

自适应粒子群优化算法及其在测试数据生成中的应用研究

史娇娇,姜淑娟,韩 寒,王令赛

(中国矿业大学计算机科学与技术学院,江苏徐州 221116)

摘 要: 针对粒子群算法易陷入局部最优解及搜索精度低的问题,本文提出一种约简的自适应粒子群优化算法并应用于测试数据的自动生成.首先对粒子群进化方程约简,提出无速度项的粒子群进化方程;然后,对约简后的方程提出基于惯性权重的自适应调整方案,将惯性权重直接作用于粒子的位置,并根据粒子的适应度及聚集度将种群划分为三部分.实验表明,该方法可以有效地提高自动生成测试数据的效率.

关键词: 粒子群算法;测试数据自动生成;进化方程约简;惯性权重;自适应调整方案;粒子聚集度

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2013) 08-1555-05

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2013.08.016

Adaptive Particle Swarm Optimization Algorithm and Its Application in Test Data Generation

SHI Jiao-jiao, JIANG Shu-juan, HAN Han, WANG Ling-sai

(College of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China)

Abstract: According to the particle swarm algorithm that easily falls into the local optimal solution and the problem of low search accuracy, this paper proposes reduced adaptive particle swarm optimization algorithm for generating test data automatically. First, this paper reduces the evolution equations of particle swarm and presents evolution equations without velocity. Then this paper proposes adaptive adjustment scheme based on inertia weight and inertia weight is directly acted on the particle position. According to the particle fitness and particle aggregation degree, the population will be divided into three parts. The experiments show that our approach can effectively improve the efficiency of generating test data automatically.

Key words: particle swarm algorithm; automatic test data generation; reduce the evolution equation; inertia weight; adaptive adjustment scheme; particle aggregation degree

1 引言

软件测试是保证软件质量的重要手段,自动生成测试数据是软件测试的核心问题之一.虽然粒子群算法(Particle Swarm Optimization, PSO)^[1,2]在近几年才用于自动生成测试数据,但在其生成过程中显示出较大的优势.

目前国内外对 PSO 和测试数据生成方法已有一些研究. Andreas Windisch 等人^[3]将 PSO 应用到软件测试中,利用分支覆盖准则生成测试数据.结果表明 PSO 相对遗传算法在生成测试数据上具有竞争力.陈翔等人^[4]将 PSO 与贪心法策略结合,提出基于 one-test-at-a-time 和类 IPO 两种策略,并分析算法框架中不同设定对组合测试用例集规模的影响.查日军等人^[5]提出组合测试数

据生成的交叉熵与粒子群方法,并对两种算法生成的测试数据集进行约简.吴晓军等人^[6]定义了粒子搜索中心的概念并计算其在随机状态下在全局最优解与局部最优解间的概率密度. Zhu 等人^[7]利用粒子适应度对惯性权重进行改进并应用到测试数据自动生成,结果表明该方法在生成效率上比传统 PSO 高.

在以上方法中虽然有些^[8]对如何避免种群过早收敛作了研究,但在如何定量评价个体收敛程度方面的研究还不多见,而本文提出基于约简的自适应粒子群算法(Reduced Adaptive PSO, RAPSPO),对粒子群进化方程进行约简后才作自适应调整,算法仅通过更新粒子位置完成进化过程,不同于以往 PSO 中的惯性权重直接作用于粒子的速度, RAPSPO 直接对粒子的位置即实际问题的解产生影响,同时结合粒子的适应度及聚集度,将种群

分为三个部分,并对每一部分的惯性权重作相应的调整,从而平衡全局与局部的搜索能力.实验表明该方法在收敛速度和搜索精度上有明显的优势.

2 基本概念

在 PSO 中设 t 代种群由 N 个粒子组成,第 i 个粒子的位置和速度分别为 $X_i = [x_i^t, x_i^t, \dots, x_i^t]$ 和 $V_i = [v_i^t, v_i^t, \dots, v_i^t]$,每个粒子在搜索空间搜索时目前为止自身搜索到的历史最好点为个体最优 $pbest$,整个种群中所有粒子发现的最好点为全局最优 $gbest$,粒子在进化过程中速度和位置的更新公式为:

$$\begin{cases} v_i^{t+1} = w v_i^t + c_1 r_1 (pbest - x_i^t) + c_2 r_2 (gbest - x_i^t) & (1) \\ x_i^{t+1} = x_i^t + v_i^{t+1} & (2) \end{cases}$$

w 为惯性权重, $c_1 > 0$ 、 $c_2 > 0$ 为学习因子, $r_1, r_2 \in (0, 1)$ 为随机变量, $v_i \in [-v_{max}, v_{max}]$, $x_i \in [-x_{max}, x_{max}]$.

