·计算机工程与应用·

采用HC-MARPSO算法的软件测试数据生成方法

雷 航，韩 炫

(电子科技大学计算机科学与工程学院 成都 611731)

【摘要】在吸引排斥粒子群算法(ARPSO)基础上，引入新的种群多样性度量指标和排斥操作，提出改进的吸引排斥粒子群算法(MARPSO)。结合爬山算法(HC)的局部收敛能力和改进的吸引排斥粒子群算法避免早熟的特点，提出基于爬山算法和改进吸引排斥粒子群算法(HC-MARPSO)的软件测试数据自动生成方法。实验结果表明，该算法在生成测试数据的效率上高于遗传算法、粒子群算法。

关 键 词 测试数据自动生成； 爬山算法； 粒子群算法； 软件测试； 吸引排斥粒子群算法

中图分类号 TP311 文献标识码 A doi:10.3969/j.issn.1001-0548.2012.06.013

Software Test Data Generation Method Using Hill Climbing Algorithm Combined with a Modified ARPSO

LEI Hang and HAN Xuan

(School of Computer Science and Engineering, University of Electronic and Science Technology of China Chengdu 611731)

Abstract A modified attractive and repulsive particle swarm optimization (MARPSO) algorithm is proposed, which is based on the attractive and repulsive particle swarm optimization (ARPSO) algorithm, by employing new diversity-measure and repulsive operator. Combining both the local convergence ability of hill climbing(HC) algorithm and the characteristic avoid precocious of MARPSO, the way of automatic generation of the software test data based on hill climbing algorithm combined with MARPSO(HC-MARPSO) is proposed. Finally, the results of experiment show that this new algorithm can generate test data more effective than other algorithms, such as genetic algorithm and particle swarm optimization algorithm.

Key words automatic test data generation; hill climbing algorithm; particle swarm optimization algorithm (PSO); software test; the attractive and repulsive particle swarm optimization algorithm (ARPSO)

软件测试是保证软件质量和可靠性的重要手段，在软件开发周期中所占比重日益增大。软件测试的许多操作和过程具有重复性，自动化方法的实现有利于提高软件测试效率，测试数据自动生成问题成了软件测试的基本问题。国内外关于测试数据自动生成方法的研究有很多，大致可以分为随机法、静态法、动态法和试探法4类。静态法中主要有符号执行法和区间算术法等。动态法中主要有程序插装、迭代松弛法等[1]。

近几年来，测试数据自动生成方法主要集中在试探法，利用不同的智能搜索算法解决测试数据自动生成的问题。文献[2]对基于遗传算法测试数据生成方法进行了研究，比较不同适应度函数对测试数据生成效率的影响，得出基于分支谓词的适应度函数比基于扩展哈明距离的适应度函数更加有效的结论。文献[3]针对遗传算法在面向路径测试数据生成中的应用，提出了一种新的适应度函数，认为当前路径与目标路径有更多重叠部分时有更高的适应度。文献[4]把克隆选择算法应用于面向路径的测试数据生成，并与遗传算法GAHD、GAFM比较，表明其更加有效。文献[5]介绍了蚁群算法和遗传算法在测试数据生成上的应用，分析了两种算法间的关系，比较了各自的测试数据生成效率，结果表明蚁群算法有更好的表现。文献[6]将粒子群优化算法应用在测试数据生成上，提出新的适应度函数，提高了粒子群优化算法的效率。

粒子群优化算法与遗传算法、蚁群算法相比，原理简单易于实现，需要调整的参数少，同时具有

收稿日期：2011 – 03 – 11；修改日期：2011 – 09 – 19
基金项目：国家核高基项目(2009ZX01039-003-001)
作者简介：雷航(1960 –)，男，博士，教授，主要从事软件测试、软件可靠性工程方面的研究.

