文章编号:1672-058X(2013)08-0059-05

云进化算法在 NoC 互连测试中研究*

余丙荣[1],张结龙[2],武昌俊[1],周明龙[1]
(1.安徽机电职业技术学院电气工程系,安徽 芜湖 241000;2. 武汉中原电子集团有限公司,武汉,430010)

摘 要:提出了一种基于云进化算法的 NoC 互连测试方案。该方案利用云模型对物种进化统一建模,重点解决云模型对进化的表示和对进化的控制两个问题,结合 NoC 互连测试问题,在功耗限制条件下,建立基于云进化算法的互连测试模型,以获取最优测试矢量集;实验结果表明:该算法取得了较好的测试效果,有效提高了测试效率。

关键词:云进化算法;NoC;互连测试
中图分类号:TP347 **文献标志码**:A

片上网络(NoC)作为全新的集成电路体系结构,采用全局异步-局部同步的通讯机制,显著改善了传统总线形式的片上系统(Systems-on-Chip,SoC)性能,被认为是下一代片上系统的首选结构。随着 NoC 规模的增加,复杂度的提高,测试费用大幅提高,测试开销在总开销中所占比例不断升高,甚至比设计和制造成本更高,因此 NoC 测试也就无疑成为一个重要研究对象,其中互连测试是 NoC 测试重点研究内容。目前,NoC 互连测试多采用内建自测试(Built-in Self Test,BIST)方法,如 Marcos B.Hervé,érika Cota 等采用 BIST 测试方法[1,2],完成测试矢量生成,但该测试方案存在不足,如算法测试效率不高、硬件开销大、功耗没有考虑等问题。本文在功耗限制条件下,采用 NoC 重用测试机制的方法,提出了基于云进化算法的 NoC 互连测试方案。针对 NoC 互连故障的特点,选取合适的测试端口和测试路径,旨在获取最优测试矢量集的同时,使得测试代价更小,测试效率更高。

1 云进化理论

1.1 云模型

云模型 $C(E_x,E_n,H_e)$ 是一种定性知识描述和定性概念与其定量数据表示之间的不确定性转换模型[3],云模型在知识表达时具有不确定中带有确定性、稳定之中又有变化的特点,其中正态云模型是最重要的一种云模型。

1.2 正态云发生器

正态云模型是一个遵循正态分布规律的,具有稳定倾向的随机数集。生成云滴的算法称为云发生器,正态云发生器生成过程如算法 1 所示。

收稿日期:2013-01-17;**修回日期**:2013-04-07.
* **基金项目**:安徽高校省级自然科学研究项目(KJ2012Z043).
作者简介:余丙荣(1971-),男,安徽安庆宿松人,副教授,硕士,从事智能控制研究.

算法1：正态云发生器[4]

INPUT:E_x,E_n,H_e,n %数字特征和云滴数

OUTPUT:{(x_1,m_1),…,(x_n,m_n)} %n 个云滴

FOR $i=1$ *to* n

{

$E_n'=RANDN(E_n,H_e)$ %生成期望值为%E_n,方差为 H_e 的正态随机数

$x_i=RANDN(E_x,E_n')$

$\mu_i=\exp(\frac{-(x_i-E_x)^2}{2(E'_n)^2})$

drop(x_i,μ_i)　%生成第 i 个云滴 x_i,u_i 为第 i 个云滴的确定度

}

1.3 云进化算法(CBEA)

云进化算法利用云模型 $C(E_x,E_n,H_e)$ 描述进化过程,其中 E_x 称为优良种子个体,体现祖先遗传优良特性;E_n 称为进化熵,控制遗传粒度;H_e 称为进化超熵,控制子代个体的凝聚离散程度和搜索半径广度。利用正态云发生器产生下一代云滴,产生云滴的过程能够刻画遗传变异,体现了自然界物种进化的基本原理[3]。

精英个体和进化代是算法自适应调整的重要参考依据[5]：

(1) 精英个体。精英个体是指进化过程中得到的适应能力最强的个体,分别为当代精英个体和跨代精英个体。

(2) 进化代。出现跨代精英的进化代称为非平凡进化代,没有出现跨代精英的进化代称为平凡进化代;两个跨代精英个体之间相隔的进化代数称为连续平凡代,是连续没有出现跨代精英个体的进化代数。连续出现跨代精英的进化代数称为连续非平凡代数。

