

改进的粒子群算法在测试数据生成中的应用

邓璐娟, 卢华琦, 孙义坤, 刁海港

(郑州轻工业学院 计算机与通信工程学院, 河南 郑州 450002)

摘要:自动化测试中,测试数据的自动生成技术是提供软件测试效率和效果的瓶颈。粒子群算法(PSO)具有简单、易实现、可调参数少等特点,在测试数据生成方法中得到初步应用。在具体应用过程中,为克服 PSO 易陷入局部极值的缺陷,对算法进行了改进,应用加入移动步长的混合粒子群算法(SwPSO)自动生成测试数据,提高了 PSO 算法摆脱局部极小点的能力。文中对算法的原理和实现做了详细描述,并将其与传统的基于标准粒子群算法(PSO)和遗传算法(GA)来实现软件测试数据自动生成方法进行实验对比。结果表明,改进后的粒子群算法可以更高效地生成测试数据。

关键词:软件测试;测试数据;粒子群算法

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2010)07-0216-03

Improved Adaptive PSO Application on Automatic Test Data Generation

DENG Lu-juan, LU Hua-qi, SUN Yi-kun, DIAO Hai-gang

(Sch. of Computer and Comm. Eng., Zhengzhou Univ. of Light Ind., Zhengzhou 450002, China)

Abstract: Automatic generation technology of test data is an important field of software test, which is an effective method to promote the efficiency and the effect of software test. Particle swarm optimization(PSO) has few parameters need to be tuned, and has been initially applied on test data generation method. In the specific application process, to overcome the deficiencies of easily falling into local extreme on PSO, proposes a hybrid particle swarm optimization(SwPSO), which added moving-step and improve performance of basic PSO algorithm. Describe the principle and realization of the approach in detail. By comparing with both particle swarm optimization and genetic algorithm, this test data generation based on SwPSO is proved to be more efficient.

Key words: software testing; test data; particle swarm optimization algorithms

0 引言

测试数据生成是软件测试过程中的重要环节,研究其自动生成方法对提高软件测试的效率有着重要的价值。1980年以来,许多学者对测试数据的自动生成进行了研究,提出了一些方法,如:随机法、符号执行法、迭代松弛法和启发式算法^[1]。目前研究较多的用于生成测试数据的启发式算法有:禁忌搜索^[2]、模拟退火、遗传算法^[3]、粒子群算法等,这些算法在一定程度上提高了测试数据生成的效率,但还存在缺陷和不足。遗传算法存在局部搜索能力差及其早熟现象,模拟退火算法存在全局搜索能力差、效率不高的问题,而粒子群算法有迭代后期全局搜索能力不足,不易找到要求的最优解的缺陷。文中针对粒子群算法本身存在的不足

加以改进,提出了一种混合粒子群算法(SwPSO),用改进后的粒子群算法生成测试数据,能很快产生局部极值,提高生成测试数据的收敛速度。

1 基本粒子群算法及其改进

1.1 PSO 算法基本原理

粒子群优化算法(PSO)是一种进化计算技术(evolutionary computation),最早由 Eberhart 和 Kennedy (1995)提出了粒子群算法^[4],其基本思想源于对鸟群捕食的行为研究,但后来发现 PSO 与神经网络、遗传算法一样,也是一种很好的优化工具。系统初始化为一群随机粒子(随机解),通过迭代搜寻最优值。每一次迭代中,粒子通过跟踪两个“极值”更新自己。第一个就是粒子本身所找到的最优解,叫做个体极值,记为 P_i ;另一个极值就是整个种群目前找到的最优解,叫全局极值,记为 P_g 。

设搜索空间为 D 维,总粒子数为 n ,该粒子群可用

收稿日期:2009-12-02;修回日期:2010-03-06

基金项目:河南省新世纪优秀人才支持项目(2005HANCET-03)

