文章编号:1006-2467(2019)10-1218-07 DOI: 10.16183/j.cnki.jsjtu.2019.99.001

基于人工蜂群的硬件木马测试向量生成方法

王晓晗, 王 韬, 李雄伟, 张 阳, 黄长阳
(陆军工程大学石家庄校区 装备模拟训练中心, 石家庄 050003)

摘 要: 针对已有测试向量生成方法对以电路惰性节点作为输入的硬件木马触发覆盖率低的问题,提出了一种基于人工蜂群的测试向量生成方法. 首先分析了用于触发惰性节点组合的测试向量的分布规律,并构建数学模型对其进行描述;然后利用人工蜂群算法生成测试向量,结合其分布规律对局部区域进行高效搜索以发现能触发更多惰性节点组合的测试向量,同时对全局进行快速搜索,有效避免了"早期收敛"问题. 实验结果表明:使用本文方法生成的测试向量测试电路,对电路中惰性节点组合的平均触发覆盖率达到 95.86%,与已有方法相比提高了 22.43%,具有更好的硬件木马激活效果.

关键词: 硬件木马; 集成电路; 惰性节点; 人工蜂群算法; 激活策略
中图分类号: TN 918 **文献标志码**: A

Test Pattern Generation Method for Hardware Trojan Detection Based on Artificial Bee Colony

WANG Xiaohan, *WANG Tao*, *LI Xiongwei*, *ZHANG Yang*, *HUANG Changyang*
(Equipment Simulation Training Center, Army Engineering University Shijiazhuang Campus, Shijiazhuang 050003, China)

Abstract: The exiting test pattern generation method have the problem of low trigger coverage for hardware Trojan detection. In order to solve this problem, a test pattern generation method based on artificial bee colony algorithm is proposed. Firstly, the distribution regularity of test patterns which can trigger the combination of inactive nets is analyzed. And the mathematical model is constructed to describe the test pattern. Then, the test pattern is generated by artificial bee colony algorithm. Combining with its distribution regularity, this method can search local regions efficiently to find test patterns that can trigger more combinations of inactive nets. At the same time, it can search global world quickly and effectively avoid the problem of "premature convergence". The experimental results show that using the test vectors generated by this method to test circuit, the average trigger coverage rate of hardware Trojan can reach 95.86%. Compared with the existing method, this method improves 22.43%, and has better hardware Trojan activation effect.

Key words: hardware Trojan; integrated circuit (IC); inactive net; artificial bee colony algorithm; activation strategy

收稿日期:2018-11-07
基金项目:国家自然科学基金资助项目(61602505)
作者简介:王晓晗(1992-),男,河北省衡水市人,博士生,主要研究方向为芯片安全.
通信作者:李雄伟,男,副教授,博士生导师,电话(Tel.):0311-87994929;E-mail:lxw-wys@163.com.

硬件木马作为一种新型硬件攻击形式[1]，能够在集成电路(Integrated Circuit，IC)设计与制造的任一环节被植入到 IC 中，一旦被成功植入，便可在特定条件下激活并泄露 IC 中的秘密信息，甚至破坏 IC 功能，严重威胁 IC 安全. 目前，具有成本低、灵敏度高等特点的旁路分析技术是对芯片内硬件木马进行检测的主流方法[2]. 其主要做法是利用芯片的旁路特征参数(功耗[3]、电磁[4]、延迟[5]等)来检测芯片中的硬件木马. 无论检测时选用的参考对象是“金片”(确定无木马的芯片)的旁路信号还是芯片制造/工作过程中的其他信息[6]，都需要输入恰当的测试向量完全激活(使木马表现出恶意特征)或者部分激活(放大硬件木马对整体电路旁路信号的影响)硬件木马，从而通过旁路特征参数差异检测硬件木马的存在. 然而集成电路的输入较多，无法通过全测试向量空间对集成电路进行测试(例如，40 个输入的集成电路的全测试空间为 2^{40})，因此，如何针对电路中硬件木马可能的植入位置生成行之有效的测试向量集是当前的一个研究热点. 例如针对以电路中惰性节点(电路中 0-1 翻转概率低的节点)[7]作为输入的硬件木马，文献[8]以多次触发惰性节点的方式生成测试向量集，在同等硬件木马检测覆盖下所需测试向量数仅为随机测试向量的 15%，但是此方法难以激活翻转概率低的惰性节点上的硬件木马. 针对此问题，文献[9]利用遗传算法(Genetic Algorithm，GA)生成可触发惰性节点组合的测试向量，进而激活硬件木马. 文献[10]针对多输入电路中利用电路输入潜藏的硬件木马生成最小完备的测试向量集对其进行激活. 文献[11]则针对电路的局部结构生成最优的测试向量以提高硬件木马相对原始电路的活跃度，与随机测试向量相比硬件木马的检测灵敏度提高了 41.05%. 文献[12]利用故障测试的思想生成测试向量以激活电路中任意位置植入的硬件木马，但是该方法只能检测故障型硬件木马且生成的测试向量集的规模较大.