2 NoC 互连测试问题分析

2.1 问题的提出

NoC 中互连故障主要有开路故障,呆滞性故障,桥接故障等,由于 NoC 互连测试中开路故障往往等价于呆滞性故障,并且呆滞性故障易于检测,因此本文主要研究桥接故障。假设有 n 条互连线,其中 k 条发生桥接故障,则所有可能的故障总数是从 n 中取 k 的组合：

$$C(n,k)=\frac{n!}{k!\ (n-k)!} \tag{1}$$

若两条互连线桥接时,即式(1)中 $k=2$,则所有可能故障数：

$$C(n,2)=\frac{n(n-1)}{2} \tag{2}$$

如果需要找出多条互连线的桥接故障,必须对 $k=2$ 到 $k=n$ 的所有故障求和：

$$\sum_{k=2}^{n}C(n,k)=2^n-n-1 \tag{3}$$

从式(3)可以看出,所有可能的故障总数将随 NoC 互连线的数目 n 按指数方式增长,显然 NoC 互连测试是 NP-hard 问题。当 n 很大时,很难用遍历搜索寻求最优解,为得到较优测试测试矢量,寻求一种合适的算法是关键。

云进化算法利用正态云发生器,自适应调整影响进化过程的参数,产生的云滴具有随机性和稳定倾向性的特点,算法既具有快速寻优能力,又能够解决进化方向的无记忆性和随机性问题。将云进化算法应用于 NoC 互连测试研究,用以解决互连测试难题。由于目前尚未检索到运用智能算法解决类似问题的文献,为使实验结果

具有可比性,且鉴于遗传算法(Genetic Algorithm,GA)为求解复杂系统优化问题提供一个通用的框架,不依赖于问题的具体领域,其应用广泛[6],因此本文选择遗传算法作为比较对象,分析两种算法优劣。

2.2 拓扑结构

拓扑结构对系统的性能和面积开销有显著影响,直接体现了节点在芯片中的分布和节点间连接方式,因此设计测试算法时需要充分考虑拓扑结构的特点。由于 2D-mesh 拓扑结构设计简单、实现和可扩展性比较好、便于分析和解决问题,是目前广泛应用的拓扑结构之一,因此本文选取 2D-mesh 结构作为测试研究对象,2D-mesh 拓扑结构如图 1(a)所示。

NoC 采用基于分组交换的传输形式,互连网络中传输的数据为数据包形式,如图 1(b)所示。数据包由头(header)、数据信息(payload)和尾(trailer)组成。为提高通信效率,将数据信息分割成更小数据段(flit),数据头部包括工作模式和路径信息,工作模式又分为测试模式和正常模式。路径信息采用如下方式编码:0 表示 x 方向,1 表示 y 方向。

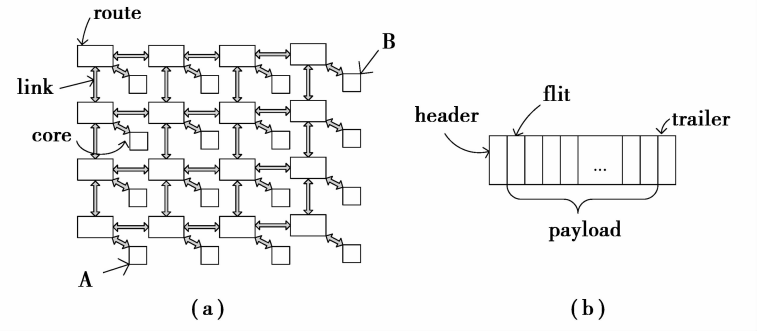

图 1　2D mesh 拓扑结构和数据包结构

2.3 测试功耗模型

本文从通信角度出发,针对 2D-mesh NoC 建立功耗模型[7],并将测试功耗作为算法约束函数,单比特数据从节点 1 传输到节点 2 的功耗由式(4)计算:

$$E_{\text{bit}}^{1,2} = (h_{1,2} + 1)E_{\text{router}} + h_{1,2}E_{\text{link}} \tag{4}$$

其中 E_{route},E_{link}分别表示在路由器和链路上消耗的能量,$h_{1,2}$表示节点 1 到节点 2 的步数。

2.4 测试策略

本文采用基于 NoC 重用机制的测试方法。首先对生成的测试矢量进行打包,打包后的数据以分组交换的方式在网络中传输,借助自动测试设备(Automatic Test Equipment,ATE),通过 intput/output 端口连接 NoC,由 intput 端口送入网络,并通过路由器送到被测试链路,测试响应同样由路由器送到 output 端口,由外部测试分析机制接收,这样完成一条链路测试。