定义 1 粒子平均适应度值.在 PSO 中设种群规模为 N ,第 i 个粒子的适应度为 f_i ,则当前粒子的平均适应度值 f_{avg} 为:

$$f_{avg} = \frac{1}{N} \sum_{i=1}^N f_i \quad (3)$$

3 RAPSO

3.1 约简粒子群进化方程

在 PSO 进化方程中有速度和位置两个要素.在测试数据生成中,粒子位置 x_i 代表所求问题的解,算法执行的过程就是使 x_i 无限逼近问题的最优解的过程,所以只要合理的设定 x_i 就可以完成 PSO 的更新.粒子速度代表粒子运行的快慢程度,最大值 v_{max} 依靠经验人为设定,它的大小不能表示有效逼近最优位置,相反如果设置不当,还会使粒子偏离原来正确的进化方向,从而导致粒子发散,造成 PSO 后期收敛缓慢、精度低的问题.根据式 $x = vt$,将式(1)(2)叠加后可消去粒子速度,得到无速度项的进化方程:

$$x_i^{t+1} = w * x_i^t + c_1 * r_1 * (pbest - x_i^t) + c_2 * r_2 * (gbest - x_i^t) \quad (4)$$

式(4)中参数与基本概念中出现的参数一一对应.约简后的 PSO 进化方程由原来的二阶降低到一阶,减少了粒子分析和引导粒子进化的过程.

3.2 基于惯性权重的自适应调整方案

惯性权重 w 对 PSO 的收敛速度和精度几乎起着决定性的作用.本文根据粒子的适应度与聚集度,动态调整 w .

定义 2 粒子聚集度.在 PSO 中设粒子的平均适应度值为 f_{avg} ,将适应度值大于 f_{avg} 的粒子再次计算求它们的平均值 f'_{avg} ,设种群的全局最优值为 f_g ,则粒子

群的聚集度 δ 为:

$$\delta = |f_g - f'_{avg}| \quad (5)$$

根据本文提出的自适应调整方案,利用 f_{avg} 和 f'_{avg} 将种群分为三部分,并对每一部分设置相应的 w ,使粒子根据自身在种群中的相对位置自适应改变.这种设置不仅使粒子保持了多样性,也平衡了全局与局部的搜索能力,使种群具有更好的收敛速度.算法 1 为 RAPSO,以下对其具体分析.

算法 1 RAPSO

input: program; a program to be tested

output: the set of test data that are solution to test corresponding branches

variable declaration: $gbest$. fitness: the global optimum of population; t : the current iterations; T : the maximum iterations

1 begin

2 for each particle

3 Initialize Population;

4 Calculate fitness;

5 if (fitness > $gbest$. fitness)

6 $gbest$. fitness = fitness;

7 end if

8 end for

9 while ($t < T$ or fitness = = 100)

10 Calculate $f_{avg}, f'_{avg}, \delta$

11 if ($f_i > f'_{avg}$) $w - i = w_{max} - (w_{max} - w_{min}) |f_i - f'_{avg}| / |f_g - f'_{avg}|$;

12 end if

13 else if ($f_{avg} < f_i < f'_{avg}$) $w - i = w_{min} + (w_{max} - w_{min}) (1 + \cos(t/T) \pi) / 2$;

14 end if

15 else if ($f_i < f_{avg}$) $w - i = c + 1 / (1 + \exp(-\delta))$;

16 end if

17 for each particle

18 Reduce updating formula;

19 $x_i^{t+1} = w * x_i^t + c_1 * r_1 * (p_{id} - x_i^t) + c_2 * r_{12} * (p_{gd} - x_i^t)$;

20 Update particle position;

21 end for

22 end while

23 end

(1)第 11-12 行,个体适应度值 f_i 大于 f'_{avg} .这部分粒子的性能比较好即比较接近问题的解,所以这部分粒子的位置改变不应过大,粒子的位置越靠近最优解,惯性权重值 w 应越小. $w - i$ 调整如下:

$$w - i = w_{max} - \frac{(w_{max} - w_{min}) |f_i - f'_{avg}|}{|f_g - f'_{avg}|} \quad (6)$$