作者简介:邓璐娟(1964-),女,湖南浏阳人,博士,教授,硕士生导师,研究方向为软件测试、控制理论与控制工程。

如下参数来表示: $x_i(x_{i,1}, x_{i,2}, \dots, x_{i,K})$: 第 i 个粒子在 D 维空间中的当前位置。 $V_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$: 第 i 个粒子在 D 维空间中的当前速度。 $P_i = (p_{i,1}, p_{i,2}, \dots, p_{i,D})$: 第 i 个粒子在 D 维空间中的历史最优位置。 $P_g = (p_{g,1}, p_{g,2}, \dots, p_{g,D})$: 整个粒子群体经历过的最好位置。迭代公式:

$$v_{i,d} = w * v_{i,d} + c_1 * r_1 * (p_{i,d} - x_{i,d}) + c_2 * r_2 * (p_{g,d} - x_{i,d}) \quad (1)$$

$$x_{i,d} = x_{i,d} + v_{i,d} \quad (2)$$

其中: $i = 1, 2, \dots, m$; $d = 1, 2, \dots, D$; c_1 和 c_2 是非负常数, 称为学习因子。 r_1 和 r_2 是介于 $[0, 1]$ 之间的随机数; $v_{i,d} \in [-v_{\max}, v_{\max}]$, $x_{i,d} \in [-x_{\max}, x_{\max}]$, v_{\max} 、 x_{\max} 是依据不同的目标函数和不同的搜索空间而不同的常数, w 称为惯性权重, 它是一个非负数, 目前较常用的是 Shi 建议的线性递减权值策略, w 较大时适于对解空间进行大范围的探查, w 较小时适于进行小范围搜索, 即

$$w = \frac{w_{\text{ini}} - w_{\text{end}}}{\text{Maxgen}} (\text{Maxgen} - t) + w_{\text{end}} \quad (3)$$

其中, w_{ini} 、 w_{end} 表示 w 的初始值和迭代终止值。 Maxgen 为最大代数, t 为当前的迭代次数。

1.2 PSO 算法的改进

由于标准 PSO 算法存在后期搜索能力不足, 易发生早熟收敛等现象, 文中对此加以改进, 其主旨思想^[5]为: 首先定义整个粒子群的中心点, 而后让每个粒子的当前位置与中心点相比较, 假如中心点位置较好, 粒子就向中心点移动一步, 步长随机; 否则, 在其邻域位置随机移动一步。根据这种移动策略, 位置较差的粒子可以很快聚集到一个相对较好的区域内, 自身位置较好的一些粒子则在其邻域内进行下一步的局部搜索。

改进的 PSO 算法流程如下:

第一步 初始化个体规模 m , 迭代次数 Maxgen_1 、 Maxgen_2 , 移动步长 Step , 学习因子 c_1 、 c_2 , 惯性权重 w_{ini} 、 w_{end} 和最大速度 v_{\max} ;

第二步 把当前位置设为个体的历史最优位置, 所有个体中最优个体的位置设为全局最优位置, 令 $k = 0$, 当 $k < \text{Maxgen}_1$ 时执行以下循环:

(1) 找出所有个体的中心点位置 X_c , 计算其目标值 $Y_c = F(X_c)$:

$$X_c = \sum_{i=1}^m X_i / m \quad (4)$$

(2) 对每个个体 X_i 执行以下操作: 如果 $Y_c < Y_i$, 则个体向 X_c 处移动, 即

$$X'_i = X_i + c_1 * \text{rand}(X_c - X_i) \quad (5)$$

否则, 在其邻域内随机移动一步, 即

$$X'_i = X_i + c_1 * (\text{rand}() - 0.5) * \text{Step} \quad (6)$$

式中 rand 表示随机数, $\text{rand}()$ 表示随机向量。

(3) $K = k + 1$;

第三步 把个体的当前位置记为粒子的初始位置, 个体的最优位置记为粒子历史最优值, 全局最优值保持不变。

第四步 确定粒子的初始速度: $v = (v_1, v_2, \dots, v_m)$, $v_i = (x_{i \max} - x_{i \min}) * \text{rand}$, 其中 $x_{i \max}$ 表示此时群体中所有个体在第 i 维上的最大值, $x_{i \min}$ 为最小值。

第五步 令 $k = 0$, 当 $k < \text{Maxgen}_2$ 时执行以下循环:

(1) 按照公式(3) 计算 w 值, 按公式(4) 对所有粒子的速度和位置进行更新;

(2) 更新粒子个体极值及全局极值, $k = k + 1$;

