

基于新型 PSO 算法优化 BP 神经网络的 软件缺陷预测方法研究

邓伟康,刘 锋,朱二周

(安徽大学 计算机科学与技术学院,安徽 合肥 230601)

摘 要: 针对传统的软件缺陷预测模型存在预测准确率低和适应性差的问题,本文提出了一种改进的 PSO 算法 (IVPSO),并将其与 BP 神经网络相结合,以此来构建一个新的、预测性能和效果更加优秀的模型---IVPSO-BP. 首先,对粒子群算法进行改进并利用其对 BP 网络进行优化;其次,基于优化的 BP 算法去建立一个预测模型;最后,将该模型与 PSO-BP 模型、J48(传统的机器学习方法)、BP 进行实验比较. 通过对最终实验的数据进行分析表明,IVPSO-BP 模型具有更高的寻优性能和准确性.

关键词: 软件缺陷预测;粒子群算法;神经网络

中图分类号: TP311

文献标识码: A

文章编号: 1000—7180(2017)04—0039—05

Software Defect Prediction Model Based on IVPSO-BP

DENG Wei-kang, LIU Feng, ZHU Er-zhou

(School of Computer Science and Technology, Anhui University, Hefei 230601, China)

Abstract: In view of the traditional software defect prediction model, the problem of low accuracy and poor adaptability, this paper proposes a new software prediction model (IVPSO-BP) with better performance and effect, and combining with two by improving PSO algorithm for optimizing BP. Firstly, To improve Particle Swarm Optimization, and use it to optimize BP. Secondly, this paper employs optimized BP to build a prediction model. Finally, compares the experiment results with other machine learning methods-BP, J48 and PSO-BP. Through analyzing the data of the final experiment, the results indicated that proposed method owe a higher prediction precision.

Key words: software defect detection; particle swarm optimization (PSO); artificial neural network (ANN)

1 引言

随着互联网时代的到来,市场对软件产品的质量要求也不断提高,软件的安全性、可靠性与人们的生命、财产安全息息相关,因此,对软件安全进行研究,确保为我们服务的每一个软件都是稳定与安全的.

现有的软件缺陷预测方法有线性判别 (Linear Discriminant Analysis, LDA)、支持向量机 (Support Vector Machine, SVM)^[1]、人工神经网络 (Artificial Neural Network, ANN)^[2]、分类回归树

(Classification and Regression Tree, CART)^[3]等. 但传统的预测模型往往是利用软件的缺陷相关的数据进行统计分析,多数存在一些问题. 例如,线性判别^[4]不适合对非高斯分布的样本进行降维,可能过度拟合数据;参数的选择对支持向量机非常重要,但是很难获得一个较好的方法来进行选择;分类回归树的缺点是泛化性能差;人工神经网络难以克服的缺点是其速度的收敛较慢以及会深陷极小值点. 目前,对软件的缺陷进行预测的方法非常多,它们都有其优点和缺点,而改进 BP 神经网络是现在非常常见和实用的做法.

收稿日期: 2016—08—28; **修回日期:** 2016—09—15

基金项目: 国家自然科学基金 (61300169)

本文针对传统 PSO (Particle Swarm Optimization, PSO)^[5] 算法容易陷入局部最优值导致典型的早熟收敛问题, 提出一种改进 PSO 的优化算法—IVPSO (Improved Variation PSO). 改进后的算法不但克服了自身的缺点, 还适用于优化 BP 网络.

2 相关工作

2.1 标准粒子群算法

粒子群算法 PSO 源于复杂适应系统 (Complex Adaptive System, CAS), 是一种具有随机性的搜索算法. 它被认为是群集智能 (Swarm intelligence, SI) 的一种, 是团队型的, 利用众人的“合力”来解决问题. 如在一个数量非常庞大的群体, 单个个体的能力是有限的, 但是如果大家齐心协力、系统的合作就能够解决很多复杂的问题, 对于优化方面的问题解决效果较好. 基本的粒子群算法有很多不足, 改进的点比较多, 本文的做法是采用新的惯性权重更新方法, 使其适用于优化 BP 网络做缺陷预测.