很好的全局优化性能，但是容易陷入局部最优。文献[7]在基本粒子群优化算法的基础上，提出吸引排斥粒子群算法(ARPSO)。该算法能保证种群多样性，避免早熟现象，但其收敛速度缓慢且存在不收敛的问题。本文在ARPSO算法的基础上进行改进，提出改进的吸引排斥粒子群算法(MARPSO)，提高了算法的收敛速度；结合爬山算法与MARPSO算法的特性，提出了基于HC-MARPSO的软件测试数据自动生成方法。

1 爬山算法和MARPSO的提出

1.1 爬山算法

爬山算法[8]是一种常用的寻找局部最优解的方法，其基本原理是一种启发式搜索策略：设爬山者最初位于P点，目标是爬上峰顶。爬山有4种选择，即向东、向南、向西和向北。爬山者每走一步前先分别计算4个方向所到达新位置与原位置的高度差，即启发信息，然后根据该信息决定向何方走，一般选取高度差最大的方向为将走的方向。当到达某点，若4个方向的高度差计算结果都导致高度下降，则认为该点就是峰顶，搜索结束。

由于爬山算法每一步都是选梯度最大的方向前进，而非盲目攀登，因此能快速到达峰顶。但爬山算法有局限性，如果山多峰且起始点P处于非主峰区域时，就难以找到真正的顶峰。

1.2 基本粒子群优化算法

粒子群优化算法最早由文献[9]提出，其基本概念源于对鸟群觅食行为的研究，是进化计算和智能优化邻域的又一个研究热点，近年来已在约束优化、自动控制、模式识别与图像处理等领域得到了广泛应用。该算法通过群体中个体之间的协作和信息共享实现最优解的搜索，主要步骤为：首先生成初始种群，即在搜索空间中随机初始化一群粒子，使每个粒子都是优化问题的一个可行解，并为每个粒子确定一个适应度。被确定适应度的粒子在运动过程中，有一个速度决定其运动距离和方向。每个粒子将追随当前的最优粒子经过逐代搜索得到最优解。在每一代搜索的过程中，每个粒子都会跟踪两个极值，这是因为粒子本身是找到的最优解，也是整个种群找到的最优解。粒子群优化算法的基本原理就是加速每个粒子朝它自己所经历的和种群所经历的最好位置逼近。

基本粒子群优化算法的一般数学表示为：设在一个n维空间中，有一个包含m个粒子的种群，即$\boldsymbol{X}=\{X_1,X_2,\cdots,X_m\}$，种群中第$i$个粒子位置为$\boldsymbol{X}_i=\{X_{i1},X_{i2},\cdots,X_{in}\}^{\mathrm{T}}$，其速度是$\boldsymbol{V}_i=\{V_{i1},V_{i2},\cdots,V_{in}\}^{\mathrm{T}}$，第$i$个粒子的个体极值为$\boldsymbol{P}_i=\{P_{i1},P_{i2},\cdots,P_{in}\}^{\mathrm{T}}$，种群的全局极值是$\boldsymbol{P}_g=\{P_{g1},P_{g2},\cdots,P_{gn}\}^{\mathrm{T}}$，依据粒子群优化算法的基本原理，种群中第$k$+1代的第$i$个粒子将依据式(1)、式(2)更新自己的速度和位置：

$$v_{ij}(k+1)=\omega v_{ij}(k)+c_1r_1(p_{ij}-x_{ij}(k))+c_2r_2(p_g-x_{ij}(k)) \tag{1}$$

$$x_{ij}(k+1)=x_{ij}(k)+v_{ij}(k+1) \tag{2}$$

式中，j=1,2,$\cdots$,n；ω为惯性权重，可随迭代线性变化；c_1和c_2为加速因子，通常为非负常数；r_1和r_2为[0,1]范围内的均匀随机数；$v_{ij}\in[-v_{\max},v_{\max}]$，$v_{\max}$为速度阈值。

从式(1)可知，粒子i主要通过3部分来更新自己的速度：1) “惯性”部分，表示粒子当前速度受到上次速度的影响；2) “自身认知”部分，表示粒子当前速度受到自身经验的影响；3) “群体认知”部分，表示粒子当前速度受到群体中最好经验的影响。