对于以惰性节点作为输入的组合型硬件木马而言，为了使其完全/部分激活，需猜测其可能的植入位置(惰性节点组合)，并生成测试向量触发该惰性节点组合. 因此理想的测试向量集需触发尽可能多的惰性节点组合，即具有较高的触发覆盖率，从而完全/部分激活可能植入的硬件木马. 对于该问题，文献[9]方法生成的测试向量集虽然能够覆盖大多数硬件木马可能的植入位置，但触发覆盖率相对较低，因此，本文在分析能够触发惰性节点组合的测试向量分布规律的基础上，引入人工蜂群(Artificial Bee Colony，ABC)算法，并提出一种基于人工蜂群算法的测试向量生成方案，以提高测试向量集的触发覆盖率.

1　问题分析

通过计算集成电路中节点的翻转概率和设定合适的阈值可筛选出电路中翻转概率低的惰性节点[7]，这些惰性节点在大多数测试向量下为某一定值(0 或 1)，即 P_0^i 远小于 P_1^i 或 P_0^i 远大于 P_1^i，其中 P_0^i 和 P_1^i 分别表示节点 i 取值为 0 和 1 的概率，当惰性节点取概率低的值时表示该惰性节点被触发. 相较于其他植入位置的硬件木马而言，输入均为惰性节点的硬件木马更难被激活，换言之，能够同时触发多个惰性节点的测试向量更难被发现. 这类硬件木马可植入的位置数随电路中惰性节点数和硬件木马输入数的变化而变化，若电路中惰性节点数为 n，硬件木马输入个数为 r，则惰性节点组合(惰性节点的所有组合)数为 2^n，其中 C_n^r 个惰性节点组合可能被植入硬件木马.

通过对这些惰性节点组合对应的测试向量进行分析可知其分布规律：一方面，仅有部分/全部输入对触发惰性节点组合是有用的. 根据电路的逻辑连接关系，电路节点的取值仅与与其直接相连的部分/全部输入相关. 如图 1(虚线部分)所示，以 c17 电路为例，节点 n_{22} 的取值仅取决于输入 n_1，n_2，n_3 和 n_6 的值. 同理，对于惰性节点组合而言，亦可根据此特点圈定一组可触发该组合的测试向量，假设电路总输入个数为 m，与惰性节点组合对应的输入个数为 s，当 s 个输入的值能触发某一惰性节点组合时，无论其他 $m-s$ 个输入为何值均能触发该惰性节点组合. 令 s 个输入的值不变，其余 $m-s$ 个输入的值均为 X，可得到触发该惰性节点组合的一个触发模式. 在该触发模式下共有 2^{m-s} 个测试向量能触发该惰性节点组合，这些测试向量在测试向量空间中紧密分布构成一个超立方体，显然该立方体的大小可变. 如图 2 所示，将测试向量的每一位看作一个维度，则图 2 为 3 输入电路的全测试向量分布，两个虚线区域分别为能够触发 2 个惰性节点组合的测试向量集，其中红色虚线部分为 $s=2$ 时的一组测试向量(触发模式为 $X00$)，绿色虚线部分为 $s=1$ 时的一组测试向量(触发模式为 $1XX$). 需要注意的是，对于某个惰性节点组合而言，触发该组合的测试向量可能不存在，也可能有一组或多组(即 s 个输入有多组值可触发该惰性节点组合)测试向量. 另一方面，在能够触发某个惰性节点组合的一组测试向量中，若存在某个

测试向量能够触发更多的惰性节点组合，则该测试向量也存在于可触发其他惰性节点组合的一组测试向量中. 假设有 5 个惰性节点，分别为 n_1，n_2，n_3，n_4 和 n_5，开始根据前 4 个惰性节点找到一组可触发该组合的测试向量，如果在该组测试向量中有一个测试向量能够同时触发 5 个惰性节点，那么该测试向量也存在于可触发其他惰性节点组合（例如惰性节点组合 n_2，n_3，n_4 和 n_5，或者惰性节点组合 n_1，n_2，n_4 和 n_5）的一组测试向量中. 根据这一特点，可以通过多组测试向量取交集的方式发现可触发更多惰性节点组合的测试向量. 如图 2 所示，2 个虚线区域交集中的测试向量为“100”，能同时触发 2 个惰性节点组合.

