

基于 LSGA 的最小测试用例集自动生成

刘 冬^{1,2}, 靳蓓蓓³, 阙向红²

(1 皖医弋矶山医院计算机中心, 安徽 芜湖 241001; 2 华中科技大学 网络与计算中心, 湖北 武汉 430030;
3 安徽师范大学, 安徽 芜湖 241000)

摘 要: 测试数据的生成是一个复杂的问题且其技术和方法还不成熟. 根据实现语句覆盖的测试目标, 提出了最大稳定遗传算法(LSGA). 该算法充分考虑了遗传算法的稳定性并在构造适应度函数和路径编号时提出了“邻近者优先”原则和“就近路径编号”原则. 这个算法可以生成满足测试目标的最小用例集且其性能明显优于遗传算法.

关键词: 测试用例集; 测试用例; 基本路径集; 最大稳定遗传算法; 遗传算法; 软件测试

中图分类号: TP301

文献标识码: A

文章编号: 1000-7180(2011)12-0115-04

Automatic Generation of Minimal Test Suite Based on LSGA

LIU Dong^{1,2}, JIN Bei-bei³, QUE Xiang-hong²

(1 Computing Center, Yijishan Hospital of Wannan Medical College, Wuhu 241001, China;
2 Computing Center, Huazhong University of Science and Technology, Wuhan 430030, China;
3 Anhui Normal University, Wuhu 241000, China)

Abstract: Test data generation is a complicated problem and its method and technique are not mature. According to the test target, which achieves statement coverage, this paper proposes the largest steady genetic algorithm (LSGA). It considers the steady of GA and this paper proposes the “the neighbor first” principle and “nearby-path numbering” principle when fitness function is built and path is numbered. It can generate the minimal test suite to meet the test target and its performance is superior to GA.

Key words: test suite; test case; basic path suite; largest steady genetic algorithm; genetic algorithm; software testing

1 引言

软件测试是保证软件质量和可靠性的重要手段和方法, 然而它是一个复杂的过程, 测试用例的自动生成问题是一个非常困难的问题, 目前还没有一种比较完美的解决方案. 自从遗传算法的出现, 人们也不断尝试将遗传算法引入软件测试中. 文献[1-3]中将遗传算法(GA)用于测试用例自动生成问题的解决, 并给出了相应的性能分析. Michael^[4]通过两个具体的实例揭示了遗传算法的搜索性能明显优于梯度下降法(随机搜索算法). 文献[6-7]表明, 遗传

算法生成的测试数据数量比随即法少. 以前的研究成果都是解决测试用例满足某个测试需求的问题. 也就是针对某个测试需求, 利用各种生成算法生成满足该测试需求的测试用例. 例如面向路径的测试数据自动生成研究^[5], 为了实现路径覆盖, 由测试人员指定一条路径, 利用模拟退火遗传算法或其他启发式算法对该路径进行测试生成一个测试用例, 满足覆盖此路径, 而要生成覆盖不同路径的测试用例, 需要不断由测试人员指定测试路径. 这样就存在一个问题, 如果要覆盖待测程序中所有路径, 就必须指定所有可能的测试路径.

收稿日期: 2011-03-18; 修回日期: 2011-05-21

基金项目: 安徽省教育厅自然科学基金(KJ2010B363); 皖南医学院中青年科研基金(WK201038F)

根据实现语句覆盖的测试目标,结合基本路径测试技术和程序插装,本文提出了最大稳定遗传算法(LSGA),并将该算法用于最小测试用例集的自动生成问题,然后构造算法实现的核心技术,并在构造适应度函数和路径编号时提出了“邻近者优先”原则和“就近路径编号”原则.这样就可以生成一个满足测试目标的最小测试用例集.

2 问题描述

在软件测试中,测试用例的生成问题可以描述如下:

定义 1 给定一个程序或软件 P , 测试目标 R , 测试需求 $r_i (r_i \in R \text{ 且 } r_i \neq r_j)$, 设 P 的输入空间为 T , 对于每个测试需求 r_i , 都至少存在一个 $t (t \in T)$, 满足 r_i . 通过一定的测试策略求得测试 P 的一组测试用例集 T' , $T' = \{t_1, t_2, \dots, t_m\}$, ($T' \subset T$), 其中测试用例集 T' 和测试目标 R 满足关系 $S(T', R')$, 即: $S(T', R) = \{(t, r) \in T' \times R, \text{测试用例 } t \text{ 满足测试需求 } r\}$.