(2)第 13-14 行,个体适应度值 f_i 介于 f_{avg} 和 f'_{avg} 之间.这部分粒子的性能在种群中处于中等,它们的全局和局部搜索能力都相对较好.这部分粒子的位置会随着 PSO 的执行及迭代次数的增加,在搜索解的过程中非线性的逐渐减小,按照图 1 中粒子位置的变化趋势,

确保粒子的相对位置在前期保持较大,使 PSO 具有很好的搜索能力,而在后期又保持较小,使 PSO 具有精细的搜索精度.利用余弦定律可以很好的表示这部分粒子的位置的变化情况, w_i 调整如下, t/T 分别表示当前/最大进化代数.

$$w_i = w_{\min} + (w_{\max} - w_{\min}) \frac{1 + \cos \frac{t}{T} \pi}{2} \quad (7)$$

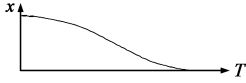


图1 性能居中的粒子的位置变化趋势图

(3)第 15-16 行,个体适应度值 f_i 小于 f_{avg} ,这部分粒子的性能在种群中较差.当种群过早收敛时,增大 w 会破坏当前的稳定性;当个体发散时,减小 w 会使个体趋于收敛.按照指数函数变化曲线,使 w_i 在进化过程中随 δ 的增大而减小, w_i 调整如下:

$$w_i = c - \frac{1}{1 + \exp(-\delta)} \quad (8)$$

c 为常数,实验证明 $c = 1.5$ 时粒子的收敛能力和搜索能力较好.从式(8)可知在进化过程中 w_i 根据 δ 的取值而动态的自适应调整.当 δ 值较大说明粒子比较分散,减小 w 可以增加种群的收敛能力;当 δ 较小说明粒子比较聚集,增大 w 可使粒子的搜索能力增强,具备跳出局部极值的能力.

4 将 RAPS0 用于测试数据自动生成

4.1 适应度函数构造

利用 PSO 自动生成测试数据的关键在于适应度函数的构造.函数 $f(x_i)$ 为第 i 个分支对应的分支函数, w_i 为对应的权值,适应度函数为 $f(x)$.

$$f(x) = \sum_{i=1}^n w_i \times f(x_i) \quad (9)$$

$E1$ 和 $E2$ 是算术表达式.

(1)当 $f(x_i)$ 期望值为真,如果 $f(x_i)$ 的实际值和期望值一致,则 $f(x_i) = 0$;如果 $f(x_i)$ 的实际值为假,各分支对应的 $f(x_i)$ 为:

$$f(x_i) = \begin{cases} abs(E1 - E2) + \alpha, & E1 > E2 \text{ or } E1 < E2 \\ abs(E1 - E2), & E1 \geq E2 \text{ or } E1 \leq E2 \\ abs(E1 - E2), & E1 = E2 \\ \alpha, & E1 \neq E2 \end{cases} \quad (10)$$

(2)当 $f(x_i)$ 期望值为假,如果 $f(x_i)$ 的实际值和期望值一致都为假,那么 $f(x_i) = 0$;如果 $f(x_i)$ 的实际值为真,那么各个分支对应的 $f(x_i)$ 为:

$$f(x_i) = \begin{cases} abs(E1 - E2), & E1 > E2 \text{ or } E1 < E2 \\ abs(E1 - E2) + \alpha, & E1 \geq E2 \text{ or } E1 \leq E2 \\ abs(E1 - E2), & E1 = E2 \\ \alpha, & E1 \neq E2 \end{cases} \quad (11)$$

α 为正数区分边界 $E1 > \text{或} \geq E2 \setminus E1 < \text{或} \leq E2$. 设 f_m 为选定路径上各分支函数值.适应度函数为 $f(x)^{[9]}$.将式(12)中 $f(x_i)$ 的倒数相加得适应度函数 $fit(x)$.

$$f(x) = \psi(f_1) + \psi(f_2) + \dots + \psi(f_m) \quad (12)$$

$$\psi(x) = \begin{cases} 0, & x < 0 \text{ or } x = 0 \\ x, & x > 0 \end{cases} \quad (13)$$

$$fit(x) = \frac{1}{1 + f(x_i)} \quad (14)$$

4.2 基于 RAPS0 的测试数据自动生成模型

基于 RAPS0 的测试数据自动生成模型(图 2)分三个部分:测试环境构造、RAPS0 和测试运行.