NoC 测试时间的大小直接决定了测试成本的高低,针对 16×16 及以上的大规模 NoC 互连测试,本文采用分块并行测试方法,以减少测试时间。划分时需要设计好最小测试单元,如果测试单元太小,则并行测试次数增多,测试时间增大;如果测试单元太大,则会影响并行测试效果。根据测试经验,选取 4×4 作为最小测试单元。同时,鉴于对角节点在互连测试中具有优势[8],在此选其作为测试端口,与 ATE 相连,如图 1(a)所示,节点 A 和节点 B 分别作为 intput 和 output 端口。

3　云进化算法互连测试实现

测试生成研究过程中,首先需用云模型表示进化过程,结合互连测试,提出编码方案;其次针对互连测试实际问题,需制定算法调整策略,实现云模型对算法的控制;最后给出算法的具体的实现过程和算法参数取值。

3.1 编码方案

互连测试采用基于故障模拟的方法实现,测试矢量是二进制(0, 1)编码格式,则测试矢量空间表示如下:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

其中 m 表示种群大小,即测试矢量集的大小。n 表示测试矢量的长度,与 *NoC* 链路位宽相等。

云进化算法利用云模型 $C(E_x, E_n, H_e)$ 进行迭代寻优,更新种群,应用时,需要将云模型 $C(E_x, E_n, H_e)$ 与互连测试实际问题相结合。文中将 E_x 表示为最优测试矢量,由于该参数为实数形式,因此算法迭代操作时,需将二进制形式的互连测试矢量 $(a_{i1}, a_{i2}, \cdots, a_{in})$(其中 $i \in [1, m]$),按数值转化规则转为实数形式,迭代操作完成后再转为二进制形式。

3.2 算法调整策略

云进化算法能够自适应调整进化变异,因此根据进化过程中具体情况,制定进化调整策略和变异调整策略,获取最佳测试结果[5]。

(1) 进化调整策略。进化过程中重点解决局部求精和局部求变两个问题,通过动态调整云模型参数 E_n 和 H_e,优化子代种群产生。

局部求精:如果在进化过程中出现跨代精英个体,说明算法可以找到新的极值领域,或更加逼近老的领域,此时进行局部求精操作。可通过减少 E_n,降低遗传粒度;减少 H_e,以增大子代个体的凝聚离散程度和降低搜索半径,从而达到快速局部求精的目的。实现时将 E_n 和 H_e 减少为原来的 $1/K(K>1)$,K 为求精系数。

局部求变:如果在进化过程中,当代或者几代均未发现跨代精英个体,即连续平凡代数达到一定的阀值 K_{local},说明算法陷入局部最优,需要进行局部求变操作,跳出当前位置,寻找新的领域。实现时将 E_n 和 H_e 调高为原来的 L 倍,L 为求变系数,一般 $L \leqslant K$,取 $L = \lfloor \sqrt{K} \rfloor$。

(2) 变异调整策略。当经过若干进化代没有得到适应性更加优异的个体,且进化调整策略无效时,即连续平凡代数达到阀值 $K_{\text{global}}(K_{\text{global}} > K_{\text{local}})$,说明算法可能陷入局部,此时需要进行突变操作,跳出局部。本文通过取历史精英个体的平均值,作为种子个体的方法,产生下代种群。

3.3 算法实现过程

算法包含以下几个步骤:初始化;个体适应度评估;种群淘汰和种子选择;新种群的产生;循环上述步骤,直至满足停机条件,算法流程如图 2 所示。为使得测试功耗更低,本文对算法的进化单位和算法调整策略进行改进。

(1) 初始化。根据 NoC 电路规模和结构的特点,确定种群的大小,测试矢量长度以及种群的丰富度,随机生成种群个体。随机生成过程中,需要确保测试矢量每位取 0,1 的概率相等。

(2) 个体适应度评估。种群中个体优劣由个体检测到的故障数来衡量,利用故障模拟器对每个个体实施模拟,得到每个个体的适应值。测试过程中,测试矢量由对角节点 A 输入,测试响应由另一对角节点 B 输出,如图 1(a)所示,在功耗限制条件下,选择最短传输路径,将测试矢量打包,依次施加到并行故障模拟器[9]中进行故障模拟。

(3) 种群淘汰和种子选择。评估每个种群的平均适应度,按比例淘汰平均适应度最差的 m 个种群。在剩余种群中,找出适应度最好的 m 个个体作为种子个体。

(4) 新种群的产生。更新种群方法如下:

1) 云模型更新种群。每个种子个体作为云正态发生器算法 1 所示参数 E_x,根据算法调整策略,调节 E_n

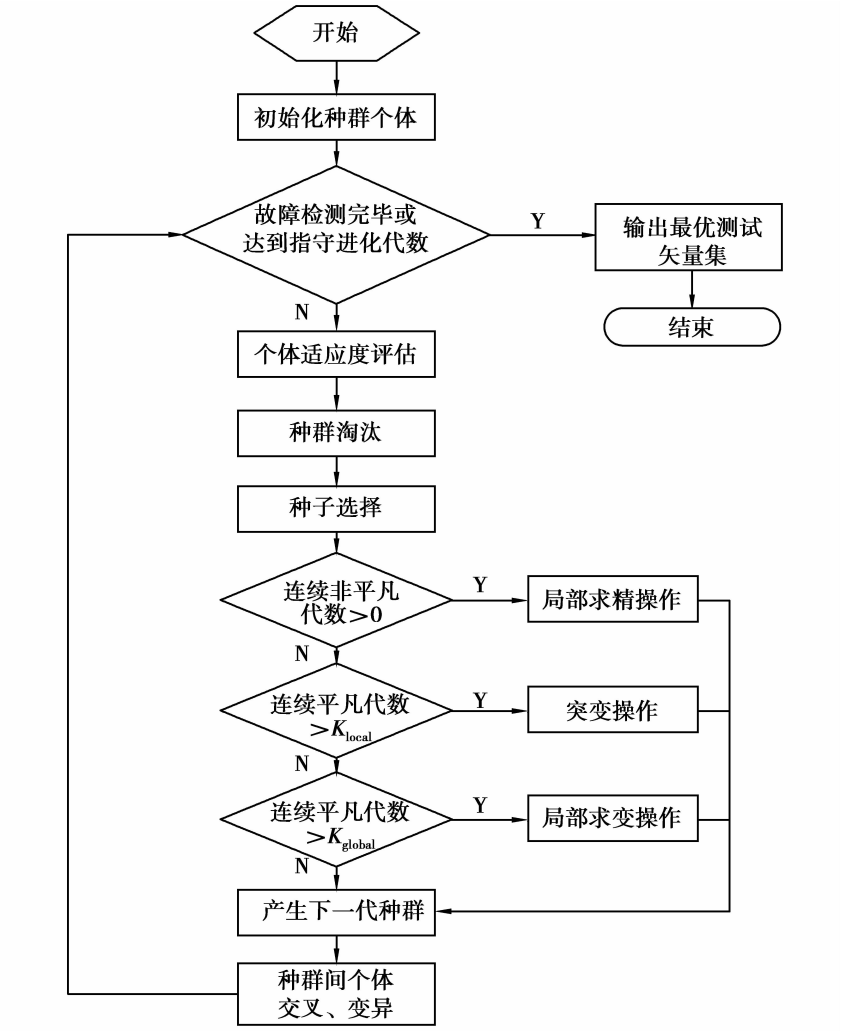

图 2　算法流程图

和 H_e,选择产生下代个体数目 n,更新种群。文中 m 个种子分别通过正态云发生器繁殖一个种群,替换被淘汰种群,实现种群的进化和变异。在此过程中针对互连测试的特点,对算法进行改进,改进如下:

① 算法进化单位。当 NoC 规模比较大时,传统的个体进化单位不再适应大规模寻优的需要,因此本文根据近代进化理论,将算法的进化单位改为种群,而不再是个体,从而有效提高算法更新效率。

② E_n 和 H_e 参数关系。在进化调整过程中,算法中 E_n 和 H_e 的关系及取值十分关键,直接影响了算法寻优效果,本文两者关系及取值并不局限于局部求精超熵控制表和局部求变超熵控制表[5] 中给定的参考值,而是结合互连测试中存在的问题,将 E_n 和 H_e 之间的关系调整为 $E_n=\lambda H_e$,同时经过多次调试,寻求最佳 λ 值。

2) 种群间个体交叉变异操作。对于运用正态云发生器产生的种群,需要按一定概率进行种群间个体交叉变异操作。交叉操作,防止种群近亲繁殖,有利于增强不同种群间基因交流;变异操作,有利于提高云滴的健壮性。至此新种群产生工作完成。