针对前面存在的问题我们可以给出问题的形式化定义.

定义 2 构造测试 P 的一组测试用例集 T' , $T' = \{t_1, t_2, \dots, t_m\}$, ($T' \subset T$), 使 T' 能满足 R , 且保证 T' 规模最小, 则称 T' 就是满足 R 的最小测试用例集.

3 LSGA 自动生成算法

为了增加遗传算法的稳定性,本文在基本遗传算法的基础上提出了最大稳定遗传算法(Largest Steady Genetic Algorithm, LSGA).

3.1 LSGA 算法定义

本文的 LSGA 的形式化定义如下所示.

定义 3 LSGA 算法可以定义为一个 9 元组:

$LSGA = (C, F, P_0, M, \Phi, \Gamma, \Psi, \Pi, T)$

其中, C : 个体的编码方法;

F : 个体适应度评价函数;

P_0 : 初始种群;

M : 群体大小;

Φ : 选择算子;

Γ : 交叉算子;

Ψ : 变异算子;

Π : 稳定数量;

T : 算法终止条件.

3.2 LSGA 算法思想

根据以上 LSGA 算法定义可以给出 LSGA 算法的算法描述.

第 1 步: 计 $K = 0$, 随机产生初始种群.

$P_0 = (X_1(0), X_2(0), \dots, X_n(0))$.

第 2 步: 用产生的种群运行被测程序.

第 3 步: 计算 $P(K)$ 中每个个体的适应度.

第 4 步: 个体评价. 从 K 代种群中选择出最优的 $m (m \geq 0)$ 个个体 $X_i(K)$, ($i = 1, 2, \dots, n$), 将 $X_i(K) \rightarrow X_j(K+1)$, ($j = 1, 2, \dots, m$).

第 5 步: 依据适应度值, 再从种群 $P(K)$ 中取出 $n-m$ 对个体, 进行杂交可以得到 $n-m$ 个新个体.

第 6 步: 独立地对这 $n-m$ 个新个体进行变异, 得到下一代种群 $P(K+1)$ 中的后 $n-m$ 个新个体 $X_i(K+1)$, ($i = 1, 2, \dots, n-m$), 并将这 $n-m$ 个新个体并入到新种群 $P(K+1)$ 中, 即:

$P(K+1) = \{X_1^*(K), X_2^*(K), \dots, X_m^*(K), X_1(K+1), X_2(K+1), \dots, X_{n-m}(K+1)\}$

第 7 步: 检验停止标准, 若满足则停止, 否则 $K = K+1$, 并返回第 2 步.

3.3 基于 LSGA 的测试用例自动生成模型

本文首先利用随机生成初始测试用例集, 然后对测试用例集中的每个测试用例进行二进制编码得到初始种群, 再利用 LSGA 算法对初始种群进行优化, 采用一定的优化策略就可以产生高效率的测试用例组成最小测试用例集. 如图 1 所示的是基于 LSGA 算法的面向独立路径的测试用例自动生成模型(Model of Test Suite Generation of Path-Based Software, MTSPBS). 从图中可以清楚地看出 LSGA 算法是如何对初始测试用例集进行优化进而得到最小的测试用例集. 测试用例集中的测试用例用于被测程序的运行, 每一个测试用例运行后都会返回该测试用例通过的路径信息, 然后 LSGA 算法利用适应度函数进行评价.

3.4 编码参数的选择

如何确定编码参数, 也是制约算法效率的一个因素. 对于一个待测程序, 并不需要将其中所有的变量都进行编码, 否则会造成系统资源的浪费, 从而降低算法性能. 具体选择什么样的参数进行编码, 可以参考一个原则, 即“有用参数编码原则”, 也就是只选取与测试单元中影响路径的相关变量进行编码, 其它变量一概不进行编码.