首先, 标准粒子群算法对粒子进行初始化, 即对粒子的初始位置和速度随机赋值, 然后通过对最优粒子的跟踪来获得最优解. 例如, 在一个广阔的多维的领域内, 设为 n 维, 一个有 m 个粒子的群体, 假定任意一个粒子 i , 其位置为 $X^i = (x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,n})$, 它的速度 $V^i = (v_{i,1}, v_{i,2}, v_{i,3}, \dots, v_{i,n})$, 当粒子不停的运动进行迭代时, 会不断的改变自己的状态, 而它的状态变化的依据是 P_{best} 和 G_{best} . P_{best} 是粒子自身追寻到的最优解和个体极值, G_{best} 是在种群中当前发现的最优解, 粒子在运动的过程中, 当发现 P_{best} 和 G_{best} 时, 便会以下面的公式对其速度及位置赋新的值.

$$v_{i,j}(t+1) = w \times v_{i,j}(t) + (c_1 \times r_1 + c_2 \times r_2) \times [p_{i,j} - x_{i,j}(t)] \quad (1)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \quad i = 1, 2, \dots, m; \\ j = 1, 2, \dots, n; \quad (2)$$

式中, w 对速度产生影响并赋予其一定的惯性, 被称为惯性权重; c_1 和 c_2 是学习因子, r_1 和 r_2 是 $[0, 1]$ 之间的随机数.

2.2 改进粒子群算法

在标准的 PSO 算法中, w 的值是一个重要同时可以改变的参数, 所以它是对算法改进的很重要的一个点. w 增大时, 粒子的速度也会随之变大, 此时利于全局搜索; w 减小时, 速度也会逐渐变小, 此时利于局部搜索. 如何对 w 的值进行控制, 使其利于

对问题的解决, 是研究的热点. Shi 提出的方法是线性递减惯性权重 (Linearly Decreasing Inertia Weight, LDW)^[6], 即以线性的方式对 w 进行改变以达到提升算法的收敛性, 如式 (3) 所示:

$$w = w_{\text{max}} - \frac{t \times (w_{\text{max}} - w_{\text{min}})}{t_{\text{max}}} \quad (3)$$

式 (3) 中, w 的取值有一定的范围 $[w_{\text{min}}, w_{\text{max}}]$, t 表示当前迭代的次数, t_{max} 是其所能取得的最大值.

但是 w 以线性方式递减, 会在初期使收敛速度降低, 而后由于 w 的减小, 会削弱算法的多样性, 出现局部最优. 本文采用的是一种非线性权重的 IPSO^[7] 方法来改善标准 PSO 算法的不足, w 的更新公式如下:

$$w = w_{\text{max}} - (w_{\text{max}} - w_{\text{min}}) \times \arcsin\left(\frac{t}{t_{\text{max}}} \times \frac{\pi}{4}\right) \quad (4)$$

根据式 (4) 式可知, 在 t 比较小时候, w 近似等于 w_{max} , w 取值大时利于全局搜索; 随着 t 的不断增大, w 非线性的减小, w 值较小时能保证算法具有很好的局部搜索能力, 这样能够很灵活调整全局搜索和局部搜索.

另外, 本文在原有的 PSO 算法中, 加入遗传算法的变异^[8]思想. 随着算法不断迭代的, 种群运动搜索的空间会逐渐缩小, 引入变异操作, 能够改变这种状况, 使粒子能够进入到更加广阔的空间, 脱离原先的最优位置, 这样就保证了算法的泛化能力.

2.3 神经网络

目前, 已经存在的神经网络模型有很多, 它的应用领域十分广泛, 而在常见的一些模型中, 多数采用 BP 神经网络以及改进的形式, 它是一种按误差反向传播的, 带有教师监督的典型多层前向的学习算法.

