

基于粒子群算法的测试用例自动生成方法研究

贾冀婷

(西安邮电学院 计算机学院, 陕西 西安 710121)

摘 要:软件测试是保证软件质量、提高软件可靠性的关键,而提高生成测试用例的自动化程度又是提高软件测试自动化程度的关键。为了提高生成测试用例的自动化程度,改进了基本粒子群优化算法,提出了基于改进的粒子群优化算法的测试用例自动生成系统框架,并给出了基于改进的粒子群优化算法的测试用例自动生成算法。实验结果表明,使用文中提出的算法,自动生成测试用例所需的迭代次数和平均运行时间明显优于遗传算法等测试用例自动生成算法,一定程度上提高了生成测试用例的自动化程度。

关键词:软件测试;自动测试;测试用例;粒子群

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2010)09-0024-04

Research of Automatic Testcase Generation Functions Based on Particle Swarm Optimization Algorithm

JIA Ji-ting

(Computer Department, Xi'an University of Posts and Telecommunications, Xi'an 710121, China)

Abstract: The software testing is an important step that guarantees software quality and reliability, and improving the automation ability of testcase generation is the key point for the entire process. In order to improve the automation ability of testcase generation, has improved particle swarm optimization algorithm and has put forward the frame of automatic testcase generation system based on particle swarm optimization algorithm, and automatic testcase generation functions based on particle swarm optimization algorithm. Experiment has proved that this improved algorithm's convergence speed is generally faster than other algorithms such as genetic algorithm, and improving the automation ability of testcase generation.

Key words: software testing; automatic software testing; testcase; particle swarm

0 引言

随着计算机技术的不断发展,软件的规模越来越大,随之而来软件的错误也越来越隐蔽,造成的后果也越来越严重。因此,提高软件质量及可靠性已成为软件工程领域的重要任务。而软件测试则是保证软件质量、提高软件可靠性的关键^[1]。统计表明,软件测试作为保证软件质量和可靠性的重要手段,在整个软件生命周期中占有重要的地位,占整个开发成本的60%以上。尤其在航空航天、军事国防等系统中,测试费用所占的比例更大。可见软件测试需要耗费巨大的人力、物力和时间,故提高软件测试的自动化程度对于确保软件开发质量、降低软件开发成本都是非常重要的。其中,提高生成测试用例的自动化程度又是提高软件

测试自动化程度的关键^[2,3]。

文中在基本粒子群优化算法的基础上,结合软件测试中测试用例自动生成的需求,改进了基本粒子群优化算法,提出了基于改进的粒子群优化算法的测试用例自动生成系统框架,并给出了基于改进算法的测试用例自动生成算法,一定程度地提高了软件测试自动化程度。

1 基本粒子群优化算法原理

粒子群优化算法,最初是由 Eberhart 博士和 Kennedy 博士提出,于1995年发展并作为一种随机优化方法,后来又进行了有效的拓展。其基本思想是通过群体中个体之间的协作和信息共享来寻找最优解。它包括有进化计算和群体智能的特点,是进化计算和智能优化领域的一个研究热点,并且已在神经网络训练、目标函数优化、模糊控制系统、动态环境优化等领域被广泛应用。

收稿日期:2010-02-27;修回日期:2010-05-25

基金项目:西安市科技计划项目(YF07024)

作者简介:贾冀婷(1978-),女,河北徐水人,硕士,研究方向为软件与理论、软件测试。

粒子群优化算法与其它进化算法的区别在于它是对社会行为的模拟,基本思想不是源于自然选择中进化规律的冲突,而是源于群体组织的社会行为。

粒子群优化算法是通过种群中个体之间的协作和信息共享来实现最优解的搜索。其主要步骤为:首先生成初始种群,也就是在解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,并为每个粒子都确定一个适应值^[4-6]。被确定适应值的每个粒子在运动过程中,由一个速度决定其距离和方向。每个粒子将追随当前的最优粒子经过逐代搜索从而得到最优解。在每一代搜索的过程中,每个粒子都会跟踪两个极值,其一是粒子本身迄今为止找到的最优解,其二是整个种群迄今为止找到的最优解^[7]。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。

粒子群优化算法的一般数学表示为:设在一个 n 维的空间中,有一包含 m 个粒子的种群,即 $X = \{X_1, \dots, X_i, \dots, X_m\}$, 种群中第 i 个粒子位置为 $X_i = \{X_{i1}, X_{i2}, \dots, X_{in}\}^T$, 其速度是 $V_i = \{V_{i1}, V_{i2}, \dots, V_{in}\}^T$ 。第 i 个粒子的个体极值为 $P_i = \{P_{i1}, P_{i2}, \dots, P_{in}\}^T$, 种群的全局极值是 $P_g = \{P_{g1}, P_{g2}, \dots, P_{gn}\}^T$, 依据粒子群优化算法的基本原理,种群中第 i 个粒子将根据公式(1)、(2)改变速度和位置^[8]:

$$v_{id}^{(t+1)} = v_{id}^{(t)} + c_1 r_1 (p_{id}^{(t)} - x_{id}^{(t)}) + c_2 r_2 (p_{gd}^{(t)} - x_{id}^{(t)}) \quad (1)$$

