第26卷 第6期
2011年12月

郑州轻工业学院学报(自然科学版)
JOURNAL OF ZHENGZHOU UNIVERSITY OF LIGHT INDUSTRY(Natural Science)

Vol.26 No.6
Dec. 2011

文章编号:1004-1478(2011)06-0031-04

基于GATS算法的面向对象测试用例自动生成

王倩, 张锦华

(周口师范学院 计算机科学系, 河南 周口 466001)

摘要:在遗传算法的基础上,引入禁忌搜索算法,提出了一种面向对象测试用例自动生成的方法.该方法设计了一种新的类对象编码方式,并在此基础上构造了类测试用例自动生成所需的适应度函数,使每一个测试用例在局部区域中再次寻找最优值,从而改进整体算法搜索最佳值的能力.实验结果表明,该方法结合遗传群体优化和禁忌搜索较强的爬山能力,能够实现快速全局优化,自动生成高质量的测试用例.

关键词:面向对象测试用例;遗传算法;禁忌搜索

中图分类号:TP311 **文献标志码**:A

Automatic generation of object-oriented test case based on GATS algorithm

WANG Qian, ZHANG Jin-hua

(*Dept. of Comp. Sci., Zhoukou Normal Univ., Zhoukou* 466001, *China*)

Abstract: Based on genetic algorithm and introducing tabu search algorithm, a method which is test cases automatically generating for an object-oriented class was proposed. The method designs a new code of class object and constructs the fitness function for the test cases automatically generating of class objects required. Each test case in the local area finds the optimal value again, therefore the ability of search for the best value of overall algorithm was improved. The experiment results showed that by this algorithm, the combination of optimization of genetic groups and tabu search strong climbing ability quickly realized the global optimization and generate high-quality test cases.

Key words: object-oriented test case; genetic algorithm(GA); tabu search(TS)

0 引言

随着软件规模的不断扩大,高效率和高质量的软件开发成为现代软件工程学研究的热点.在软件测试技术中,高效的测试用例生成是简化测试工作、提高测试效率的必要手段.早期生成的测试用例数量多且测试效率低,所以需要一种好的优化算法对早期生成的测试用例进行优化.1980年代D. L. Bird等[1]采用随机法自动生成测试用例,但生成的测试数据集是一个庞大的集合,没有针对性.随着人工智能技术的不断发展,遗传算法GA(genetic algorithm)的优越性不断体现出来.文献[2-4]提出了如何将GA应用到生成最小测试集合中,文献[4]提出了如何把目标转换为覆盖准则.但是GA本身存在优化效率较低等局限性,因此GA只能在短时间内找到接近全局最优解的近似解,不能保证收敛

收稿日期:2011-05-08

作者简介:王倩(1984—),女,河南省周口市人,周口师范学院助教,硕士,主要研究方向为软件工程.

到全局最优解.鉴于此,本文拟在GA的基础上,引入禁忌搜索TS(tabu search)算法,以期实现高质量的面向对象测试用例的自动生成.

1 相关算法

1.1 GA

GA首先由随机得到的一些染色体算出适应度,然后对这些染色体执行遗传操作,保留适应度高的染色体,去除适应度低的染色体.新的群体具有上一代的优秀基因,故比上一代强.反复执行遗传算法,朝向更优解的趋势,直至满足某种预定的收敛条件.GA的步骤如下:

1)对最大迭代数N,初始群体数M,迭代数t进行初始化.

2)计算每个个体的适应度值.

3)进行遗传操作(选择、交叉、变异),运算之后得到下一代群体.

4)如果$t>N$,在进化的过程中输出最优解,即获得最大适应度值的个体,结束运算;否则,迭代数t加1,转到步骤2).

1.2 TS

禁忌搜索的思想于1986年由F. Glover[5]提出.TS是一种全局逐步寻优算法,是对人类智力过程的一种模拟.TS算法通过灵活的存储结构以及相应的禁忌准则来避免迂回搜索,且由藐视规则实现某些被禁忌的优良状态的赦免,进而保证全局优化并且多样化的有效性.