1.3 吸引排斥粒子群优化算法

为了解决基本粒子群算法易陷入局部最优的缺点，文献[7]提出了ARPSO算法。在基本粒子群速度更新公式中加入符号变量dir，根据种群多样性指标diversity调整加速因子的正负号，动态地改变“吸引”和“排斥”状态，以改善算法过早收敛于局部最优的问题。

定义 1 种群多样性指标函数为：

$$\text{diverstiy}=\frac{1}{m|L|}\sum_{i=1}^{m}\sqrt{\sum_{j=1}^{n}(P_{ij}-\tilde{P}_j)^2}$$

式中，m是种群规模；$|L|$表示搜索空间的最大对角线长度；n是解空间维数；P_{ij}表示当前第i个粒子的个体极值的第j维分量；$\tilde{P}_j$是当前种群中所有个体极值的第j维分量的均值。

吸引排斥粒子群的速度更新公式为：

$$v_{ij}(k+1)=\omega v_{ij}(k)+\text{dir}\times(c_1r_1(p_{ij}-x_{ij}(k))+c_2r_2(p_g-x_{ij}(k)))$$

$$\text{dir}=\begin{cases}-1 & \text{dir}>0\text{且diversity}<\text{dLow}\\ 1 & \text{dir}<0\text{且diversity}>\text{dHigh}\end{cases} \tag{3}$$

式中，dLow和dHigh分别是种群多样性指标函数的阈值。当种群多样性指标低于dLow时，dir为–1，粒子群进行排斥操作，当前粒子远离个体极值和全局极值；当种群多样性指标高于dHigh时，dir为1，粒子群进行吸引操作，当前粒子向个体极值和全局极值收敛。

1.4 改进的吸引排斥粒子群算法

粒子群算法的早熟现象是由于初期群中粒子过于集中在局部最优而失去搜索其他位置的能力。ARPSO算法引入排斥阶段，能保持种群多样性，改善了基本粒子群优化算法早熟的问题。ARPSO算法利用个体极值间的聚集程度来度量种群多样性，该指标能较好地反应种群的覆盖率，但运行中存在不收敛问题。当diversity<dLow时进行排斥操作，粒子跳出局部最优解，却不能保证粒子可以找到新的最优解来更新个体极值和diversity，从而导致排斥操作会无限制进行下去。本文在ARPSO算法基础上，提出MARPSO算法。MARPSO算法采用新种群多样性指标函数newdiversity，该函数利用种群中粒子当前位置到全局极值P_g的距离分布来近似度量群体的多样性，它可以避免上述问题，同时降低了算法的计算量。

定义 2　新种群多样性指标函数为：

$$\text{newdiverstiy}=\frac{1}{m|L|}\sum_{i=1}^{m}\sqrt{\sum_{j=1}^{n}(X_{ij}-P_{gj})^2} \tag{4}$$

式中，X_{ij}表示当前第i个粒子的第j维分量；P_{gj}表示当前全局极值的第j维分量。

吸引排斥粒子群在种群多样性指标低于dLow时持续执行排斥操作，直到种群多样性指标高于dHigh时再进行吸引操作。通过更改dir函数式(3)可以提高扩散速度，减少算法参数，同时提高算法效率。当种群多样性指标低于dLow时执行一次大步长的排斥操作，而后继续执行吸引操作。新的dir函数为：

$$\text{newdir}=\begin{cases}-c & \text{newdiversity}<\text{dLow}\\ 1 & \text{others}\end{cases} \tag{5}$$

式中，c为排斥因子，c越大表示个体极点和全局极点对粒子的斥力越强。MARPSO算法中粒子速度和位置的更新公式为：

$$v_{ij}(k+1)=\omega v_{ij}(k)+\text{newdir}\times$$
$$(c_1r_1(p_{ij}-x_{ij}(k))+c_2r_2(p_g-x_{ij}(k))) \tag{6}$$

$$x_{ij}(k+1)=x_{ij}(k)+v_{ij}(k+1) \tag{7}$$

2 基于HC-MARPSO算法的测试数据生成方法

2.1 HC-MARPSO算法的基本思想

结合爬山算法和MARPSO算法的优点，本文提出了HC-MARPSO算法。该算法在MARPSO算法过程中加入爬山算法的分支，当MARPSO算法中个体极值陷入局部最优解的数目大于一定比例时，将这些个体极值从粒子群中提取出来作为爬山算法的初始解集合。然后根据调用爬山算法来优化这些初始解，最后将最优解输出。HC-MARPSO算法流程如图1所示。