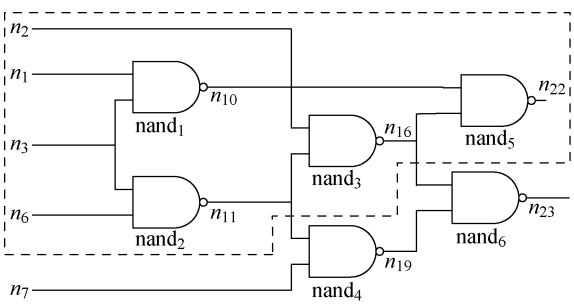

图 1 c17 电路

Fig. 1 c17 circuit

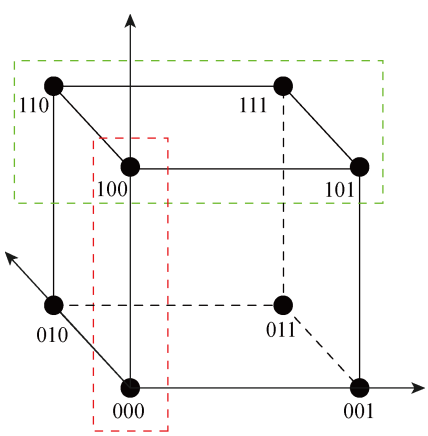

图 2 测试向量分布示意图

Fig. 2 Schematic diagram about distribution of test vector

对于能够触发惰性节点组合的测试向量集而言，其测试向量在测试向量空间中以一个个独立子集合的形式存在，每个子集合对应一种触发模式. 此时仅采用交叉和变异的方式生成新测试向量的文献[9]方法很容易产生“早期收敛”问题，丧失进化能力，难以发现未知触发模式对应的测试向量，因此本文引入人工蜂群算法生成测试向量.

2 人工蜂群算法简介

人工蜂群算法是由 Karaboga 在 2005 年提出的一种模拟蜂群采蜜行为的智能优化算法[13]，该算法通过模拟蜜蜂搜索蜜源、开采蜜源和放弃蜜源的行为搜索最优解. 在搜索过程中该算法可同时兼顾局部搜索和全局搜索（符合可触发惰性节点组合的测试向量集的分布特性），较大概率的搜索最优解. 且该算法易于实现，执行效率高，执行过程中使用的参数较少，具有良好的鲁棒性.

在人工蜂群算法中蜜源是指待搜索的解，适应度函数则是用来评估蜜源的质量，若发现适应度高的新蜜源则替换掉原蜜源，从而发现最优解. 在搜索过程中根据蜜蜂的职能不同可将蜜蜂分为 3 种类型：采蜜蜂，观察蜂和侦察蜂. 其中采蜜蜂和观察蜂均是对蜜源进行开采（在蜜源附近进行搜索以发现局部最优解），但是采蜜蜂需与蜜源一一对应，而观察蜂则可以根据蜜源的价值（通过收益率函数计算可得）选择蜜源进行开采. 侦察峰则是在蜜源开采完成后用以发现新蜜源（寻找全局最优解）. 人工蜂群算法寻优的过程就是先初始化蜜源，然后不断重复 3 种蜜蜂的工作直至达到最大迭代次数.

3 基于人工蜂群算法的测试向量生成

原始人工蜂群算法在搜索局部最优解和全局最优解时，均采用随机的方式生成新蜜源，对于生成可触发惰性节点组合的测试向量而言，采用这种方式搜寻新测试向量的效率相对较低. 因此，为了利用所寻测试向量的分布规律，提高算法的搜索效率，本文对人工蜂群算法进行以下几种改进：

（1）定义数学模型对能触发惰性节点组合的测试向量进行描述.

（2）定义算法的适应度函数，使其适用于发现符合条件的新测试向量.

（3）进行全局搜索时，按照已发现触发模式的形式生成新的触发模式，若新触发模式下的任一蜜源能触发某一惰性节点组合，则该触发模式能触发该惰性节点组合. 这种方式一方面可以减少搜索空间，另一方面也便于发现可触发惰性节点组合的其他触发模式.