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)} \quad (2)$$

式中 $d = 1, 2, \dots, n; i = 1, 2, \dots, m$ 。 m 表示种群规模的大小; t 是当前的进化代数,表示粒子的飞行步数; r_1, r_2 是分布在 $[0, 1]$ 间的随机数; c_1 和 c_2 是加速常数或学习因子,分别调节该粒子向自身已寻找的最优位置和同伴已寻找的最优位置方向飞行的最大步长。

公式(1)第一部分为“记忆”部分,是粒子先前的速度,表示粒子当前的速度要受上一次速度的影响;第二部分为“自身认知”部分,是从当前点指向粒子自身最好点的一个矢量,表示粒子自身的经验或记忆;第三部分为“群体认知”部分,是从当前点指向种群最好点的一个矢量,表示粒子间的信息共享和相互合作。粒子正是通过自身的经验和同伴中最好的经验来决定下一步的运动。

公式(1)中的第一部分起到了平衡全局和局部搜索能力的作用;第二部分使粒子拥有了局部搜索能力,能更好地开发解空间;第三部分则体现了粒子间的信息共享,使粒子能更广阔地在空间探索。只有在这三

部分共同的作用下,粒子才能有效地搜索到最好的位置。

2 基于粒子群优化算法的测试用例自动生成方法

2.1 粒子群优化算法的改进

通过对基本的粒子群优化算法的基本原理的深入研究,文中结合软件测试中测试用例自动生成的需求对基本的粒子群优化算法进行改进。

对于一个给定的粒子,如何决定其它粒子中哪个粒子对它最具影响力,最合乎逻辑的选择当然源于对自然的观察和对动物行为的探测。在自然界中,这种最合乎逻辑的选择主要表现在两个方面:一是一个机体最容易被它邻近的别的机体所影响;二是在所有邻近的机体中,它最容易被比它更好的粒子所影响。这个粒子应该满足的两个条件是:一是它必须临近被更新的粒子;二是它自己所经历的最好位置应该比被更新粒子当前的适应值好。

由于在处理高维的复杂问题时,基本粒子群优化算法容易陷入局部最优^[9],文中针对此问题对其进行了改进,即在局部最优位置、全局最优位置对粒子当前位置影响的基础上,增加了“邻居”最优位置对粒子当前位置的影响。将种群中除此粒子外的其它所有粒子都看作此粒子的邻居,使粒子不仅能从全局最好粒子中学习经验,而且能从“邻居”即临近的粒子中适应值最好的粒子吸取经验,使粒子之间能更好地共享社会信息。

文中针对基本粒子群优化算法公式(1)进行了改进,则该公式变为:

$$v_{id}^{(t+1)} = v_{id}^{(t)} + c_1 r_1 (p_{id}^{(t)} - x_{id}^{(t)}) + c_2 r_2 (p_{gd}^{(t)} - x_{id}^{(t)}) + c_3 r_3 (p_{kd}^{(t)} - x_{id}^{(t)}) \quad (3)$$

由公式(3)、(2)组成的迭代算法为改进的粒子群优化算法。

由公式(3)和(2)可得:

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t)} + c_1 r_1 (p_{id}^{(t)} - x_{id}^{(t)}) + c_2 r_2 (p_{gd}^{(t)} - x_{id}^{(t)}) + c_3 r_3 (p_{kd}^{(t)} - x_{id}^{(t)}) \quad (4)$$

其中:下标 d 表示粒子的第 d 维; i 表示粒子; c_3 为学习因子或加速常数; r_3 为分布于 $[0, 1]$ 之间的随机数; p_k 表示第 i 个粒子的“邻居”中具有最好适应值的粒子 k 的位置;其余参数与基本粒子群优化算法公式中的参数一致。

从收敛性来分析算法的改进,公式(1)中的“自身认知”部分和“群体认知”部分使得粒子群优化算法具有一定的局部收敛能力。公式(4)中增加了“邻居”最

优位置对粒子当前位置的影响,提高了算法的局部收敛能力。该算法引导粒子朝着粒子邻域中适应值最好的粒子飞去,比基本的粒子群优化算法表现好。改进后的粒子群优化算法在避免陷入局部最优的同时还保持了较好的收敛速度。在算法中,避免了个体最好解与全局最好解或邻域最好解之间的杂交,算法的搜索效率有了很大的提高,较好地满足了软件测试中测试用例自动生成的需求。