(3) 判断是否满足收敛条件, 如果满足, 输出结果; 否则转到第三步。

2 用改进的粒子群算法自动生成测试数据

2.1 构造适应值函数

指定 $p = n_1, n_2, \dots, n_m$ 为被测试程序的一条路径, 现在要求解的问题目标是找到一个程序输入 $\bar{x} \in D$, 使得该程序以 x 为输入运行, 经过的路径 p 。该问题目标^[6]可化为一系列的子目标, 用函数极小化搜索技术^[5]来解决每个子目标。假定分支谓词是如下关系表达式: $A \text{ op } B$, 其中 A 和 B 是算术表达式, 关系运算符 $\text{op} \in \{<, \leq, >, \geq, =, !=\}$ 。适应值函数的构造方法^[7]是检测条件语句的真假值关系, 如果能满足给定的真假值, 则适应值函数值为 0; 否则进行如表 1 所示的运算。

表 1 适应值函数计算方法

表达式	适应值函数	
	0; (表达式为 TRUE)	K; (表达式为 FALSE)
$A = B$	0	$\text{abs}(A - B) + k$
$A \neq B$	0	k
$A < B$	0	$(A - B) + k$
$A \leq B$	0	$(A - B) + k$
$A > B$	0	$(B - A) + k$
$A \geq B$	0	$(B - A) + k$
$A \vee B$	\	$\min(\varphi(a), \varphi(b))$
$A \wedge B$	\	$\varphi(a) + \varphi(b)$
Boolean	0	k

文中采用“分支函数叠加法”^[8]来构造适应值函数。分支函数^[7,9]是一个分支谓词到实际值的映射, 若一条路径包含多个判断语句, 可以把每个适应值函数进行叠加, 叠加后的结果为此个体的适应值函数。设待测路径上有 m 个分支点, n 个参数, 则 m 个分支函数分别为: $\phi_1 = f_1(x_1, x_2, \dots, x_n)$, $\phi_2 = f_1(x_1, x_2, \dots,$

$x_n), \dots, \phi_m = f_1(x_1, x_2, \dots, x_n)$ 。

该待测路径的分支函数为:

$$F = \Phi(\phi_1) + \Phi(\phi_2) + \dots + \Phi(\phi_m) \quad (7)$$

$$\text{其中, } \Phi(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases}$$

2.2 采用改进 PSO 算法生成测试数据的步骤

(1) 初始化: 随机在各变量的输入范围内产生一组随机数据。然后结合代码插装^[10]技术, 在待测路径所经过的每个分支点前插入一个分支函数, 由(7)式计算适应度值。

(2) 选定种群大小 m 、最大允许迭代次数 $\text{Maxgen}_1, \text{Maxgen}_2$, 适应值阈值 ε, c_1, c_2 和 $w_{\text{ini}}, w_{\text{end}}$; 初始化 X' 和 V 分别为 $(0, 100)$ 和 $(0, 1)$ 间的随机数。

(3) 迭代次数 $t = 0; F_g = 0; F_g(i) = (0, 0, \dots, 0)$

(4) while($F_g \leq \varepsilon \& t < \text{Maxgen}_2$)

(5) for($i = 0; i < m; i++$)

{ 评定粒子的适应度:

if ($F_i > F_g(i)$) { $F_g(i) = F_i; p_i = x_i; \}$

if ($F_i > F_g$) { $F_g = F_i; p_g = x_i; \}$

}

(6) for ($i = 0; i < m; i++$) 按式(1) 计算 v_i ; 按式(2) 计算 x_i ;

(7) $t = t + 1$

在上述流程中, $X = (x_1, x_2, \dots, x_m), x_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$ 是第 i 个粒子的位置; $V = (v_1, v_2, \dots, v_m)$, 其中, $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})^T$ 是第 i 个粒子的速度; F_i 是第 i 个粒子迄今搜索到的最优适应值; F_g 是粒子群迄今搜索到得最优适应值, 对应的最优粒子位置是 P_g 。

3 实验结果及分析

选择软件测试文献中使用最广的三角形问题^[11,12]、二元一次方程 $ax^2 + bx + c = 0$ 求解问题。为验证改进的粒子群优化算法(SwPSO) 在生成测试数据时的可行性及有效性, 文中将其与传统基于粒子群算法(PSO)、遗传算法(GA) 实现测试数据生成的方法进行了以下实验比较:

(1) 生成等边三角形的测试数据;

(2) 生成直角三角形的测试数据;

(3) 二元一次方程存在两个相等实根的测试数据。

实验数据设置如下:

改进 PSO 算法参数设置: $m = 100, w$ 由 0.9 线性递减至 0.2, $c_1 = c_2 = 1.2, \text{Maxgen}_1 = 400, \text{Maxgen}_2 =$

1000, $\text{Setp} = 0.015$;

简单 PSO 算法参数设置: $m = 100, w$ 由 0.9 线性递减至 0.2, $c_1 = c_2 = 1.2, \text{Maxgen} = 1400$;

遗传算法参数设置: $m = 100, \text{Maxgen} = 1600$, 选择概率为 0.9, 交叉概率为 0.7, 变异概率为 0.03。

三种算法各生成 50 次数据, 记录首次出现最优解的迭代次数及所耗费的时间, 取平均值(见表 2)。

表 2 三种算法的性能比较

	GA		PSO		SwPSO	
	G	T(s)	G	T(s)	G	T(s)
等边三角形	162	3.25	13	1.06	4	0.47
直角三角形	256	5.68	59	1.13	26	0.63
相等实根	476	4.19	107	1.38	49	0.59

由表 2 可以看出, 对于等边三角形来说, 应用 GA 一般在 160 代左右产生适应度最优个体。采取标准粒子群算法在 15 代左右就产生了最优解。而如果使用改进的粒子群算法后, 最优个体的生成代数还可大大提前。对于其他两个程序有类似的结果。就运行时间来说, 改进的粒子群也比其他两种算法需要的时间少。从进化的代数和运行时间上都可以得到令人满意的结果, 证明提出的改进粒子群算法是合理有效的。

4 结束语

测试数据的自动生成是软件测试中一个关键问题, 改进其生成方法, 可以提高软件测试自动化程度和测试的效率。文中在基于粒子群算法生成测试数据的基础上, 提出了一种新的混合粒子群算法生成测试数据的方案, 对初始化粒子群进行了优化, 并融入了移动步长, 加速产生局部极值, 克服了标准粒子群算法容易陷入局部收敛解而出现早熟等缺点。并与基本的粒子群算法和传统的遗传算法测试数据生成方法进行了对比, 证实使用该方法进行测试数据自动生成是行之有效和高效的。进一步的工作可以结合其它智能优化算法(如遗传算法、蚁群算法等)来继续改进算法的寻优性能和减小陷入局部最优的概率。

参考文献:

- [1] Gotlieb A, Denmat T, Botella B. Goal - Oriented Test Data Generation for Programs with Pointer Variables [C] // Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC'05). Washington: IEEE Computer Society, 2005: 449 - 454.
- [2] Eugenia D, Javier T, Raquel B. Automated software testing using a metaheuristic technique based on Tabu Search [C] // In 18th IEEE International Conference on Automated Software Engineering. Montreal, Canada: [s. n.], 2003: 310 - 313.

的向量以计算查询与文档的相似度^[12]。

这种权重计算方式中 w_{ij} 的大小与 t_i 在文档 d_j 中出现的次数成正比,而与 t_i 在整个文本集中出现的次数成反比。计算相似度公式见公式(3),它是通过考察特征向量余弦夹角实现的。

$$\text{Sim}(Q, d_j) = \frac{\sum_{k=1}^M W_{ki} \times W_{kj}}{\sqrt{(\sum_{k=1}^M W_{ki}^2)(\sum_{k=1}^M W_{kj}^2)}} \quad (3)$$

文物信息获取系统的信息分类方法是以文物信息的时代特征为依据,通过设置分类关键词得到不同的分类权重,最终通过权重进行分类。事先制定好每个文物分类包含哪些关键词,如果文物信息标题或介绍中包括哪个分类的某些关键词,则认为该条文物信息属于该分类。每一个文物信息可以按照多个方式分类,事先定义好使用什么分类方法以及应包含哪些关键词,然后用文物信息的标题和内容与每个分类的关键词比较,与哪个分类的符合度高,就属于哪个分类。举例来说,如按时代分,可以得到文物分类如表 2。