（4）进行局部搜索时，通过子集合取交集的方式发现可触发更多惰性节点组合的测试向量. 因此需设定人工蜂群算法中的蜜蜂具有记忆功能，能够记忆和共享已经发现的所有触发模式以及与之对应的子集合.

3.1 数学模型

根据待搜寻测试向量的分布特点需要定义测试向量间距离、蜜源集、蜜源集中心等基本概念.

定义 1 设 $\boldsymbol{v}_k$ 和 $\boldsymbol{v}_j$ 为任意 2 个测试向量,i 为测试向量的第 i 位,则测试向量之间的距离为

$$d(\boldsymbol{v}_j,\boldsymbol{v}_k)=\max_i|(\boldsymbol{v}_j)_i-(\boldsymbol{v}_k)_i| \quad (1)$$
$$i=1,2,\cdots,m$$

定义 2 蜜源集(NSP).将每个测试向量称为蜜源(NS),则在所有的蜜源中,可由同一触发模式表示的一组测试向量称为蜜源集(NSP).

定义 3 设 $\boldsymbol{v}_{\mathrm{m}}$ 为蜜源集 NSP 的中心,与惰性节点组合对应的输入个数为 s,则蜜源集的中心可表示为

$$\boldsymbol{v}_{\mathrm{m}}=\frac{1}{2^{m-s}}\sum_{j=1}^{2^{m-s}}\boldsymbol{v}_j,\quad \forall\,\boldsymbol{v}_j\in\mathrm{NSP} \quad (2)$$

定义 4 在一个蜜源集 NSP 中,蜜源集中的测试向量 $\boldsymbol{v}_j$ 与蜜源集中心 $\boldsymbol{v}_{\mathrm{m}}$ 的距离满足以下关系:

$$\{\forall\,\boldsymbol{v}_j\in\mathrm{NSP}:\ d(\boldsymbol{v}_j,\boldsymbol{v}_{\mathrm{m}})\leqslant 0.5\} \quad (3)$$

定义 5 对于 2 个蜜源集 NSP_1 和 NSP_2,若各自蜜源集中心 $\boldsymbol{v}_{\mathrm{m1}}$ 和 $\boldsymbol{v}_{\mathrm{m2}}$ 之间的距离满足:

$$d(\boldsymbol{v}_{\mathrm{m}},\boldsymbol{v}_{\mathrm{m}})\leqslant 0.5 \quad (4)$$

则 2 个蜜源集相交,即 2 个蜜源集具有共同的测试向量.

3.2 适应度函数设计

适应度函数是人工蜂群算法成功与否的关键因素.根据前文中的分析,能够触发惰性节点组合的每个蜜源集中都可能包含触发其他惰性节点组合的测试向量,即每发现一个能触发某个惰性节点组合的蜜源集均有助于发现能触发更多惰性节点组合的测试向量,因此适应度函数设计为

$$f(\boldsymbol{v})=\mathrm{sign}(n_{\mathrm{a}}-r+1)\sum_{s_j\in S}n_{\mathrm{u}}(s_j) \quad (5)$$

式中:S 为惰性节点组合集;s_j 为集合 S 中的一个惰性节点组合;$n_{\mathrm{u}}(s_j)$ 为从算法开始到发现测试向量 $\boldsymbol{v}$ 为止的所有测试向量中,能够触发惰性节点组合 s_j 的测试向量总数;n_{a} 为测试向量 $\boldsymbol{v}$ 能够触发的惰性节点数;r 为硬件木马的输入数;

$$\mathrm{sign}\ x=\begin{cases}1, & x>0\\ 0, & x=0\\ -1, & x<0\end{cases} \quad (6)$$

3.3 测试向量生成算法

结合待搜寻测试向量的分布规律,下面给出采用人工蜂群算法生成测试向量的详细方案.

步骤 1 初始化.随机选择 q 个惰性节点组合,根据惰性节点组合与电路输入的对应关系可转化为布尔可满足性问题(SAT)[14],然后利用 minisat 求解器对其进行求解,得到 q 个能够触发惰性节点组合的蜜源集,从每个蜜源集中随机选择一个测试向量作为蜜源,记为矩阵 $\boldsymbol{B}=\{b(j,i)\mid j=1,2,\cdots,q;\ i=1,2,\cdots,m\}$,并根据式(2)计算出 q 个蜜源集中心(代表蜜源集,可根据蜜源集中心和定义 4 写出蜜源集中的所有测试向量,需要注意的是蜜源集中心与触发模式的格式相同,只需将值为 X 的位换为 0.5 即可,值为整数的位即为与惰性节点组合对应的输入),记为矩阵 $\boldsymbol{C}=\{c(j,i)\mid j=1,2,\cdots,q;i=1,2,\cdots,m\}$,其中 m 为电路的输入个数.