2.2 基于改进的粒子群优化算法的测试用例自动生成系统框架

基于改进的粒子群优化算法的测试用例自动生成系统是软件测试平台的重要组成部分,主要包括测试环境构造模块、粒子群优化算法包的实现模块以及测试运行模块这三大块。

(1)测试环境构造模块是整个基于改进的粒子群优化算法的测试用例自动生成系统的基础。它主要完成通过对被测程序的静态分析获取有用的参数以及对被测程序进行函数插桩。

(2)粒子群优化算法包的实现模块是测试用例生成系统的重要组成部分。它首先对测试环境构造模块中获取的有用参数及其范围确定种群规模的大小,并根据编码规则进行编码,产生初始种群,然后由测试运行模块部分得到的分支跟踪信息计算适应值,按照个体评价规则,种群循环调用粒子群优化算法产生新一代的种群,直到条件终止。

(3)测试运行模块是测试环境构造模块和粒子群优化算法包实现模块的桥梁与实现。它主要完成实时调用并运行函数插桩后的被测程序,得到分支跟踪信息,将该信息输出给粒子群优化算法包的实现模块,并根据粒子群优化算法包模块中的个体评价结果决定程序是运行还是终止。

2.3 适应值函数构造

如何构造适应值函数是粒子群优化算法应用于解决问题的优化目标。适应值函数的构造将直接影响到粒子群优化算法搜索寻找测试数据的速度。一个好的适应值函数将有利于种群中每个粒子向最优解空间进行搜索。

构造适应值函数需要满足以下几个条件:

- (1)单值、连续、非负、最大化。
- (2)合理、一致性。
- (3)计算量小。
- (4)通用性强。

适应值函数的构造成功与否直接影响到算法的成功与否。适应值函数的函数值表征了某个个体对应参数域中的测试用例的测试效率,适应值高的个体对应

的测试用例测试效率就高,这样的测试用例可以在更短的时间内覆盖尽可能多的构件化软件路径,发现更多的构件化软件错误。

文中采用分支函数叠加的方法来构造适应值函数。分支函数是一个实值函数,它是分支谓词到实值的一个映射,能量化地描述在测试数据的驱动下,被测单元的实际执行路径对选定路径的覆盖程度^[10,11]。其中, $F(x)$ 是一个实值函数,当分支谓词为假时, $F(x)$ 为正或零;当分支谓词为真时, $F(x)$ 为负或零。

基于本系统的特点,构造适应值函数如下:

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: $f_1 = f_1(x_1, x_2, \dots, x_n)$, $f_2 = f_2(x_1, x_2, \dots, x_n)$, \dots , $f_m = f_m(x_1, x_2, \dots, x_n)$ 。而该路径的分支函数为:

$$F = C - (F(f_1) + F(f_2) + \dots + F(f_m)) \quad (5)$$

其中, $F(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases}$; C 为一个较大的整数。

2.4 基于改进的粒子群优化算法的测试用例自动生成算法

基于改进的粒子群优化算法的测试用例自动生成算法,将测试数据作为粒子群向量 x 的元素。首先随机生成测试数据,然后用改进的粒子群优化算法搜索最佳的测试数据,使得适应值达到最大。

算法具体步骤如下:

(1)分析被测程序,确定适应值函数,对被测程序插桩;

(2)选定粒子数 m 、适应值阈值、最大允许迭代次数 T 、 c_1 、 c_2 、 c_3 和 r_1 、 r_2 、 r_3 ;

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

(4)while($F_g \leq \epsilon$ & $t < T$)

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

{使用粒子群中的每一个粒子来执行插桩后的程序;根据粒子运行之后的结果,评定粒子的适应值:

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

if ($F_i > F_k(i)$) { $F_k(i) = F_i$; $p_k = x_i$; }

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

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

{按公式(3)计算 v_i ;按公式(2)计算 x_i ;

(7) $t = t + 1$;

在上述步骤中, F_i 是第 i 个粒子的适应值, x_i 为其对应的粒子位置; F_p 是第 i 个粒子迄今搜索到的最优适应值,其对应的粒子位置是 p_i ; F_k 是第 i 个粒子的“邻居”中具有最优适应值的粒子 k 的适应值,其对应的粒子位置是 p_k ; F_g 是粒子群迄今搜索到的最优适应值,其对应的最优粒子位置 p_g 。

3 实验及结果分析

以一个简单判断三角形类型的程序作为被检测程序,使用文中的基于改进的粒子群优化算法的测试用例自动生成算法自动生成测试用例,并与遗传算法、人工免疫算法的测试实验数据进行了比较。实验结果采用迭代次数和运行时间作为评价指标^[12],生成所有目标路径测试用例的迭代次数越少、时间越短,说明算法效果越好。