良好的学习规则是算法学习能力的基础, 梯度下降法规约 BP 神经网络如何进行学习, 它是一种调整减小网络的总误差有效的方法. 一般情况下, 采用三层结构的网络就可以很好的用于解决优化设计方面的问题, 因此本文选用三层的神经网络, 即包括输入、隐含及输出三层. 当有信号输入时, 且信号传播是正向的, 同一层神经元的状态只会受到上一层的影响. 若在输出层没有获得期望的输出^[9], 则会出现向反方向传输, 在反向传输的过程中各个层神经元的值会被不断调整, 这样误差信号就会降低到最小, 整个过程描述如下:

(1) 网络初始化

网络的初始化有:每一层都会有各自权系数 w_{ij} ,一般随机给定一个很小的非零数;分别对各层的阈值初始化;确定学习速率以及神经元的激励函数。

(2) 计算各层的输出

首先输入一个样本 $X = (x_1, x_2, x_3, \dots, x_n)$, 以及一个输出样本 $Y = (y_1, y_2, y_3, \dots, y_n)$. 根据公式 (5) 和 (6) 计算各层神经元的输出。

$$H_i = f\left(\sum_{j=1}^m w_{ij} - a_i\right) \quad i = 1, 2, 3, \dots, q \quad (5)$$

$$O_k = \sum_{i=1}^q H_i w_{ki} - b_k \quad k = 1, 2, \dots, L \quad (6)$$

$$f(x) = \frac{1}{1 + e^{-x}} \quad (7)$$

式中, H 是隐含层输出, i 是其节点数, a 是其阈值, f 是其激励函数; w_{ij} 是输入层和隐含层间的连接权值; O 为输出层的输出, b 是它的阈值, w_{ki} 为隐含层与输出层间的连接权值。

(3) 计算网络的输出 O 和期望输出 O' 的误差 e

$$e_k = O'_k - O_k \quad k = 1, 2, \dots, L; \quad (8)$$

(4) 网络的连接权值更新和阈值更新

$$w_{ij} = w_{ij} + \mu H_i (1 - H_i) x(j) \sum_{k=1}^L w_{ki} e_i$$

$$j = 1, 2, \dots, m; i = 1, 2, \dots, q; \quad (9)$$

$$w_{ki} = w_{ki} + \mu H_i e_k$$

$$i = 1, 2, \dots, q; k = 1, 2, \dots, L; \quad (10)$$

$$a_i = a_i + \mu H_i (1 - H_i) x(j) \sum_{k=1}^L w_{ki} e_i$$

$$j = 1, 2, \dots, M; \quad (11)$$

$$b_k = b_k + e_k \quad k = 1, 2, \dots, L; \quad (12)$$

可以根据网络输出误差 e , 利用上述的公式对 w_{ij} 和 w_{ki} 的值进行调整, 同时 a 和 b 的值也可以根据 e 进行调整, μ 是学习速率。

(5) 当输出误差不符合要求, 返回步骤 (2), 直到输出的结果满足终止条件。

3 基于 IVPSO-BP 方法软件缺陷模型的建立

3.1 IVPSO 优化 BP 神经网络

本文的 IVPSO 算法对网络的权值和阈值具有很好的优化效果, 建立的基于 IVPSO-BP 模型, 即克服了它的缺点同时又提高了其泛化能力。在算法结束时能够寻到全局最优附近的点, 而 BP 算法又有比较强的局部寻优能力, 进行局部寻优, 从而达到了网络训练的目的。其过程如下:

(1) 初始化粒子群

首先, 设定种群的规模为 30, 对每个粒子的位置、速度以及个体最优解和群体最优解任意附一个初始值; 然后, 权重 w 的值取 $w_{\max} = 0.9, w_{\min} = 0.4, c_1$ 和 c_2 的值取 2, 设定 $T_{\max}$ 为大进化代数。