步骤 2 利用式(5)计算 q 个蜜源的适应度,得到一维矩阵 $\boldsymbol{F}=[f(1)\ \ f(2)\ \ \cdots\ \ f(q)]$,并将 q 个蜜源和 q 个蜜源集中心分别存储到测试向量集 V 和蜜源集中心的集合 V_{m} 中.

步骤 3 为每个蜜源集(共 q 个)设定一个控制参数,用以控制对每个蜜源集进行探索的次数,得到一维矩阵 $\boldsymbol{L}_i=[l(1)\ \ l(2)\ \ \cdots\ \ l(q)]$.

步骤 4 统计 V_{m} 中蜜源集中心的数目 N_{vm},将其作为当前 q 个蜜源集的最大探索次数.

步骤 **5** 采蜜蜂阶段.生成 q 个采蜜蜂,每个采蜜蜂对应矩阵 $\boldsymbol{C}$ 中的 1 个蜜源集中心,若蜜源集 j 的控制参数 $l(j)$,小于阈值 N_{vm},则采蜜蜂对该蜜源集进行开采,否则不进行任何操作.

(1) 开采蜜源集时,根据定义 5 判断 V_{m} 中的第 $l(j)$ 个蜜源集与当前的蜜源集 j 是否相交,若不相交则重复迭代 $l(j)=l(j)+1$,直至找到与蜜源集 j 相交的蜜源集为止或者 $l(j)$ 的值超过阈值 N_{vm}.

(2) 若 2 个蜜源集相交则更新矩阵 $\boldsymbol{L}_i$ 中 $l(j)$ 的值,并根据 2 个蜜源集中心生成新的候选蜜源 $\boldsymbol{v}_{\mathrm{sol}}$,具体做法是:$c(j,i)$ 和 $V_{\mathrm{m}}(l(j),i)$ 中有一个值为整数(0 或 1),则 $\boldsymbol{v}_{\mathrm{sol}}$ 中第 i 位的值与其相同,否则随机选择 0 或 1 作为该位的值.

(3) 利用式(5)计算 $\boldsymbol{v}_{\mathrm{sol}}$ 的适应度并与 $f(j)$ 进行比较,$\boldsymbol{v}_{\mathrm{sol}}$ 的适应度高则替代原蜜源,同时更新矩阵 $\boldsymbol{B}$ 的第 j 行和 $f(j)$.

步骤 6 计算蜜源集的收益率.根据蜜源集对应蜜源的适应度计算蜜源集的收益率,令收益率为 $\boldsymbol{F}=[f_{\mathrm{p}}(1)\ \ f_{\mathrm{p}}(2)\ \ \cdots\ \ f_{\mathrm{p}}(q)]$,则每个蜜源集的收益率可表示为

$$f_{\mathrm{p}}(j)=\frac{0.9f(j)}{\max\limits_i f_{\mathrm{p}}(i)}+0.1 \quad (7)$$
$$i,j=1,2,\cdots,q$$

式中:$\max\limits_i f_{\mathrm{p}}(i)$ 为矩阵 $\boldsymbol{F}$ 中的最大值.

步骤 7 跟随蜂阶段.生成 q 个跟随蜂,每个跟

随蜂可根据蜜源集的收益率随机选择一个蜜源集进行开采，具体做法是按照顺序不断迭代选择蜜源集，并在 0-1 之间生成随机数，当随机数小于蜜源集的收益率时，则该跟随蜂对当前蜜源集进行开采，其开采过程与步骤 5.1～步骤 5.3 相同. 需要注意的是收益率越大的蜜源集被开采的概率越大，对其进行开采的跟随蜂的数量也越多.

步骤 8 侦察蜂阶段. 遍历矩阵 $\boldsymbol{C}$ 中的 q 个蜜源集，当蜜源集 j 的控制参数 $l(j)$ 超过最大探索次数 N_{vm} 时，侦察峰将探索新的蜜源集和蜜源，具体做法是在 V_{m} 中随机选择一个蜜源集中心，并按照其格式生成候选蜜源集中心(保证值为 0.5 的位不变，其他位随机生成 0 或 1)，在候选蜜源中心对应的蜜源集中随机选择候选蜜源 $\boldsymbol{v}_{\mathrm{sol}}$，并计算 $\boldsymbol{v}_{\mathrm{sol}}$ 的适用度，若高于蜜源集 j 的适用度，则替代当前蜜源集和蜜源，同时更新矩阵 $\boldsymbol{B}$ 和 $\boldsymbol{C}$ 的第 j 行以及 $f(j)$，并在 V_{m} 中记录该蜜源集中心，否则不进行任何操作.