TS也是一种迭代搜索,搜索速度快,且具有记忆功能,在搜索过程中可以接受劣解,具有很强的爬山能力,使用禁忌算子可以很好地避免陷入GA局部极小值.但TS也有缺陷,它对于初始解具有较强的依赖性且搜索过程过于单一.

2 现有混合算法分析

GA和TS都是基于种群的算法,并且在不同学科得到了证实.遗传个体算法的优质基因通过遗传操作得到不断提升,如果某个个体的适应度值不高,会被淘汰,而且它的一些有用的经验也将同时失去,所以GA缺乏精确找到最优解的能力.而TS在搜索过程中可以保留劣质解,摆脱局部最优解,是一种全局迭代寻优并且具有很强的局部搜索能力的算法,但是对初始解有较强的依赖性[6-10].

文献[11]在文献[12]的基础上将TS引入到GA中,把TS作为遗传算法的变异算子和交叉算子,从而改善GA爬山能力较弱的缺陷.

3 GATS算法在面向对象类测试中的应用

3.1 GA的改进

所编写的类以后要应用于不同的环境,这是面向对象类测试的难点所在,所以,要进行足够的模拟测试.文献[13]提出了将GA用于面向对象类测试.面向对象程序是成员方法的调用,在类测试里,GA的基因就是成员方法不同的调用序列.文献[13]中的算法有2点不足之处:

1)在进行编码操作时,GA中的染色体可以分为类构造、类状态设置、类测试3部分,其顺序是不变的.正是由于这一点,会遗忘如下情况:

$$*m = M(),m.f(),$$

$$*n = N(m),n.f()\cdots n.underTest()$$

其中,M,N属于2个类;*underTest*()方法是类N的待测试的方法.由遗忘情况得知,m调用了f方法,然后n执行构造函数,所以,这3部分顺序不变的要求是不能完全满足的.

2)通过计算每个个体的适应度值,划出分隔点,淘汰适应度值小的,留住适应度值较大的,可以得到遗传操作中的选择算子.操作虽然较简单,但是存在GA早熟现象,易陷入局部最优.

本文针对以上不足,从3个方面对文献[13]中的算法加以改进:1)在全局范围内染色体3部分的顺序不是不变的,而是制定某个类的顺序,从而避免遗漏.2)进行遗传操作中的选择算子时,要想到精英基因存在的情况,故在GA里引入精英策略.3)把混合算法的思想引入面向对象类测试中,并做修改,使之能适应类的测试.

3.2 构造适应度函数

根据类对象的特点,本文设计一种新的类对象编码方式并在此基础上构造类测试用例自动生成所需的适应度函数.

一个测试用例是构造函数和方法调用的序列,包含参数值.执行完测试用例后断言预期状态.类测试使用的染色体应含有执行该序列规范的操作以及相关参数的值.依据同样的规则,它的序列是变化的而且必须构建.

测试用例在GA中的对应物是染色体,它应具有一序列被执行的操作以及相关的参数值.测试用例在GA中对应物的语义如下:用"{}"和"?"表示选择扩展,"< >"表示非终端符号,"@"符号把一个测试用例分为构造函数、方法调用序列和实际输入值3部分.

<*chromosome*> ::= <*actions*>@ <*values*>

<*actions*> ::= <*action*>{; <*actions*>}?

//各个<*action*>可创建1个新对象并赋1个染色体变量($id)

//各个<*action*>可作为标识为$id的对象的成员函数的调用

//各个<*action*>还可创建基本数据类型的变量

<*action*> ::= $id = constructor({ <*parameters*>}?)

| $id = claa#null

| $id. method({ <*parameters*>}?)

<*parameters*> ::= <*parameter*>{, <*parameters*>}?

<*parameters*> ::= builtion-type{ <*generator*>}?

| $id

<*generator*> ::= [low;up]

|[genClass]

// <*generator*>是根据指定范围随机生成数据

<*values*> ::= <*value*>{, <*value*>}?