(2) 设置粒子群的适应度函数

式 (13) 是 BP 网络的均方差 (Mean Square Error), 本文选取它作为粒子群的适应度函数:

$$E = \frac{1}{M} \left(\sum_{k=1}^M \sum_{j=1}^{n_0} (y_{kj} - \overline{y_{kj}})^2 \right) \quad (13)$$

式中, y_{kj} 是样本 k 的理论输出, $\overline{y_{kj}}$ 是其实际输出, M 表示样本的数目, n_0 是输出层神经元个数。

(3) 将粒子的位置和速度映射给 BP 网络, 计算每个粒子适应度。

fon₂ 函数输入 $\langle x(i, :), \text{inputnum}, \text{hiddenmim}, \text{outputnum}, p_l, t_l \rangle$, 参数 $(x(i, :))$ 是粒子的位置, inputnum、hiddenmim、outputnum 表示输入、隐含、输出三层的节点数, 适应度计算如下:

fitness(i) = fon₂ ($x(i, :), \text{inputnum}, \text{hiddenmim}, \text{outputnum}, p_l, t_l$);

如果 fitness(i) < fitness(i) 则 $p_{\text{best}} = x(i, :)$ & fitness(i) = fitness(i);

如果 fitness(i) < fitnesspg 则 $g_{\text{best}} = x(i, :)$; fitnesspg = fitness(i);

其中, fitness(i) 是粒子的当前适应值, fitness(i) 是粒子的个体最优适应值, fitnesspg 是粒子的全局最优适应值, p_{best} 是粒子个体最优值, g_{best} 是种群全局最优值, p_1 输入的数据, t_1 是输出数据。

(4) 将当前的迭代次数与最大值比较, 如果不大于 $T_{\max}$, 算法不会停止, 返回到步骤 (2), 重复执行, 如果满足, 直到满足终止条件为止。

3.2 IVPSO 优化 BP 的软件缺陷预测模型

基于 IVPSO 优化 BP 的预测模型的建立主要通过以下几个步骤:

(1) 获取用于训练的数据集。

(2) 基于训练的数据样本, 建立模型。

(3) 利用 IVPSO 算法对 BP 网络进行优化。

(4) 当模型建立好时, 用测试用例进行检测, 获得分类预测的结果, 如果结果满足设置的条件, 则此前的工作成功完成, 获得合适的模型; 否则, 返回 (3), 继续优化模型, 直到满足条件为止。

根据以上 4 个步骤, 最后得到的预测模型图, 如图 1 所示。

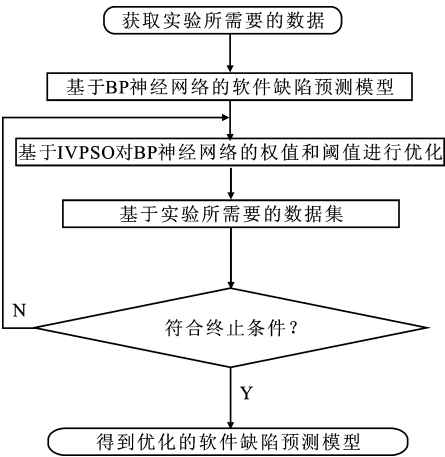


图 1 软件缺陷预测模型

4 基于 IVPSO-BP 软件缺陷预测模型实验

4.1 实验环境和实验数据

实验所用算法是在 MATLAB7.0 平台上实现的,电脑的硬件配置为双核 Pentium® Dual-Core CPU E6700 3.20 GHz 3.2 GHz,2 G 内存.

实验所用到的数据来源于美国国家航天局(NASA)的一个数据包,可以直接从网上获得,它详细的记录了每个软件项目的模块数、问题模块数及各个模块的属性等信息,有 13 个数据集,具体的内容如表 1 所示.