步骤 9 重复迭代步骤 3～步骤 8 直到超过最大迭代次数.

4 实验验证

4.1 实验配置

为验证本文方法的有效性，采用触发覆盖率(η)作为评估指标，即

$$\eta = C_{\mathrm{ag}}/C_{\mathrm{total}} \tag{8}$$

式中：C_{ag} 为测试向量集可触发的惰性节点组合数，C_{total} 为所有能被触发的惰性节点组合数. 由于电路中惰性节点组合的数目很多(与电路结构，惰性节点筛选阈值，硬件木马输入个数有关)，很容易产生组合爆炸问题，例如电路惰性节点数为 100 个，硬件木马的输入为 4 时，惰性节点组合数为 3 921 225，难以遍历所有的惰性节点组合，因此实验随机抽选 100 000 条惰性节点组合构成 S，筛选惰性节点的阈值设为 0.1，硬件木马为 4 输入的组合型硬件木马(对于时序型硬件木马的测试向量集，只需根据相同输入数的组合型硬件木马的测试向量集进行生成即可，生成时保证原测试向量集中每个测试向量均在新测试向量集中出现 N 次，N 为时序型硬件木马的状态数[8]). 实验时，分别从 ISCAS85(组合电路)和 ISCAS89(时序电路)电路中各选取 4 个电路(c1355、c3540、c5315 和 c7552)和 2 个电路(s1238、s5378)对两种方法进行对比，时序电路 s1238 和 s5378 均处于全扫描测试模式(即寄存器相当于电路的输入). 实验中算法的参数设置均与文献[9]一致，组合电路的种群规模设置为 200，时序电路的种群规模设置为 500，最大迭代次数为 200，遗传算法的交叉率为 0.9，变异率为 0.05，利用 minisat 求解器计算出 S 中 2 500(组合电路)和 5 500 (时序电路)个惰性节点组合的测试向量初始化种群. 实验的相关算法均用 MATLAB 实现，实验用的 PC 配置为 3.20 GHz 处理器(i5-4750)和 4 GB 内存.

4.2 实验结果分析

图 3 和图 4 分别为本文方法和文献[9]方案(包括 3 部分，第 1 部分利用遗传算法生成测试向量集，第 2、3 部分则是对该测试向量集的补完以提高触发覆盖率和减少测试向量数)的遗传算法部分在 c1355 和 s5378 电路下生成测试向量的触发覆盖率随迭代次数(100 次迭代)变化的对比图，横坐标为算法的迭代次数，纵坐标为触发覆盖率，'o'代表基于遗传算法(GA)的变化趋势图，'＊'代表基于人工蜂群算法(ABC)的变化趋势图. 从图中可以看出，在算法初期本文方法所生成测试向量的触发覆盖率上升更快，当触发覆盖率达到一定程度时，基于遗传算法的测试向量生成陷入局部最优，很难发现能触发其他惰性节点组合的测试向量，而本文方法仅仅是搜索速率下降，仍具有良好的全局搜索能力，避免了局部最优问题，达到了更高的触发覆盖率. 其主要原因是：一方面本文方法按照蜜源集中心的格式发现新蜜源集中心，仅需猜测与惰性节点组合对应输入的值即可，这种方式减少了测试向量的搜索空间，发现符合条件的测试向量的概率更高；另一方面是利用蜜源集取交集的方式发现能触发更多惰性节点组合的测试向量，对局部搜索更精准、高效，因此本文算法发现可触发新惰性节点组合的测试向量的效

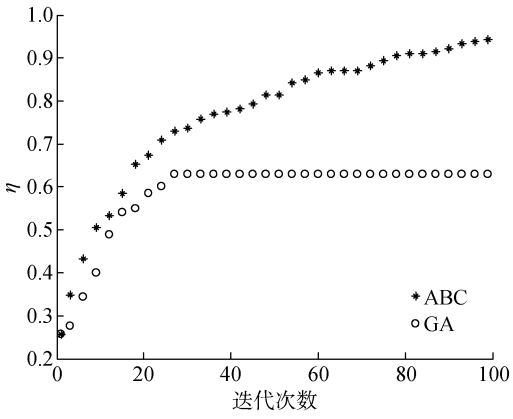

图 3 两种算法在模拟环境下针对 c1355 电路生成测试向量的对比

Fig. 3 Comparison of test vectors generated by two algorithms for c1355 circuit in simulation environment