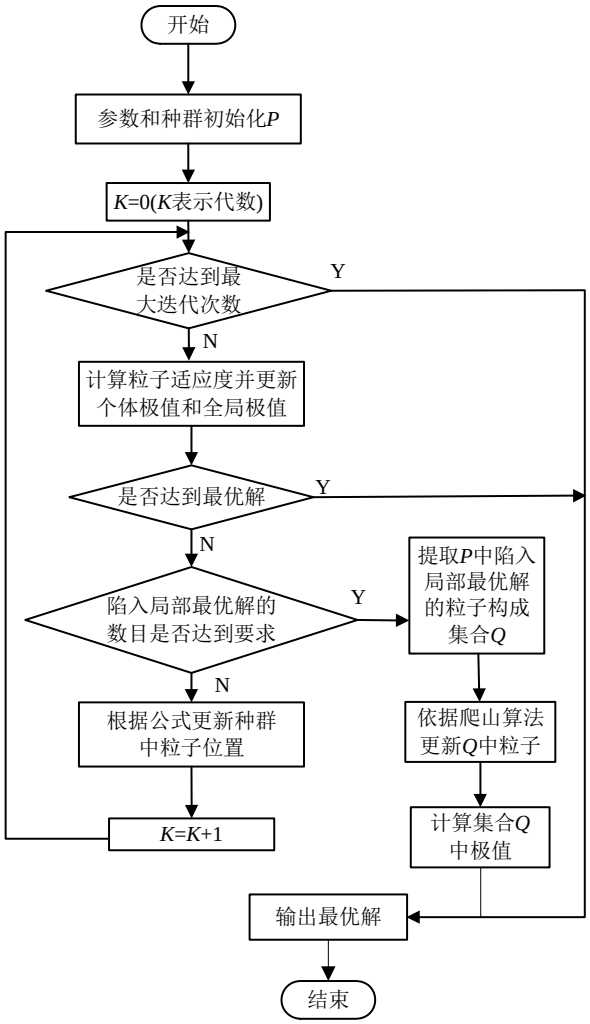

图1　HC-MARPSO算法流程图

算法具体步骤如下：

1) 分析优化问题，构造适应度函数fitness；

2) 参数初始化，确定种群规模m_P、种群维数n、最大迭代次数K、适应度阈值ε和$\mu(\varepsilon<\mu)$、惯性权重ω、加速因子c_1和c_2、速度阈值$v_{\max}$、爬山算法初始解比例r等；

3) 对种群进行初始化，当前迭代次数k=0，随机产生种群粒子的位置$\boldsymbol{X}_i$和速度$\boldsymbol{V}_i(i=1,2,\cdots,m)$；

4) 当$k<K$时，转步骤5)，否则转步骤10)；

5) 计算种群中所有粒子的适应度fitness($\boldsymbol{X}_i$)(i=1,2,$\cdots$,m)，并根据适应度更新各个粒子的个体极值$\boldsymbol{P}_i(i=1,2,\cdots,m)$和全局极值$\boldsymbol{P}_g$；

6) 当fitness($\boldsymbol{P}_g$)>ε时，转步骤7)，否则转步骤10)；

7) 计算fitness($\boldsymbol{P}_i$)(i=1,2,$\cdots$,m)中满足fitness($\boldsymbol{P}_i$)<μ的个数m_h，并记录m_h个个体极值位置为H_i(i=