表 2 文物分类

文物分类名称	包括的关键词
夏代文物	夏初 夏朝 夏代 夏末
商代文物	商朝 商初 商末 商代
秦代文物	秦初 秦始皇 秦末 秦朝 秦代
.....
宋代文物	宋朝 宋代 北宋 南宋 宋末 宋武帝 西夏
.....
清代文物	清朝 清初 清末 清代 康熙 乾隆
其他

4 结束语

通过对数字博物馆文物信息获取系统所采用的各

种主要技术进行分析,阐明了这些技术的实现方法及采用这些技术所带来的好处。信息获取方法的改进主要体现在不同搜索技术的组合、关键词预处理、多网站信息同时采集等方面,提高了信息获取的准确性;信息分析方法的改进是在充分尊重原有信息排列顺序的基础上,加入了新的得分元素,得到的结果是比较满意的;信息分类方法选择了文物信息的时代特征为基本依据,通过进一步设置分类关键词得到不同的分类权重,最终通过权重实现信息分类。

参考文献:

- [1] 鲍泓. 基于 Web Services 的虚拟文物博物馆架构[J]. 系统仿真学报, 2005(6): 1412-1417.
- [2] 张佳强, 周锦程, 王士同. 基于领域模型的信息系统分析与应用[J]. 微计算机信息, 2009(3-3): 195-196.
- [3] 王郁新. Web 服务在数字博物馆中的应用[J]. 计算机科学, 2007(10): 58-60.
- [4] 黎文. 数字博物馆关键技术[J]. 北京科协, 2005(5): 40-43.
- [5] 陆宜梅. Web 搜索技术现状分析[J]. 沈阳大学学报, 2006(4): 34-36.
- [6] 张宏斌. 智能化搜索引擎技术的研究进展[J]. 信息与控制, 2003(12): 526-530.
- [7] 姚全珠. 基于数据挖掘的搜索引擎技术[J]. 计算机应用研究, 2006(11): 29-30.
- [8] 龚正伟. 数字博物馆的建设与发展[J]. 北京科协, 2005(5): 17-19.
- [9] 王永平. 基于 Web 的数字博物馆虚拟空间分类索引研究[J]. 计算机科学, 2007(10): 58-60.
- [10] 何淑庆. URL 分级散列在分布式搜索引擎中的应用[J]. 电子技术应用, 2006(7): 25-27.
- [11] 张绚丽. 基于搜索技术的科技期刊网站建设要点研究[J]. 武汉科技大学学报, 2006(10): 76-78.

(上接第 218 页)

- [3] 英伟, 谢军, 奚红宇, 等. 遗传算法在软件测试数据生成中的应用[J]. 北京航空航天大学学报, 1998, 24(4): 434-437.
- [4] Jullier E M. Tunneling between ferromagnetic film[J]. Phys Lett, 1975, 54(3): 225-226.
- [5] 常先英, 李荣钧. 改进粒子群优化算法及其在 CVaR 模型中的应用[J]. 统计与决策, 2009(8): 144-146.
- [6] 王溪波, 马春, 杜晓舟. 面向路径的测试数据自动生成工具设计与实现[J]. 沈阳航空工业学院学报, 2009, 26(6): 54-59.
- [7] 李爱国, 张艳丽. 基于 PSO 的软件结构测试数据自动生成方法[J]. 计算机工程, 2008, 34(6): 93-94.
- [8] 虞凡, 覃征, 贾晓琳. 基于 XYZ/E 规范的软件测试用例自动生成方法[J]. 计算机工程, 2005, 31(19): 76-78.
- [9] 夏芸, 刘锋. 基于免疫遗传算法的软件测试数据自动生成[J]. 计算机应用, 2008, 28(3): 723-725.
- [10] Khurshid S, Suen Yuk Lai. Generalizing Symbolic Execution to Library Classes[J]. ACM SIGSOFT Software Engineering Notes, 2006, 31(1): 103-110.
- [11] 常瑞花, 张力, 慕晓冬, 等. 基于遗传算法的结构测试数据自动生成[J]. 火力与指挥控制, 2009, 7(3): 76-78.
- [12] 高海昌, 冯博琴, 朱利, 等. 改进的遗传算法在测试数据自动生成中的应用[J]. 系统工程与电子技术, 2006, 28(5): 1077-1081.