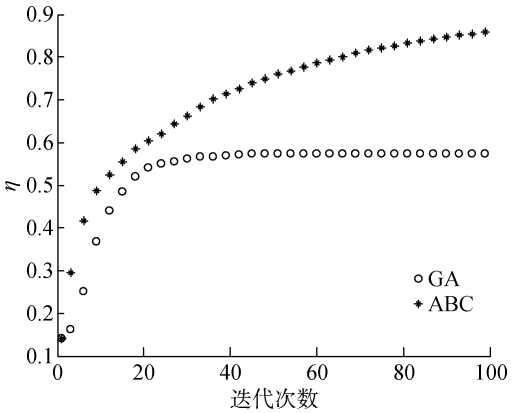

图 4　两种算法在模拟环境下针对 s5378 电路 生成测试向量的对比

Fig. 4　Comparison of test vectors generated by two algorithms for s5378 circuit in simulation environment

率更高，所生成测试向量的触发覆盖率上升速度更快.

表 1 为本文方法和文献[9]方案在六种电路下生成测试向量情况(300 次迭代)的对比，从数据中可知，在触发覆盖率方面，本文方法的平均触发覆盖率达到了 95.86%，而文献[9]的平均触发覆盖率为 73.43%，与文献[9]相比提高了 22.43%. 其主要原因是与惰性节点对应的电路输入数一般少于电路输入总数，每个独立测试向量子集中的向量数相对较多，能触发更多的惰性节点组合的测试向量数相对较少，导致文献[9]寻找到的测试向量大多功能重复(触发相同的惰性节点组合)而触发覆盖率并没有显著提升，而本文方法以取交集的方式发现新测试向量可有效避免功能重复的测试向量，精准定位能触发更多惰性节点组合的测试向量. 此外，本文采用的人工蜂群算法具有更好的全局搜索能力，更适合由电路结构因素导致的所搜索测试向量的分布特性，结合本文定义的适应度函数能尽可能的发现散落在空间中的测试向量子集，因此可以实现更高的触发覆盖率.

在测试向量数方面，本文算法生成的测试向量数一般少于文献[9]，但也存在测试向量数较多的情况，造成这种现象的原因主要与电路内部节点之间的逻辑关系有关，惰性节点之间具有一定的相关性，触发其中一个惰性节点的同时也会触发与之相关的惰性节点，具有相关性的惰性节点数越多，同一个测试向量可触发的惰性节点组合越多，触发这些惰性节点组合所需的测试向量数越少. 因此本文方法生成的测试向量数一般少于文献[9]，也不排除测试向量数更多的情况(相关的惰性节点数少). 综上所述，本文算法的性能优于文献[9]，适用于生成激活硬件木马的测试向量集.

在时间复杂度方面，考虑最坏的情况，本文算法在搜索局部最优解时需要遍历所有的蜜源集中心 N_{vm}，则单次迭代中本文算法的时间复杂度为 $O(qmN_{vm})$. 同理，在最坏情况下，遗传算法的时间复杂度为 $O(qm)$. 虽然本文算法的时间复杂度有所增加，但本文算法提高了发现符合条件的测试向量的能力以及触发覆盖率，因此增加的计算代价仍可接受.

表 1　模拟环境下两种算法生成的测试向量的对比

Tab. 1　Comparison of test vectors generated by two methods in simulation environment

电路	门数	输入数	搜索空间	迭代次数	测试向量数		η/%	
					文献[9]	ABC	文献[9]	ABC
c1355	545	41	2^{41}	300	1 740	1 692	67.92	99.44
c3540	1 668	50	2^{50}	300	15 403	9 747	77.46	97.64
c5315	2 307	178	2^{178}	300	16 497	36 345	75.83	93.96
c7552	3 513	207	2^{207}	300	14 363	13 589	74.37	90.37
s1238	508	32	2^{32}	300	16 909	15 042	81.55	99.68
s5378	2 779	214	2^{214}	300	12 198	9 649	63.43	94.09
平均	1 887	120	2^{120}	300	12 852	14 344	73.43	95.86