1,2,⋯ ,m_h)，当m_h<r*m_P时，转到步骤8)，否则转步骤10)；

8) 根据式(4)～式(7)，更新种群粒子位置X_i和速度$\boldsymbol{V}_i$(i=1,2,⋯ ,m)，k=k+1，转步骤4)；

9) 以H_i(i=1,2,⋯ ,m_h)为输入，根据爬山算法更新H_i，令H_g为H_i(i=1,2,⋯ ,m_h)中的极值；

10) 输出最优解$\boldsymbol{P}_g$或H_i，算法结束。

2.2 基于HC-MARPSO算法的测试数据生成模型

软件测试数据自动生成方法有功能测试数据自动生成方法和结构测试数据自动生成方法。将HC-MARPSO算法应用于结构测试数据自动生成过程中。基于HC-MARPSO算法的测试数据生成模型与其他测试数据生成模型相似[10]，主要由测试环境构造模块、HC-MARPSO算法实现模块和测试运行模块3部分组成，如图2所示。

1) 测试环境构造模块通过静态分析的方式对被测程序进行分析，提取出有用的参数及其取值范围，完成适应度函数的构造和对被测程序的插桩；

2) HC-MARPSO算法实现模块是测试数据自动生成的核心部分，首先从测试环境构造模块中获取有用的参数与其取值范围，确定种群的大小，根据测试运行模块提供的适应度信息来决定算法流程；

3) 测试运行模块是测试环境构造模块和HC-MARPSO算法实现模块间的桥梁，主要通过实时调用来运行函数插桩后的被测程序，得到适应度信息，将该信息返回给HC-MARPSO算法实现模块。

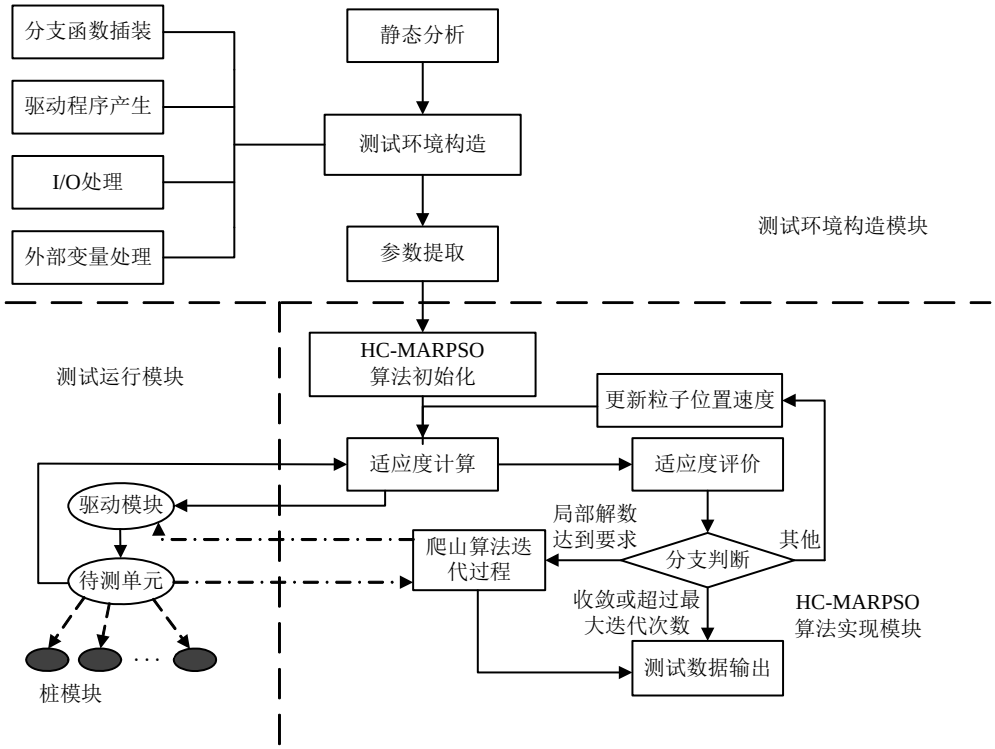

图2 基于HC-MARPSO算法的测试数据生成模型

2.3 适应度函数的构造

适应度函数的构造直接影响算法寻找测试数据的效率，本文采用分支函数叠加方法构造适应度函数。分支函数是一个分支谓词到实际值的映射，可以用来量化分支点的覆盖程度。采用文献[11]的程序插桩理论进行分支函数的插桩[11]。