3.5 构造适应度函数

适应度函数是 LSGA 算法和实际问题的接口,

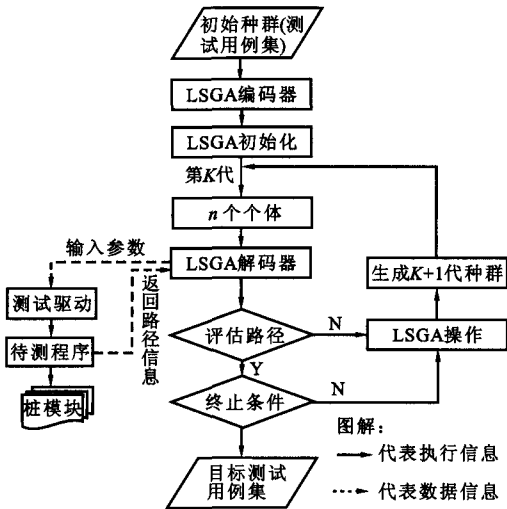


图1 MTSBS模型

在算法的应用过程中,利用适应度函数实现最优解的评价.本文中为了获得覆盖被测程序中所有独立路径的最小测试用例集,采用了记录每个测试用例所经过的路径,依据“邻近者优先”的原则,动态修改每个个体的适应度值.具体实现步骤如下:

第1步:初始化初始种群中每个个体的适应度,一般初始化为被测程序所要覆盖的独立路径数 C_{max} .

第2步:每个个体运行被测程序,记录所经过的路径 $p(x)$ 最后统计还尚未经过的路径号为

$$f_i(x), (i = 1, \dots, C_{max})$$

第3步:重新计算种群中每个个体的适应度 $F(x)$,其中,

$$F(x) = \begin{cases} C_{max} - \min(|p(x) - f_i(x)|), \\ p(x) > C_{max} (i = 1, \dots, C_{max}) \end{cases} \quad (1)$$

第4步:终止条件是否满足,若是则结束,否则进入下一代操作,重复第2步.

通过上面的过程,就将独立路径的覆盖问题转化成相应的目标函数的最优解问题.从式(1)可以知道“邻近者优先”的原则就是与未通过路径的相邻程度越强,其适应度也越大.这样,在对独立路径进行编号时,要遵照路径重复最多的两个或三个路径的路径号相邻的原则.这一路径编号原则具体可以通过程序插装技术来实现.

3.6 评估函数和终止条件

这里给出测试用例集中一个好测试用例的定义.

定义4 如果测试用例 t_i 通过了测试用例集中 T' 其他测试用例 t_j 没有通过的路径,则认为测试用例 t_i 是一个好的测试用例.

当前代测试用例集中认为是好的测试用例的个数也决定了 LSGA 算法中的稳定数量 N_s .

由此就可以确定 LSGA 算法的终止条件,即如果测试用例集中的测试用例经过了基本路径集中的所有独立路径,算法就终止执行.

3.7 路径编号

为了配合适应度函数的计算,需要通过程序插装技术在被测程序中的每条独立路径上插入一段代码,用来记录每个测试用例通过的路径号.在对路径编号的过程中,首先人工分析待测程序,得出待测程序的流图,然后根据流图分析出待测程序中一共有多少条独立路径,即得到待测程序的基本路径集.最后按照路径重复最多的两个或三个路径的路径号相邻的原则来进行编号,即:“就近路径编号”原则.

4 实例研究

采用 C 语言实现 MTSPBS 模型系统原型.利用软件测试文献中比较常用的直角三角形判定程序进行实验.

4.1 实验参数设置

实验1:LSGA算法的运行参数初始化为参数编码采用三参数级联二进制编码,每个参数的取值范围为 $[0, 15]$,且编码精度为1,每个参数用4位表示,编码长度共12位;交叉概率为0.9,变异概率为0.25;基本路径集大小为5;初始种群大小分别为6、10和20.

实验2:LSGA算法的运行参数初始化为参数编码采用三参数级联二进制编码,每个参数的取值范围为 $[0, 15]$,且编码精度为1,每个参数用4位表示,编码长度共12位;交叉概率为0.9,变异概率为0.15;基本路径集大小为5;初始种群大小为6、10和20.

4.2 实验结果及性能分析

实验1的结果如图2所示.