5　结论

基于人工蜂群算法的测试向量生成方案能够有效生成测试向量并激活以电路中惰性节点作为输入的硬件木马，通过实验进行验证，生成的测试向量能覆盖电路中大多数的惰性节点组合，证明该方法所生成的测试向量集优于文献[9]，具有一定的适用性. 此外，该方法仅需知道电路的逻辑结构即可对电

路中的硬件木马进行检测，而不受限于电路的具体形态，无论是封装芯片、硬核等多种电路形式均具有良好的应用效果. 然而本方法虽然提高了全局搜索能力，但是在算法后期搜索效率下降，为了加快算法的执行效率，需根据电路结构信息进一步分析能触发惰性节点组合的测试向量的分布规律，或者探寻其他有效的启发式探索策略，以期提高测试向量生成效率以及硬件木马激活效率.

参考文献：

[1] GOERTZEL K M. Integrated circuit security threats and hardware assurance countermeasures [J]. **Crosstalk Real-Time Information Assurance**, 2013, 26(6): 33-38.

[2] BHUNIA S, ABRAMOVICI M, AGRAWAL D, *et al*. Protection against hardware Trojan attacks: Towards a comprehensive solution[J]. **IEEE Design Test Computer**, 2013, 30(3): 6-17.

[3] AGRAWAL D, BAKTIR S, KARAKOYUNLU D, *et al*. Trojan detection using IC fingerprinting[C]// **Proceedings of the Symposium on Security and Privacy** (**SP**). Berkeley, USA: IEEE, 2007: 296-310.

[4] BALASCH J, GIERLICHS B, VERBAUWHEDE I. Electromagnetic circuit fingerprints for hardware Trojan detection[C]// **Proceedings of Electromagnetic Compatibility** (**EMC**). Dresden, Germany: IEEE, 2015: 246-251.

[5] XIAO K, ZHANG X H, TEHRANIPOOR M. A clock sweeping technique for detecting hardware Trojans impacting circuits delay [J]. **IEEE Design & Test**, 2013, 30(2): 26-34.

[6] 薛明富，王箭，胡爱群. 自适应优化的二元分类型硬件木马检测方法[J]. **计算机学报**，2017，40(95)：1-14.
XUE Mingfu, WANG Jian, HU Aiqun. Adaptive optimization of two-class classification-based hardware Trojan detection method [J]. **Journal of Computers**, 2017, 40(95): 1-14.

[7] SALMANI H, TEHRANIPOOR M, PLUSQUELLIC J, *et al*. A novel technique for improving hardware Trojan detection and reducing Trojan activation time[J]. **IEEE Transactions on Very Large Scale Integration Systems**, 2012, 20(1): 112-125.

[8] CHAKRABORTY R S, WOLFF F G, PAUL S, *et al*. MERO: A statistical approach for hardware Trojan detection[C]// **Proceedings of Cryptographic Hardware and Embedded Systems** (**CHES**). Lausanne, Switzerland: Springer, 2009: 396-410.

[9] SAHA S, CHAKRABORTY RS, NUTHAKKI SS, *et al*. Improved test pattern generation for hardware Trojan detection using genetic algorithm and boolean satisfiability[C]// **Proceedings of Cryptographic Hardware and Embedded Systems** (**CHES**). Saint-Malo, France: Springer, 2015: 577-596.

[10] LESPERANCE N, KULKARNI S, CHENG K, *et al*. Hardware Trojan detection using exhaustive testing of k-bit subspaces[C]// **Proceedings of Asia and South Pacific Design Automation Conference**. Chiba, Japan: IEEE, 2015: 755-760.

[11] XUE M F, HU A Q, LI G Y. Detecting hardware Trojan through heuristic partition and activity driven test pattern generation[C]// **Proceedings of Communications Security Conference** (**CSC**). Beijing, China: IEEE, 2014: 1-6.

[12] ZHOU Z Q, GUIN U, VISHWANI D, *et al*. Modeling and test generation for combinational hardware Trojans [C]// **Proceedings of Electrical Engineering Seminar Series**. San Francisco, USA: IEEE, 2018: 1-6.

[13] KARABOGA D. An idea based on honey bee swarm for numerical optimization [R]. Turkey: **Technical Report-TR06**, 2005: 1-10.

[14] EGGERSGLÜB S, DRECHSLER R. High quality test pattern generation and boolean satisfiablity[M]. Berlin: **Springer Science & Business Media**, 2012: 41-58.

（本文编辑：王一凡）