<*value*> ::= *integer*

| real

| boolean

| string

如下是本文所提出的用于表示面向对象类对象的编码:

$$F(x)=f(x.a):g(x.b):k(x.c):\cdots@x.a:x.b:x.c:\cdots$$

比如,triangle(int,int,int) @100,200,300.它代表通过带参数的成员函数对其成员变量进行初始化,实际参数为100,200,300.提出的编码可做如下分解:

$$\begin{cases}F_1(x)=f(x.a):g(x.b):k(x.c):\cdots@x.a:0:0:\cdots\\F_2(x)=f(x.a):g(x.b):k(x.c):\cdots@0:x.b:0:\cdots\\F_3(x)=f(x.a):g(x.b):k(x.c):\cdots@0:0:x.c:\cdots\end{cases}$$

每个成员变量或者成员对象单独拆分并用二进制表示,其余的部分用0表示,可得出路径集合$P=\{P_{F_1},P_{F_2},P_{F_3},\cdots\}$是根据$F_1(x),F_2(x),F_3(x),\cdots$所执行的程序.对于目标路径$P_i$,计算执行路径与目标路径的相似度集合$similarity=\{similarity_{i-F_1},similarity_{i-F_2},similarity_{i-F_3},\cdots\}$.验证如下.

设n表示面向对象类测试的总数,n'表示类对象所产生的成员变量个数,n''表示经过P的节点数.$t=o(t)$是执行程序所经路径中各个节点所需的时间复杂度,$t'=o(t^2)$是面向对象类对象进行的遗传操作所需的时间复杂度.依据文献[14]中的算法,得知生成测试用例所需时间为

$$Time=n*t*n''+n*n'*t'*n''$$

式中,$n*t*n''$表示n个面向对象类对象经过某个路径的各个节点所需时间,$n*n'*t'*n''$表示n个面向对象类对象中各个成员变量都参与遗传操作时所需的时间.

改进面向对象类对象生成测试用例所需时间,可得公式

$$Time'=n*n'*t*n''+n*(n'-1)*t'*n''$$

面向对象类对象的各个成员需要运行每个程序,故与$Time$中$n*t*n''$相比,$Time'$公式的$n*n'*t*n''$多了个成员变量的个数n'.

$$\begin{aligned}Time-Time'&=(n*t*n''+n*n'*t'*n'')-\\&[n*n'*t*n''+n*(n'-1)*t'*n'']=\\&n*(1-n')*t*n''+n*t'*n''=\\&n*t'*n''-n*(n'-1)*t*n''\end{aligned}$$

因为

$$\frac{n*t'*n''}{n*(n'-1)*t*n''}=\frac{t'}{(n'-1)*t}=\frac{o(t^2)}{(n'-1)*o(t)}$$

其中n'是固定的,所以$\frac{o(t^2)}{(n'-1)*o(t)}\geqslant 1$,故$Time\geqslant Time'$.

证明得知这一种类对象所需的适应度函数改进算法,使GA收敛更快且消耗时间更少.

3.3 遗传禁忌算法(GATS)

GA具有爬山能力弱等缺点,对其进行改进较为典型的策略就是精英保留.

本文提出了将GATS算法用于面向对象测试用例自动生成.GATS引入TS算子,使GA的爬山能力增强,有效地避免了"早熟"现象.该算法采用了精英保留策略,并对遗传操作的相应策略采用适合于面向对象类测试用例自动生成的策略.GATS算法加快了遗传算法的收敛速度并提高了GA的效率.

3.4 仿真结果

以"判断一个三角形是否为等腰三角形"的程序为实验对象,对GATS算法的性能进行分析.

本文采用Java语言编写.GATS算法中初始值

如下:"@"符号把类对象使用的编码分开;$p_c=0.6$,$p_m=0.09$;面向对象类对象的编码的初始数是30;GATS算法的最大迭代数是110.

图1是随机法、GA和GATS算法用于测试用例的函数覆盖率仿真实验图.