表 1 实验数据

数据集	编程语言	代码规模	模块数	缺陷模块
CM1	C	20k	505	48
JM1	C	315k	10 878	2 102
KC1	C++	43k	2 107	325
KC3	Java	18k	458	43
KC4	Perl	25k	125	75
MC1	C++	63k	9 466	68
MC2	C	6k	161	52
MW1	C	8k	403	31
PC1	C	40k	1 107	76
PC2	C	26k	5 589	23
PC3	C	40k	1 563	160
PC4	C	36k	1 458	178
PC5	C++	164k	17 186	516

软件是否安全,首先要对其包含的各个模块进行测试,判定模块是否有问题,依据其属性 Error_Count 判断是否是缺陷模块,当 Error_Count 的值小于 1 时,认为该模块是安全的,否则,就认为该模

块是不安全的,把它划入缺陷模块队列.

4.2 运行实验

本文从数据包中选取了 PC1、CM1、JM1、KC3、MW1、MC1,为减少分类的时间,对于模块比较多的数据集进行了适当的减少,对应的数据集模块选取为 300、200、500、300、200、200.

对该模型的性能进行测试,本文采用的评价是依据准确度、查准率、查全率和 F1 值,并与传统的机器学习算法进行了对比.如表 2 所示的交叉矩阵.

表 2 交叉矩阵

实际值	预测值	
	容易出错模块	不易出错模块
容易出错模块	正确的正例(TP)	错误的负例(FN)
不易出错模块	错误的正例(FP)	正确的负例(TN)

表 2 中,正例数目 $P=TP+FN$,负例数目 $N=FP+TN$,实际的总数目 $C=P+N$.

模型的评价标准定义如下:

准确度(accuracy)表示预测正确的实例的个数占总的测试实例的比例.计算公式如(14)所示:

$$accuracy=\frac{TP+TN}{C}$$
 (14)

查准率(precision)表示预测值为容易出错的模块中,正确的正例个数占总的正例数的比例.计算公式如(15)所示:

$$precision=\frac{TP}{TP+FP}$$
 (15)

查全率(recall)表示易出错且被正确预测的正例模块数占总的正例数的比例.计算公式如(16)所示:

$$recall=\frac{TP}{P}$$
 (16)

F_1 值表示查全率和查准率的调和平均.计算公式如(17)所示:

$$F_1=\frac{1}{\frac{1}{precision}+\frac{1}{recall}}$$
 (17)

4.3 实验结果及分析

实验是将六个数据集在 BP、J48、PSO-BP 和 IVPSO-BP 不同的 4 种算法上进行运行,通过实测的实验结果比较出哪个预测模型性能最好.图 2~图 7 分别为几种算法在 PC1、CM1、JM1、KC3、MW1、MC1 数据集上运行平均值,能够从图中很明显的看出各个算法在不同数据集以及在同一数据上预测能力,能够很容易判别出哪种算法的方法预测结果最好.