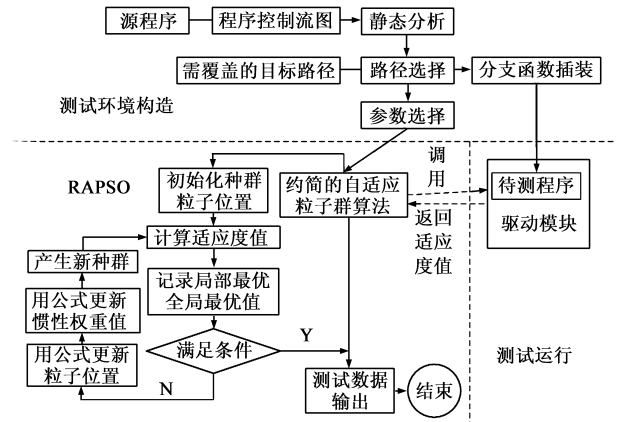


图2 基于约简的自适应粒子群算法的测试数据自动生成模型

测试环境构造是整个模型的基础部分.该部分对被测程序静态分析,选择目标路径,及相关参数选择和分支函数插装. RAPS0 是核心部分,根据测试环境构造提供的参数及其搜索域,初始化种群,并根据约简的进化公式及惯性权重公式更新粒子位置,引导粒子群向目标解进化.测试运行部分在调用待测程序时,将粒子位置映射到实际问题参数的搜索域中,并作为参数传递给待测程序.

5 实验

在 Eclipse 环境下将 RAPS0 应用到自动生成测试数据.参数设置如下:最大进化代数 = 1000,学习因子 $c_1 = 2, c_2 = 2$,惯性权重 $w_{\max} = 0.9, w_{\min} = 0.4$.不同的程序根据实验需要重新对各参数进行设置.

5.1 三角形分类程序

三角形分类程序^[10]共有 12 个分支,最大嵌套层数为 5,最难覆盖的分支是生成等边三角形的分支,受两个 AND 决策条件和两个等式决策条件构成的 4 层选择嵌套制约.

为测试 RAPS0 的优劣,对 RAPS0 和文献[7]的 AP-PSO 进行实验,记录每次找到目标路径时的进化代数,针对不同的数据范围进行 100 次搜索,因每次运行的进化

时间只有几 ms,所以记录了 100 次总的进化时间.

从表 1、2、3 可以看出:(1)随着输入数据范围的增大,两种方法的时间差距越来越大,说明 RAPS0 在大范围输入下寻优能力更强;(2)随着种群数量的增加,同输入范围内两种方法找到最优解的平均进化代数差距越来越大,RAPS0 的迭代次数明显小于 APSO 方法.

表 1 种群规模为 50 时,两种算法运行结果

种群规模	数据范围	平均进化代数		总进化时间(s)	
		RAPS0	APSO	RAPS0	APSO
50	[0,100]3	14.3	42.7	0.101	0.324
	[0,200]3	21.02	53.5	0.173	0.502
	[0,500]3	29.52	125.8	0.234	2.875

表 2 种群规模为 100 时,两种算法运行结果

种群规模	数据范围	平均进化代数		总进化时间(s)	
		RAPS0	APSO	RAPS0	APSO
100	[0,100]3	11.5	29.54	0.126	0.298
	[0,200]3	16.06	42.58	0.176	0.475
	[0,500]3	20.02	99.3	0.229	2.016

表 3 种群规模为 150 时,两种算法运行结果

种群规模	数据范围	平均进化代数		总进化时间(s)	
		RAPS0	APSO	RAPS0	APSO
150	[0,100]3	8.96	20.42	0.125	0.252
	[0,200]3	12.7	32.57	0.198	0.486
	[0,500]3	17.62	88.69	0.231	1.975

图 3 表示两种算法的平均进化代数与分支覆盖率的关系.RAPS0[1-100]表示 RAPS0 在 1 到 100 的输入范围内搜索解.从图中可知 RAPS0 在迭代初期就达到了 50% 覆盖率,随着平均进化代数的增加,很快达到 100% 的覆盖率,针对不同的输入范围,在满足分支覆盖率为 100% 时,RAPS0 平均进化代数明显小于 APSO.

