。表中选取了两种取得较好结果的测试生成器作比较。其中 GATEST[4]是纯粹基于 GA 的经典测试生成器；STRATEGATE[7]是将 GA 和确定性算法进行了交叉的测试生成器。ANT-GA 即为本文提出的基于蚂蚁算法和遗传算法的测试生成方法。

表 2 测试生成结果

电路	总故障数	GATEST		STRATEGATE		ANT-GA	
		Det	Vec	Det	Vec	Det	Vec
s298	308	264	161	**265**	306	**265**	120
s344	342	**329**	95	**329**	86	**329**	72
s382	399	347	281	**364**	1486	**364**	1492
s444	474	405	275	**424**	1945	**424**	1631
s526	555	417	281	**454**	2642	**454**	4712
s641	467	**404**	139	**404**	166	**404**	114
s713	581	**476**	128	**476**	176	**476**	114
s820	850	517	146	**814**	590	771	684
s832	870	539	150	**818**	701	736	944
s1196	1242	1232	347	**1239**	574	**1239**	604
s1238	1355	1274	383	**1282**	624	1280	704
s1423	1515	1222	663	**1414**	3943	1381	1480
s1488	1486	1392	243	**1444**	593	**1444**	808
s1494	1506	1416	245	**1453**	540	**1453**	656
s5378	4603	3175	511	**3639**	11571	3541	9537
s35932	39094	35009	197	35100	257	**35101**	247
am2910	2391	2163	745	**2198**	2509	**2198**	312
mult16	1708	1653	204	1665	1530	**1666**	248
div16	2147	1739	634	**1814**	3476	1985	652
pcont2	11300	6826	272	**6837**	194	6835	108
piir8o	19920	15013	531	15071	354	**15081**	225
piir8	29689	-	-	**18206**	443	18141	282

Det: 已检测到的故障数 Vec: 测试矢量集长度

表 2 中标为黑体的数值是 3 种算法中故障覆盖率最高的数值。比较可发现，ANT-GA 算法在小规模电路中具有比较明显的优势，针对 s298，s344，s382，s444，s526，s641，s713 和 s1196 都获得了最好的故障覆盖率；而且相应的测试矢量长度也比较短，如对 s298，s344，s641，s713 都获得了最压缩的测试集。在某些大规模电路上也获得了最好的故障覆盖率，如对 s35932，mult16 以及 piir8o 得到了超过其它所有生成器的结果，而且相应的测试矢量集是最短的。

就所耗费的 CPU 时间而言，由于硬件设备等不同，各种算法的生成时间不能直接作比较，本文只略作分析说明。调用蚂蚁算法所耗费的时间仅是作故障模拟时间的线性阶，可用算式表示，设作一次故障模拟的时间为 t，则一代 GA 耗费的时间 T_{GA}＝群体尺寸×L×8(进化代数)×3(阶段数)×t，蚂蚁算法耗费的时间 T_{ANT}＝群体尺寸×t，因此一次 ANT-GA 交叉操作所耗费的时间=T_{GA}＋T_{ANT}。已单独进行过基于蚂蚁算法的测试生成实验，CPU 时间短的不足 1s，最长的不超过 10min，但不足之处是测试矢量集过大；基于 ANT-GA 的 CPU 时间最短的仅数秒，但最长的达数十小时，在此不再一一列举。

5　结束语

本文主要是进行时序电路的测试生成新算法的研究，重点是研究基于蚂蚁算法和遗传算法的测试矢量生成。从实验结果可看出，这种交叉既充分发挥了两种算法的优点，又克服了二者的缺点，获得了较好的故障覆盖率，尤其与 GATEST 相比更明显，说明 ANT-GA 具有很强的快速搜索能力，这跟蚂蚁算法提供的启发性信息是分不开的；也获得了较短的测试矢量集，这是充分利用了 GA 的群体寻优特征及全局收敛特性的结果。结果证明采用蚂蚁算法和 GA 的交叉算法是成功的。

参 考 文 献

[1] Niermann T M, Patel J H. HITEC: A test generation package for sequential circuits. Proc. European Conf. Design Automation, Amsterdam, the Netherlands, 1992: 214 – 218.

[2] Cabodi G, Camurati P, Quer S. Symbolic exploration of large circuits with enhanced forward/backward traversals. Proc. EURODAC, Grenoble, Fr., 1994: 22 – 27.

[3] Saab D G, Saab Y G, Abraham J A. CRIS: A test cultivation program for sequential VLSI circuits. Proc. Int. Conf. Computer -Aided Design, Santa Clara, USA, 1992: 216 – 219.

[4] Rudnick E M, Patel J H, Greenstein G S, Niermann T M. Sequential circuit test generation in a genetic algorithm framework. Proc. Design Automation Conf., San Diego, USA, 1994: 698 – 704.

[5] 李智，许川佩，陈光禹. 基于蚂蚁算法的同步时序电路初始化研究. 电子测量与仪器学报, 2002, (4): 33 – 38.

[6] 李智，许川佩，莫玮，陈光禹. 基于蚂蚁算法和遗传算法的同步时序电路初始化. 电子学报, 2003, (8): 1276 – 1280.

[7] Hsiao M S, Rudnick E M, Patel J H. Dynamic state traversal for sequential circuit test generation. *ACM Trans. on Design Automation of Electronic Systems*, 2000, 5(2): 548 – 565.

[8] 陈国良，等. 遗传算法及其应用. 北京：电子工业出版社, 1996, 6: 88 – 97.

许川佩：女，1968 年生，副教授，博士生，研究方向为集成电路测试理论与技术.

李　智：男，1965 年生，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.

莫　玮：男，1956 年生，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.