假设选定路径上有m个分支点和n个参数，则每个在选定路径上各分支点前插入相应的分支函数F_1=f_1(x_1,x_2, ⋯ ,x_n)， F_2=f_2(x_1,x_2, ⋯ ,x_n), ⋯ ,F_m=f_m(x_1, x_2,⋯ ,x_n)，在待测路径结束处插入适应度函数：

$$F=\varphi(F_1)+\varphi(F_2)+\cdots+\varphi(F_m)$$

式中，$\varphi(x)=\begin{cases}0 & x\leqslant 0\\ x & x>0\end{cases}$。

3 实例验证

三角形判别程序包含了清晰又复杂的逻辑，在软件测试数据生成研究文献中广泛使用。本文应用基于HC-MARPSO的测试数据生成算法，对生成三角形指定路径的测试数据所需的迭代次数和时间进行分析，并与遗传算法(GA)、基本粒子群(PSO)算法比较。实验中3种算法在不同种群大小分别运行10次，表1记录了的最好迭代次数、最差迭代次数、平均迭代次数和平均迭代时间。

表1　生成等边三角形测试数据时三种算法的执行性能对比

种群规模	最好迭代次数			最差迭代次数			平均迭代次数			平均迭代时间/ms		
	GA	PSO	HC-MA RPSO	GA	PSO	HC-MA RPSO	GA	PSO	HC-MA RPSO	GA	PSO	HC-MA RPSO
10	30	7	19	952	612	187	541.5	217.9	81.2	207.8	29.5	32.8
20	50	20	8	914	731	249	229.4	163.5	86.1	165.5	37.5	67.2
30	1	5	3	666	273	93	132.3	85	54.6	146.9	26.6	64.1
40	7	10	3	300	406	189	98.4	87.8	49.6	151.7	40.6	76.6
50	9	6	2	290	262	83	89.6	51.3	38.9	171.9	29.7	75

实验中参数选择如下：3条边的输入域为[0,127]，最大迭代次数为1 000。遗传算法采用21位二进制对3个输入变量进行编码，选择方式采用轮盘赌，交叉方式为单点交叉，交叉率为0.85，变异率为0.05；基本粒子群采用实值编码，其中，c_1=c_2=2，随机惯性权重ω=0.5+rand/2，v_{max}=20，ε=0；HC-MARPSO参数与基本粒子群一致，而适应度阈值μ=5，dLow= 0.05，r=0.4。

从表1中的数据可知：

1) 种群规模增大，各种算法生成经过指定路径的测试数据所需的迭代次数减少，但随着种群规模的扩大，算法每次迭代所需的运行时间也在增加；2) HC-MAPRSO算法在平均迭代次数上优于GA和PSO算法；3) HC-MAPRSO算法在平均迭代时间的指标上优于GA却不如PSO算法，但HC-MAPRSO算法保持种群多样性，能收敛于全局最优，而PSO算法易收敛于局部最优。

为验证HC-MAPRSO算法的效率，选取Schaffer函数[12]为：

$$\begin{cases} f(x,y)=0.5+\dfrac{(\sin\sqrt{x^2+y^2})^2-0.5}{(1.0+0.001(x^2+y^2))^2} \\ -100\leqslant x\leqslant 100,-100\leqslant y\leqslant 100 \end{cases}$$

采用Schaffer函数来测试和比较算法性能。该函数是常用的基准测试函数，在搜索区域有无数个局部最小值和一个全局最小值f(0,0)=0，具有强烈的震荡特性，很难收敛。

表2　算法收敛性能对比

迭代次数	全局收敛次数	
	PSO	HC-MARPSO
100	6	23
200	7	28
500	14	42
1 000	20	79

选取种群规模20，PSO算法和HC-MAPRSO算法各执行100次，表2给出了PSO算法和HC-MAPRSO算法分别收敛于全局最优的次数统计。从表2可以看出全局搜索能力上HC-MAPRSO算法效率要高于PSO算法。