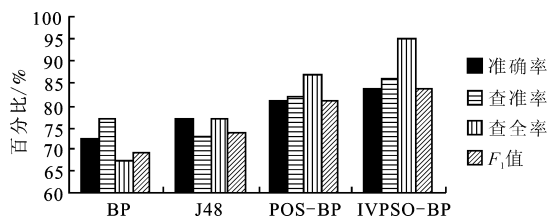


图2 JM1数据集预测结果

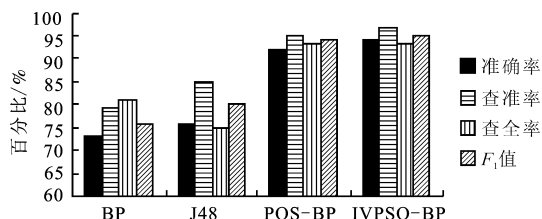


图3 CM1数据集预测结果

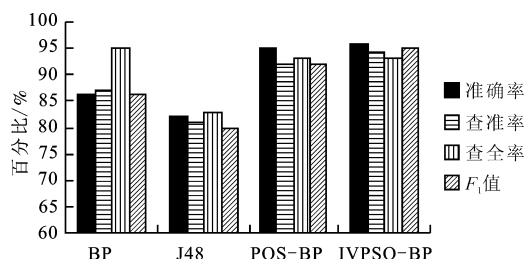


图4 KC3数据集预测结果

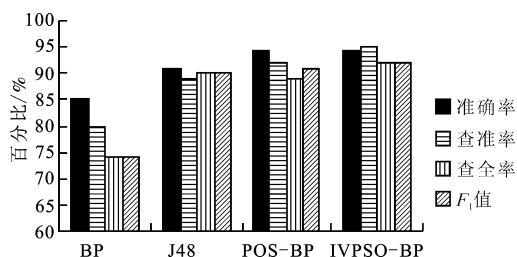


图5 PC1数据集预测结果

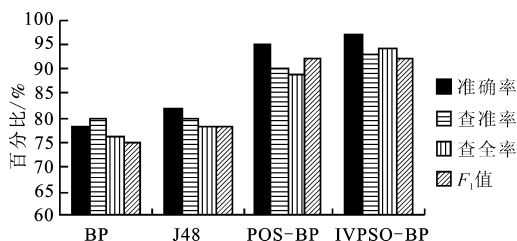


图6 MC1数据集预测结果

图2~7中的实验结果表明,BP算法和J48算法整体的预测性能非常明显都低于 PSO-BP 和 IVPSO-BP 算法.在图4出现 BP 算法的性能明显好于 J48,而图5结果相反,这可能与数据集有关,显然,预测性能最好的是本文提出的预测算法.

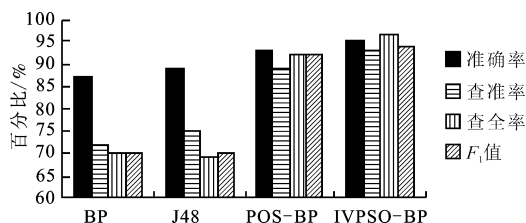


图7 MW1数据集预测结果

从上图可以看出,其中 BP 神经网络性能最差,主要因为其学习效率低,需要多次训练,没有一个有效的网络选择结构,而且泛化能力也比较差.相比之下,J48 算法较好,但也受到容量的限制.文献[10]中提出的 PSO-BP 方法是将两种算法相结合,而本文改进 PSO 的方法进一步对 BP 算法进行优化,获得的新模型的泛化性能更好,预测能力要明显优于传统的 PSO-BP 方法.

5 结束语

本文针对传统的 PSO-BP 方法的缺点,提出了基于改进粒子群算法的改进算法,解决了算法收敛慢和容易陷入局部最优的问题.对粒子群进行改进,建立了 IVPSO-BP 模型,并与 BP 网络、J48 和 PSO-BP 方法相比较,通过实验证明了本方法预测的有效性和更高的准确率.但在对权值和阈值寻优的过程耗时长,如何进一步优化模型提高预测准确率,并使降低模其运行时间,这是下一步要做的工作.

参考文献:

- [1] 姜慧研,宗茂,刘相莹.基于 ACO-SVM 的软件缺陷预测模型的研究[J].计算机学报,2011,34(6):1148-1154.
- [2] Fast M, Assadi M, De S. Development and multiutility of an ANN model for an industrial gas turbine[J]. Applied Energy,2009,86(1):9-17.
- [3] Tang LiJuan, Zhou YanPing, Jiang JianHui, et al. Radial basis function network-based transform for nonlinear support vector machine as optimized by particle swarm optimization algorithm with application to QSAR studies[J]. Journal of Chemical Information and Modeling, 2007, 47(4):1438-1445.
- [4] 杨健,杨静宇,叶晖. Fisher 线性鉴别分析的理论研究及其应用[J].自动化学报,2003,29(4):481-493.
- [5] 杨维,李歧强.粒子群优化算法综述[J].中国工程科学,2004,6(5):87-93.