(5) 停机判断。如果进化代数达到最大的迭代次数 MAX,或故障列表中所有的互连故障已检测完毕,则终止迭代,否则转到第(2)步,继续进行迭代操作。

3.4　参数选取

不同的 NoC 系统,链路的互连线个数可能不同,本文在算法级验证时假设每条链路有 8 条互连线。文中针对 4×4,8×8,16×16 三种规模的 NoC 进行迭代寻优,3 种规模 NoC 的种群参数设置如表 1 所示。

表1 种群参数设置

NoC规模	种群规模	种群丰富度	功耗上限
4×4	20	24	144e
8×8	40	24	288e
16×16	60	24	432e

表1中功耗上限通过式(4)设定,其中e表示从一个通讯节点向它的相邻节点发送一个数据包所需要的平均功耗。根据实际调试经验,取$K_{local}=2$,$K_{global}=5$,$L=K=2$,$\lambda=100$,种群间个体交叉概率为0.6,个体变异概率为0.1,最大进化代数MAX=200。局部求精时,H_e取0.1,由$E_n=\lambda H_e$,则E_n取10;当局部求变时,相应的H_e调整为0.2,E_n为20。

4 实验结果与分析

本文采用SoCIN(SoC Interconnection Network,SoCIN)电路结构[10]验证算法性能。实验环境为AMD Athlon(tm)II X2,CPU 2.8 GHz,1 G内存,实验源代码采用C语言编写,并在Visual C++6.0编译环境仿真,其中正态云模型由MATLAB 7.0的mcc方式与VC混编实现,为验证CBEA性能,本文进行了算法执行效率实验和收敛性实验。

(1) 算法执行效率实验。用云进化算法(CBEA)和遗传算法(GA)分别对4×4、8×8、16×16三种规模的NoC进行迭代寻优,测试结果如表2所示。

表2 CBEA和GA测试结果

NoC规模	CBEA		GA	
	平均运行代数	测试生成时间(时钟周期)	平均运行代数	测试生成时间(时钟周期)
4×4	57	37656	84	49 062
8×8	96	131 094	121	141 093
16×16	135	1 474 703	—	—

从表2中可以看出,在最大进化代数限制条件下,针对4×4和8×8规模的NoC,CBEA和GA均能找到最优矢量集,但CBEA平均运行代数和测试生成时间均少于GA,优化率分别为23.3%和7.09%;针对16×16规模的NoC,在规定的最大进化代数范围内,CBEA在第135代找到最优矢量集,而GA没有找到。实验(1)表明,CBEA的执行效率要高于GA。

(2) 算法收敛性实验。为了说明CBEA的收敛性,本文对4×4规模的NoC分别用CBEA和GA各寻优100次,统计每次算法的运行结果,统计结果如图3所示。

从图3中可以看出,CBEA均在200代内完成收敛,且收敛代数集中在60代左右。GA有4次超过200代,且收敛代数分布随机。实验(2)表明,CBEA相对于GA具有稳定的收敛性。

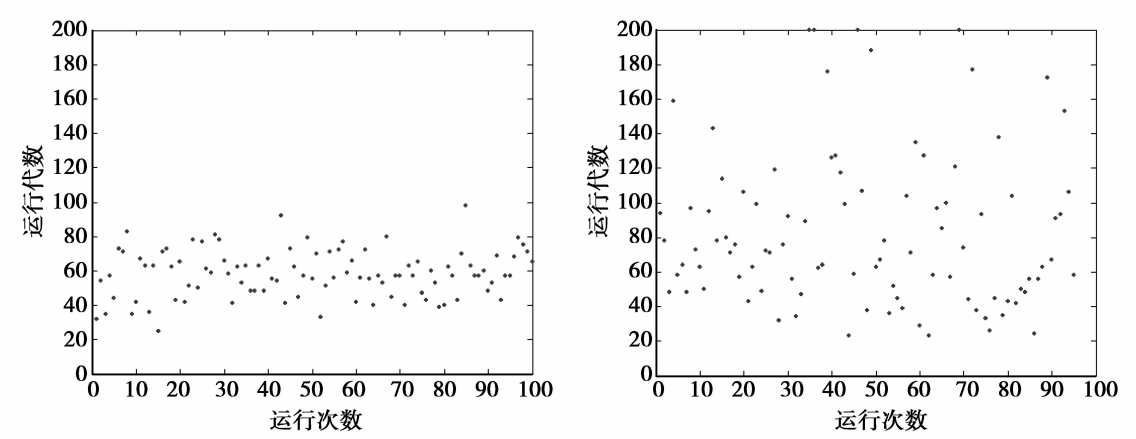

图 3　CBEA 和 GA 运行结果统计

5　结　语

将云进化算法应用于 NoC 互连测试,在功耗限制条件下,利用云模型较强的全局搜索能力和自适应调整能力,从而可以快速使算法收敛到最优,较好地避免了 GA 易陷入局部最优和选择压力过大造成的过早收敛等问题,算法取得了较好的测试效果,提高了测试效率。