从实验可以看出,种群规模越大,找到全部测试用例所需的迭代次数越少,也就是说,种群规模越大则多样性越好,越容易找到路径的测试用例。但是,种群规模越大,进行一次运行所需的时间越多。所以,在实际应用中要综合考虑这两者的因素。

表1是文中算法和遗传算法、人工免疫算法在粒子群数和种群数从10到50时,生成最优解的迭代次数情况及相应的运行时间。实验中对每个粒子群数和种群数各运行10次。表1记录了找到最优解迭代次数的平均值,生成最优解所需的时间则是10次运行的平均值。

表1 文中算法与遗传算法、人工免疫算法生成三角形测试数据的迭代次数和时间

粒子群数 (种群数)	找到最优解迭代的次数			找到最优解迭代的时间(ms)		
	文中 算法	遗传 算法	人工免 疫算法	文中 算法	遗传 算法	人工免 疫算法
10	23.5	263.3	123.2	20.6	323.5	187.1
20	19.5	239.7	110.6	19.2	286.5	162.5
30	16.7	152.9	82.1	19.5	265.4	155.7
40	9.9	161.5	87.3	18.2	268.3	157.1
50	11.7	116.5	70.5	17.6	227.7	133.9

从表1可以看出文中算法的平均迭代次数和平均运行时间都比遗传算法、人工免疫算法的要少,文中算法生成所需测试用例的迭代次数大约是遗传算法的1/11、人工免疫算法的1/5,平均运行时间大约是遗传算法的1/16、人工免疫算法的1/8。实验证明,文中改进的粒子群优化算法生成测试用例的效果明显较好。

4 结束语

文中提出的基于改进的粒子群优化算法的测试用例自动生成算法具有算法简单、易实现、设置参数少等特点,能更好地应用在软件测试平台的系统中。大大

提高了测试用例自动生成效率,缩短了开发周期,降低了开发成本。同时也提高了算法的局部收敛能力,能更快地收敛于最优解,算法的搜索效率有了很大的提高,非常适合软件测试中测试用例自动生成的应用。

由于时间关系,文中只实现了数值类型的数据,对其它类型的数据比较难处理,还需要进一步进行研究,实现更大程度的自动化。

参考文献:

[1] Gallagher M, Narasimhan V L. ADTEST: A Test Data Generation Suite for Ada Software Systems[J]. IEEE Transactions on Software Engineering,1997, 23(8):473-484.

[2] Mosley D J, Posey B A. 软件测试自动化[M]. 邓波,等译.北京:机械工业出版社,2003.

[3] 朱少民. 软件测试方法和技术[M]. 北京:清华大学出版社,2005.

[4] Kennedy J, Eberhart R. Particle Swarm Optimization[C]// In: Proc IEEE Int Conf on Neural Networks. [s.l.]:[s.n.], 1995:1942-1948.

[5] 朱玉平. 一种改进粒子群优化算法[J]. 计算机技术与发展,2008,18(11):106-108.

[6] 王艳玲,李龙澍,胡哲. 群体智能优化算法[J]. 计算机技术与发展,2008,18(8):114-117.

[7] Clerc M. The Swarm and the Queen: Towards a deterministic and Adaptive Particle Swarm Optimization[C]// Proceedings of 1999 Congress Evolutionary Computation. Piscataway, NJ:IEEE Press,1999:951-957.

[8] Kennedy J, Eberhart R C. A New Optimizer Using Particle Swarm Theory[C]// Proceedings of the sixth International Symposium on Micro Machine and Human Science. Nogyoya, Japan:[s.n.],1995:39-43.

[9] Shi Y, Eberhart R C. A Modified Particle Swarm Optimizer [C]// Proceedings of the IEEE Congress on Evolutionary Computation. [s.l.]:[s.n.],1998:69-73.

[10] 万琳. 面向C程序的测试用例自动生成实现[J]. 火力与指挥控制,2006,31(10):73-76.

[11] 郑钧泽,徐晓峰,郭东辉. 基于克隆选择算法的面向程序路径测试数据生成方法[J]. 计算机技术与发展,2009,19(8):8-10.

[12] 马臻. 基于遗传算法的构件化软件测试用例生成研究[D]. 西安:西安理工大学,2006.

(上接第23页)

[9] 李培根. 制造系统性能分析建模-理论与方法[M]. 武汉:华中理工大学出版社,1998.

[10] van der Aalst W M P. The application of Petri nets to work-flow management[J]. The Journal of Circuits Systems and

Computers, 1998, 8(1):21-66.

[11] 王璞. 流程再造[M]. 北京:中信出版社,2005.

[12] 佩帕德 J, 罗兰 P. 业务流程再造[M]. 第2版. 北京:中信出版社,1999.