通过将上述两张表的结果进行对比,可以看出在将 LUT 技术引入到组合电路中进行测试时,加入 LUT 技术的组合电路的故障覆盖率稍微低于没加入 LUT 模块时的故障覆盖率;同样地,加入两个 LUT 模块的组合电路故障覆盖率略微低于加入 1 个 LUT 模块时组合电路的故障覆盖率,这是因为加入 LUT 模块后尤其是加入 LUT 模块的数量增多时,组合电路中总的故障数量也比之前增多,并且在回溯数量受到限制时,由于电路复杂度增加,对单固定型故障进行测试产生时,需要回溯的次数也再增加,中止的故障数量也在不断变化,因此加入 LUT 模块后故障覆盖率会有所变化.但是总体上,加入 LUT 模块后,对组合电路进行测试诊断时故障覆盖率仍在 90%以上,可以检测到其他类型的故障,因此在总体上是满足要求的.

5 结束语

针对 FPGA 的设计,本文提出了将 LUT 引入 ATPG 算法的技术,并将这种技术整合到 ATALANTA 开源软件中,对包含了 LUT 的组合电路进行测试验证时,这种改进型的 ATALANTA 软件完全达到预期目标,可以检测出各类 LUT 的故障.因此,这种直接操作 LUT 的 ATPG 方法,不仅减少了设计和测试验证人员的工作量,也极大地减少了对待测试电路的测试时间.

参考文献:

[1] 何曦. 基于优化 ATPG 的可测试性设计与实现[D]. 成都:电子科技大学,2016.

[2] 曾文斌. 组合逻辑电路的测试故障模型[J]. 计算机学报,1990,12(7):529-538.

[3] 周宁. 集成电路等价性验证系统设计及其实现[D]. 兰州:兰州大学,2007:33-42.

[4] 岳园. 基于逻辑锥和 SAT 的带黑盒电路等价性验证方法[D]. 兰州:兰州大学,2007:19-31.

[5] Lee H K, Ha D S, On the generation of test patterns for combinational circuits[R]. USA:Dept of Electrical Eng. , Virginia Polytechnic Institute and State University. Technical Report No. 12_93,2002.

[6] 范逸鸣. 基于 LUT 的 FPGA 时序仿真的设计与实现[D]. 西安:西安电子科技大学,2012:7-18.

[7] Hansen M, Yalcin H, Hayes J P. Unveiling the ISCAS-85 benchmarks: a case study in reverse engineering[J]. IEEE Design and Test, 1999, 16(3): 72-80.

[8] 杨世元,胡梅,王红. 模拟电路软故障诊断的研究[J]. 微电子学与计算机,2008,25(1):1-6.

作者简介:

王志远 男,(1989-),硕士研究生. 研究方向为等价性验证算法研究. E-mail:wangzhiyuan14@lzu.edu.cn.

李彩虹 女,(1973-),博士,高级实验师. 研究方向为形式化验证、电网预测、机器学习、无线传感器网络等.

何安平 男,(1981-),博士,副教授. 研究方向为模型检测、等价性验证、异步电路系统相对时序分析等.

(上接第 43 页)

[6] Shi Y H, Eberhart R C. A modified particle swarm optimizer[C]//IEEE International Conference on Evolutionary Computation. Anchorage,1998:69-73.

[7] 陶海龙,李小平,张胜召,等. 基于 IPSO-BP 神经网络的铁路客运量预测[J]. 铁道运输与经济,2011,33(9): 78-82.

[8] 高岳林,任子晖. 带有变异算子的自适应粒子群优化算法[J]. 计算机工程与应用,2007,43(25):43-47.

[9] 王青,伍书剑,李明树. 软件缺陷预测技术[J]. 软件学报,2008,19(7):1565-1580.

[10] 余安雷,皮特常. 基于 PSO-BP 的软件缺陷预测模型[J]. 计算机工程与应用,2013,49(7):64-67.

作者简介:

邓伟康 男,(1990-),硕士研究生. 研究方向为大数据与云计算.

刘 锋 男,(1962-),教授,硕士生导师. 研究方向为并行计算与云计算.

朱二周(通讯作者) 男,(1981-),副教授,硕士生导师. 研究方向为虚拟化与程序分析. E-mail:ezzhu@ahu.edu.cn.