参考文献:

[1] MARCOS B.HRIKA C, FERNANDA L. Kastensmidt, Marcelo Lubaszewski[A]. NoC Interconnection Functional Testing: Using Boundary-Scanto Reduce the Overall Testing Time[C].IEEE 10th Latin American Test Workshop (LATW '09), 2009,1-6

[2] COTA E, KASTENSMIDT F, CASSEL M, MEIRELLES P. Lubaszewski. Redefining and Testing Interconnect Faults in Mesh NoCs [D]. Proc. IEEE Int'l Test Conf. 2007,1-10

[3] 张光卫,何锐,刘禹,李德毅,陈桂生.基于云模型的进化算法[J].计算机学报.2008,31(7):1082-1091

[4] LI X. Study on Classification and Clustering Mining Bused on Cloud Model and Data Field [D]. Beijing: PLA University of Science and Technology,2003

[5] 刘禹,李德毅,张光卫,等.云模型雾化特性及在进化算法中的应用[J].电子学报,2009,37(8):1651-1658

[6] 田延硕,刘晓云. 一种提高局部搜索能力的混合遗传算法[J]. 电子科技大学学报,2006,25(2):232-234

[7] HU J, MARCULESCU R. Energy-aware communication and task scheduling for network-on-chip architectures under real-Time constraints[A].ProcDATE'04.Paris:IEEE.2004,234-239

[8] 欧阳一鸣,冯伟,梁华国.功耗限制下的 NoC 测试端口的优化选择方法[J].计算机应用,2008,28(4):1026-1031

[9] QIAOYAN YU, PAUL A. A Flexible ParallelSimulator for Networks-on-Chip with Error Control[C]. IEEE TRANSACTIONS ON COMPU-TER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, 2010,29(1):103-116

[10] 欧阳一鸣,黄河,梁华国. 功耗优先的 NoC 通讯架构测试方法[J].合肥工业大学学报,2010,33(10):1510-1514

Research on NoC Interconnect Test in Cloud Based Evolutionary Algorithm

YU Bing-rong[1], ZHANG Jie-long[2], WU Chang-jun[1], ZHOU Ming-long[1]

(1.Department of Electrical Engineering, Anhui Technical College of Mechanical and Electrical Engineering, Anhui Wuhu 241000;
2. Wuhan Zhong Yuan Electronics Technology Trading Co.,Ltd, Hubei Wuhan 430010,China)

Abstract: This paper presents a solution using Cloud Based Evolutionary Algorithm (CBEA) to solve the Network-on-Chip (NoC) interconnect test. With the cloud model, inheritance and mutation of the species can be modeled naturally and uniformly, the algorithm focuses on solving the two problems using the could model to express the evolution and to control the evolution. Considering of the NoC interconnect test actual problems and under the power constraints, this paper establishes the test mode of the CBEA, in order to find out the optimal test vectors set. Experimental results show that the algorithm achieved good test results and improved the test efficiency.

Key words: Cloud Based Evolutionary Algorithm; Network-on-Chip; interconnect test

责任编辑:代小红

(上接第58页)

Research on Theoretical Line Loss Rate Prediction Based on PSO-SVM Model

WANG Jing[1], TIAN Li[2], XIA Kun[3], HU Zhi-ying[4]

(1. School of Electronic Engineering and Electrical Automation, Chaohu University, Anhui Chaohu 238000, China;
2. School of Electrical Engineering, Anhui Polytechnic University, Anhui Wuhu 241000, China;
3. News Center, Chaohu Wuwei Electric Power Supply Co., Ltd, Anhui Wuhu 241000, China;
4. Nanjing Guolian Electric Power Engineering Design Co., Ltd, Nanjing 210009, China)

Abstract:This paper deeply analyzes the influential factors for line loss rate, studies present line loss rate prediction methods, uses particle swarm algorithm to optimize the parameters of support vector machine, sets up support vector machine prediction model based on particle swarm optimization to simulate theoretical line loss rate prediction in order to provide guarantee for reducing line loss rate and highly-efficient utilization of power and finally uses sample experiment to verify the accuracy of this model in theoretical line loss rate prediction.

Key words: line loss rate prediction;support vector machine;particle swarm optimization

责任编辑:代小红