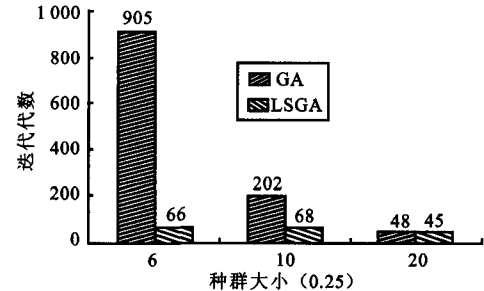


图2 实验一算法性能对比图

变异率为0.25.从图中可以明显地看出对于同

种群大小的遗传操作中,得到最小测试用例集的迭代次数,LSGA 算法明显少于 GA 算法,即:LSGA 算法明显优越于 GA 算法.另外,从图中可以看出,对于种群大小为 6 和 10 的两个种群,LSGA 算法获得最小测试用例集所用的迭代次数明显少于 GA 算法,然而种群大小为 20 的种群,其区别并不是很明显.说明当提高种群规模时,LSGA 算法和 GA 算法可以同时处理更多的解,因而可以容易找到全局最优解.同时这里的变异率设置较高,导致了种群的不稳定.综上所述,LSGA 算法对于大种群大变异率的寻优能力比 GA 算法优越,但是并不能很充分地体现其优越性.

为了提高种群的稳定性,实验二中降低变异率,其结果如图 3 所示.

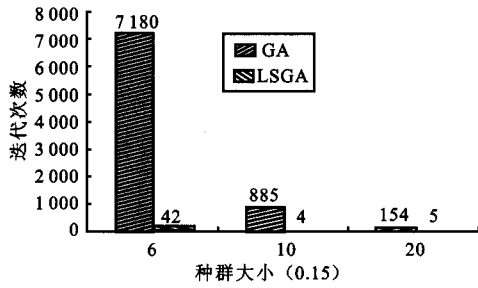


图 3 实验二算法性能对比图

其变异率为 0.15,从图中可以很清晰地看出,降低变异率保证了种群的稳定性,LSGA 算法的迭代次数明显少于 GA 算法.

5 结束语

本文针对软件测试中最小测试用例集的自动生成问题提出了一种改进遗传算法——最大稳定遗传算法(LSGA).首先通过多参数级联二进制编码实现问题的描述,然后构造适应度函数,提出“邻近者优先”原则,为了配合适应度函数的计算,本文又针对测试需求提出了“就近路径编码”原则,通过程序插装技术实现路径编码,最后分别利用 GA 算法和 LSGA 算法进行优化搜索.实验结果证明 LSGA 算法能保证种群的最大稳定性,具有较强的全局收敛性,同时有效提高了算法的局部搜索能力.

参考文献:

- [1] 胡燕京,张毅坤,朱伟,等.嵌入式软件功能路径测试用例自动生成研究[J].微电子学与计算机,2010,27(1):81-85.
- [2] 贾小云,韩薇.应用软件中自动化测试的实现[J].微电子学与计算机,2010,27(9):165-167.
- [3] Hermadi I, Ahmed MA. Genetic algorithm based test data generator. Evolutionary Computation[C]// proceedings of The 2003 Congress on Evolutionary Computation. [s. l.]; IEEE, 2003:85-91.
- [4] 汪浩,谢军凯,高仲仪.遗传算法及其在软件测试数据生成中的应用研究[J].计算机工程与应用,2001,37(12):64-68.
- [5] 伦立军,丁雪梅,李英梅.基于遗传算法的测试数据生成研究[J].计算机工程,2005,31(23):82-84.
- [6] Michael C C, McGraw GE, Schatz M A, et al. Genetic algorithms for dynamic test data generation [C]// proceedings of the 12th IEEE International Conference on Automated Software Engineering. [s. l.]; IEEE, 1997: 307-308.
- [7] Jones B F, Eyres D E, Sthamer H H. A strategy for using genetic algorithms to automate branch and fault-based Testing[J]. The Computer Journal, 1998,41(2): 98-107.
- [8] Wegener J, Baresel A, Sthamer H. Evolutionary test environment for automatic structural testing[J]. Information and Software Technology, 2001,43(4):841-854.
- [9] Eugenia D, Javier T, Raquel B. Automated software testing using a metaheuristic technique based on tabu search[C]// proceedings of the 18th IEEE International Conference on Automated Software Engineering. [s. l.]; IEEE, 2003:310-313.

作者简介:

刘冬男, (1979-), 男, 硕士, 工程师. 研究方向为软件工程、医学信息处理.

靳蓓蓓女, (1978-), 硕士, 讲师. 研究方向为信息安全理论与技术.

阙向红男, (19-), 副教授, 硕士生导师. 研究方向为软件工程、医学信息处理.