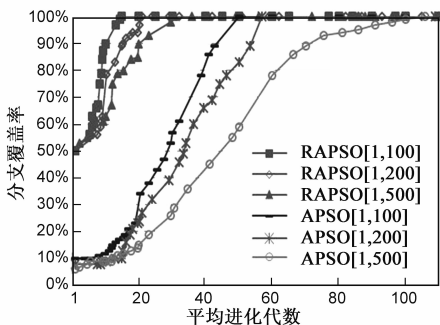


图 3 不同数据范围,平均进化代数与分支覆盖率的关系

由于在文献[7]中作者证明了 APSO 方法比传统 PSO 的性能更优,所以本文只与 APSO 方法进行比较.

5.2 判断线段与矩形的关系程序

判断线段与矩形关系的程序^[10]共有 8 个变量,分别表示矩形/线段的坐标.程序有 36 个分支有四种输出结果:直线完全在矩形内、直线完全在矩形外、直线部

分在矩形内、错误:不能构成直线或矩形.若程序输出正确,判断直线的三种情况:水平、垂直和倾斜,并判断线段与矩形的相交情况.

图 5 为两种方法找到最优解时的平均进化代数,[1,100]表示输入范围,从图 4、图 5 看出,在迭代初期 RAPS0 就达到 40% 的覆盖率,在一定的条件下 APSO 很难达到 100%,特别是对有较深嵌套层次和复杂的决策条件的分支,而 RAPS0 在 [1-100] 只需要平均 40 多代就可以达到 100%,在 [1-200] 需要平均 60 多代,在 [1-500] 只需 89.7 代就可以达到 100%.

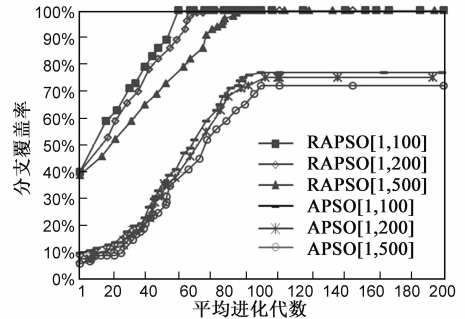


图 4 不同数据范围,平均进化代数与分支覆盖率的关系

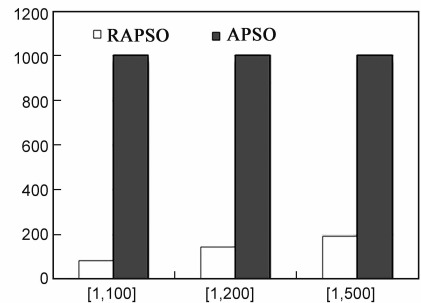


图 5 不同数据范围,两种方法平均进化代数比较

5.3 求两日期相隔天数

求两日期相隔天数程序^[10]共有 6 个变量,程序的输出结果是两日期间隔的天数.程序共有 86 个分支,约束条件包含大量的分支、等式、AND 条件及余数操作,嵌套层次较深.程序首先判断日期是否有效;有效则对给定的两日期进行排序,并判定日期是否为闰年,以此计算两日期相隔的天数.

从图 6、图 7 可看出,在 [1-100] 范围 APSO 方法虽然不能达到 100% 的覆盖率,仍能达到 79.3% 覆盖部分路径,随着输入范围的增加,APSO 甚至无法满足第 1 个节点的条件,只能达到 8.92% 的覆盖率.而 RAPS0 在 [1-100] 需要平均 83.5 代达到 100%;在 [1-200] 需要平均 127.8 代;在 [1-500] 需要 192.3 代可以达到 100% 的覆盖率.由此可以看出,RAPS0 具有良好的性能.