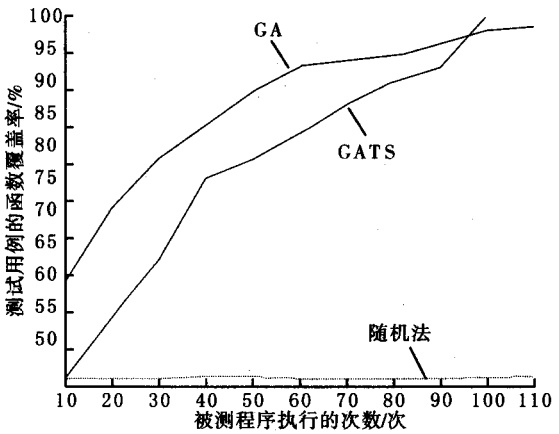

图1 随机法、GA和GATS算法性能比较仿真结果

一般情况下,GATS算法不能很快达到某个函数的覆盖率的点,而GA可以.因此,从图1可以看出,GATS算法刚开始会在GA之下,但GA易陷入局部最优,不能到达全局最优,所以GA用于测试用例自动生成的函数覆盖率达不到100%.改进的GATS算法刚开始落后于GA,是由于引入TS,使改进算法的时间复杂度较高,但随着被测程序执行的次数增多,GATS算法优于GA,在迭代100次时,GATS算法的覆盖率能达到100%.随机法本身的不足使其不能达到函数的覆盖率.

4 结语

根据类对象的特点,本文设计了一种新的类对象编码方式,并在此基础上构造了类测试用例自动生成所需的适应度函数.由时间复杂度对比分析可知,改进的方法所用时间少、促进遗传算法更快收敛.同时本文通过引入TS,在GA生成测试结果后继续采用TS来加强每个测试用例的区域搜索能力,进而获得最优解.与此同时,本文在提出的混合算法中,又根据面向对象程序的特点,在类测试的这一层,改进了上述的混合算法,明确提出了自己的算法,使之能应用到类测试中.

参考文献:

[1] Bird D L,Munoz C U. Automatic generation of random self-checking test cases[J]. IBM Systems J,1983,22(3):229.

[2] Sthamer H. The automatic generation of software test data using genetic algorithms [D]. Pontyprid: University of Glamorgan,1996.

[3] Michael C,Mcgraw G. Automated software test data generation for complex programs[C]//Proc of the 13th IEEE Int Conf on Automated Software Eng(ASE98). Washington DC: IEEE Computer Society,1998:136-146.

[4] Pargas R,Harrold M J,Peck R. Test-data generation using genetic algorithms [J]. J of Software Testing Verifications and Reliability,1999(9):263.

[5] Glover F. Tabu search part i ii ORSA[J]. J on Comp,1989(3):1.

[6] 杨海峰,程和平.利用遗传禁忌算法研究换料优化[C]//第十一届反应堆数值计算和粒子输运学术会议暨2006年反应堆物理会议,哈尔滨:[s.n.],2006:155-164.

[7] 戴庆,赵艳玲.基于遗传禁忌算法的任务分配与调度的研究[J].河北科技大学学报,2007,28(4):269.

[8] 方叶祥,钱存华,蒋南云,等.基于遗传禁忌算法的双资源约束下并行生产线调度研究[J].运筹与管理,2007,16(5):153.

[9] 李玮玮,王建东,方黎明,等.基于遗传禁忌算法的贝叶斯网边定向方法[J].计算机工程,2009,35(12):178.

[10] 周万里.基于遗传禁忌算法的网格资源调度[D].济南:山东大学,2009.

[11] 钱肖英.基于遗传算法的测试数据自动生成方法的研究[D].杭州:浙江工商大学,2008.

[12] 李广,谢强,丁秋林.基于遗传禁忌算法的Ontology划分[J].计算机工程,2009,35(17):175.

[13] Tonellap. Evolutionary testing of classes[C]//Proc of ISSTA'04,New York:ACM Press,2004:119-128.

[14] 朱小梅.面向对象软件的测试技术研究[D].四川:西南石油大学,2006.