4　总　结

改进的吸引排斥粒子群算法通过高效的排斥操作来保障种群的多样性，避免了算法早熟现象，同时新种群多样性指标函数的采用也克服了吸引排斥粒子群算法存在不收敛的问题，但其存在收敛缓慢的不足。相反爬山算法具有很快的局部收敛速度和较高的搜索精度，但缺乏跳出局部最优的能力。本文结合两者的优点，提出了HC-MAPRSO算法，基准函数Schaffer函数的测试结果表明该算法的全局收敛能力要强于粒子群算法。将其应用于软件测试数据生成领域，提出基于该算法的测试数据生成模型。针对三角形判别程序的实验结果表明，该算法的测试数据生成效率高于遗传算法和粒子群算法。

参　考　文　献

[1] 单锦辉. 面向路径的测试数据自动生成方法研究[D]. 长沙: 国防科技大学, 2002: 15-20.
SHAN Jin-hui. On path-wise automated test data generation [D]. Changsha: National University of Defense Technology, 2002: 15-20.

[2] CHEN Yong, ZHONG Yong, SHI Ting-ting, et al. Comparison of two fitness functions for GA-based path-oriened test data generation[C]//The 2009 5th International Conference on Natural Computation. [S.l.]: [s.n.], 2009, 4: 177-181.

[3] CAO Yao, HU Chun-hua, LI Lu-ming. An approach to genertate software test data for a specific path automatically with genetic algorithm[C]//The 2009 8th International Conference on Reliability, Maintainability and Safety. [S.l.]: [s.n.], 2009: 888-892.

[4] XU Xiao-feng, CHEN Yan, Li Xiao-chao, et al. A path-oriented test data generation approach for automatic software testing[C]//The 2008 2nd International Conference on Anti-counterfeiting, Security and Identification. [S.l.]: [s.n.], 2008: 63-66.

[5] SRIVASTAVA P R, RAMACHANDRAN V, KUMAR M, et al. Generation of test data using meta heuristic approach[C] //The 2008 IEEE Region 10 Conference. [S.l.]: IEEE, 2008: 1-6.

[6] LI Ai-guo, ZHANG Yan-li. Automatic generating all-path test data of a program based on PSO[C]//The 2009 World

Congress on Software Engineering. [S.l.]: [s.n.], 2009, 4: 189-193.

[7] RIGET J, VESTERSTROM J S. A diversity-guided particle swarm optimization-the ARPSO[R]. Denmark: EVAlife, Dept of Computer Science, University of Aarhus, 2002.

[8] 毛颖, 罗蕾. 基于遗传算法和爬山算法的测试用例生成研究[J]. 计算机技术与发展, 2006, 16(10): 250-252
MAO Ying, LUO Lei. Research on automated test case generation based on genetic algorithm combined with mountain climbing algorithm[J]. Computer Technology and Development, 2006, 16(10): 250-252

[9] SHI Yu-hui, EBERHART R. A modified particle swarm optimizer[C]//Evolutionary Computation Proceedings, the 1998 IEEE International Conference on. [S.l.]: IEEE, 1998: 69-73.

[10] 贾冀婷. 基于粒子群算法的测试用例自动生成方法研究[J]. 计算机技术与发展, 2010, 20(9): 24-27.
JIA Ji-ting. Research of automatic testcase generation functions based on particle swarm optimization algorithm [J]. Computer Technology and Development, 2010, 20(9): 24-27.

[11] 尉小环, 高惠敏, 李峰. 微粒群算法在软件测试数据生成中的应用[J]. 太原科技大学学报, 2009, 30(4): 294-296.
WEI Xiao-huan, GAO Hui-min, Li Feng. Application of the particle swarm optimization in software testdata generation[J]. Journal of Taiyuan University of Science and Technology, 2009, 30(4): 294-296.

[12] 纪震, 廖惠连, 吴青华. 粒子群算法及应用[M]. 北京: 科学出版社, 2009.
JI Zhen, LIAO Hui-lian, WU Qin-hua. Particle swarm optimization algorithm and its application[M]. Beijing: Science Press, 2009.

编 辑 漆 蓉