6 结论

寻找有效的测试数据自动生成方法是软件测试中

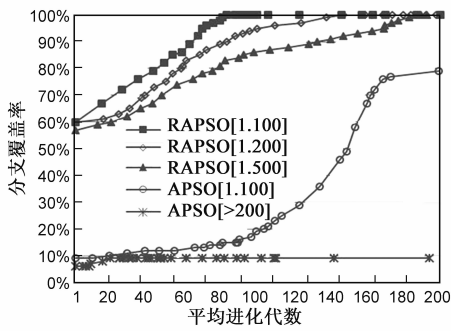


图6 不同数据范围,平均进化代数与分支覆盖率的关系

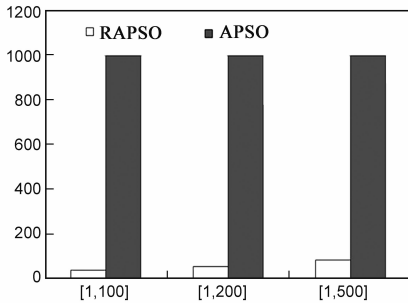


图7 不同数据范围,两种方法平均进化代数比较

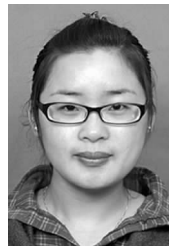
的难题.本文提出的 RAPSO 方法解决了粒子群算法易陷入局部最优解及搜索精度低的问题.通过对粒子群进化方程的约简,避免了依靠经验设定速度带来的耗时及难以把握的缺点;提出的基于惯性权重的自适应调整方案,平衡了算法全局与局部的搜索能力.同时对 3 个复杂程序进行实验对比,证明 RAPSO 具有更好的寻优能力.

参考文献

- [1] 陶新民,刘福荣.一种多尺度协同变异的粒子群优化算法[J].软件学报,2012,23(7):1805-1814.
Tao Xin-min, Liu Fu-rong. Multi-scale cooperative mutation PSO [J]. Journal of Software, 2012, 23(7), 1805-1814. (in Chinese)
- [2] 罗金炎.连续型粒子群优化算法的均方收敛性分析[J].电子学报,2012,40(7):1364-1367.
Luo Jin-yan. The analysis of continuous particle swarm optimization algorithm's mean square convergence [J]. Acta Electronica Sinica, 2012, 40(7): 1364-1367. (in Chinese)
- [3] Windisch A, Wappler S, Wegener J. Apply particle swarm optimization to software testing [A]. Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation [C]. New York: ACM, 2007. 1121-1128.
- [4] 陈翔,顾庆,等.一种基于粒子群优化的成对组合测试算法框架[J].软件学报,2011,22(12):2879-2893.
Chen Xiang, Gu Qing, et al. Framework of particle swarm optimization based pairwise testing [J]. Journal of Software, 2011, 22(12): 2879-2893. (in Chinese)

- [5] 查日军,张德平,等.组合测试数据生成的交叉熵与粒子群算法比较[J].计算机学报,2010,33(10):1896-1908.
Cha Ri-jun, Zhang De-ping, et al. Test data generation algorithms of combinatorial testing and comparison based on cross-entropy and particle swarm optimization method [J]. Journal of Computers, 2010, 33(10): 1896-1908. (in Chinese)
- [6] 吴晓军,李峰,等.均匀搜索粒子群算法的收敛性分析[J].电子学报,2012,40(6):1115-1120.
Wu Xiao-jun, Li Feng, et al. The convergence analysis of the uniform search particle swarm optimization [J]. Acta Electronica Sinica, 2012, 40(6): 1115-1120. (in Chinese)
- [7] Zhu X M, Yang X F. Software Test data generation automatically based on improved adaptive particle swarm optimizer [A]. Proceedings of the International Conference on Computational and Information Sciences [C]. Washington: IEEE, 2010. 1300-1303.
- [8] 李爱国.多粒子群协同优化算法[J].复旦学报,2004,43(5):923-925.
Li Ai-guo. Particle swarms cooperative optimizer [J]. Journal of Fudan University, 2004, 43(5): 923-925. (in Chinese)
- [9] Korel B. Automated software test data generation [J]. IEEE Trans on Software Engineering, 1990, 16(8): 870-879.
- [10] Díaz E, Tuya J, Blanco R, Dolado J J. A tabu search algorithm for structural software testing [J]. Computers and Operations, 2008, 35(10): 3052-3072.

作者简介



史娇娇 女,1988 年出生,江苏徐州人.中国矿业大学研究生.研究方向为软件测试、测试数据生成.

E-mail: sjiao888@126.com



姜淑娟(通信作者) 女,1966 年出生,山东莱阳人.中国矿业大学教授、博士生导师、CCF 会员.主要研究领域为编译技术、软件工程等.

E-mail: shijiang@cumt.edu.cn