

文章编号:1001-9081(2008)03-0723-03

基于免疫遗传算法的软件测试数据自动生成

夏芸^{1,2}, 刘峰¹

(1. 安徽大学 计算机科学与技术学院, 合肥 230039; 2. 安徽师范大学 数学计算机科学学院, 安徽 芜湖 241000)
(940xiayun@163.com)

摘要: 提出了一种应用于软件测试中的基于免疫遗传算法(IGA)的软件测试数据自动生成的算法。该算法在传统的遗传算法中引入免疫算子, 免疫算子其中包括获取疫苗、注射疫苗和免疫选择。实验结果表明, 该算法的效果比传统的遗传算法效果好。

关键词: 免疫算法; 遗传算法; 测试数据; 软件测试

中图分类号: TP18; TP311.5 **文献标志码:** A

Automated software test data generation based on immune genetic algorithm

XIA Yun^{1,2}, LIU Feng¹

(1. College of Computer Science and Technology, Anhui University, Hefei Anhui 230039, China;
2. College of Mathematics and Computer Science, Anhui Normal University, Wuhu Anhui 241000, China)

Abstract: A kind of software test data automated generation method based on Immune Genetic Algorithms (IGA) was proposed. Immune operators including vaccine extraction, vaccination and immune selection were introduced into the traditional genetic algorithm. The experimental results show that this algorithm is more effective than the traditional one.

Key words: immune algorithm; Genetic Algorithm (GA); test data; software test

0 引言

软件测试是保证软件可靠性的主要手段, 其目的是发现软件错误、提高软件质量。软件测试的工作量一般要占到软件总开发时间的 40%~60%, 而测试中又有很大部分适于自动化, 而测试数据自动生成是实现软件测试自动化的关键问题。由于测试数据的自动生成是在一个数据域中寻找满足给定的测试标准的一组测试输入数据的过程, 所以, 出现了把测试数据的生成问题转化成搜索问题的思想。因而, 近几年一些研究者将遗传算法应用到面向路径的测试用例的搜索中。

但是随着研究和应用的深入, 遗传算法也暴露出许多缺点, 如未成熟收敛和局部搜索能力差。遗传算法 (Genetic Algorithm, GA) 基本思想是在编码后对编码空间进行搜索, 基本上不利用问题空间的知识, 而仅用适应度函数值来评价个体并指导搜索。虽具有全局并行搜索的优点, 但由于没有充分利用待求问题的自身特征, 从而造成搜索效率低下、收敛速度慢, 有时甚至无法收敛到全局最优解。而人工免疫算法采用生物免疫系统原理, 在遗传算法的基础上, 力图利用待求问题中的一些特征信息或知识来抑制迭代过程中出现的退化现象。本文将免疫遗传算法 (Immune Genetic Algorithms, IGA) 引入到面向路径的测试数据自动生成领域, 有效改善遗传算法的未成熟收敛缺陷, 提高遗传算法的全局和局部搜索能力。

1 基于 IGA 测试数据生成的模型

基于 IGA 的测试数据生成的模型, 如图 1 所示。该模型可以分为两部分: 左半部分为免疫遗传算法模块, 右半部分为测试环境的构造模块。

该模型的核心是免疫遗传算法模块, 它首先随机生成初始参数种群, 然后按照参数的编码要求对种群的个体进行解码得到每个参数的实际值, 再以当前种群的实际参数值为输入参数调用插装的被测程序, 由此得到该种群每个个体的目标函数值并返回给免疫遗传算法模块, 免疫遗传算法模块再依据该种群中的个体目标函数值进行评价, 并进行相应的交叉、变异、提取疫苗、注射疫苗和免疫选择操作, 从而形成一个新的种群, 如此往复, 直到找到覆盖选定路径的目标参数值或者达到预定的最大迭代数。

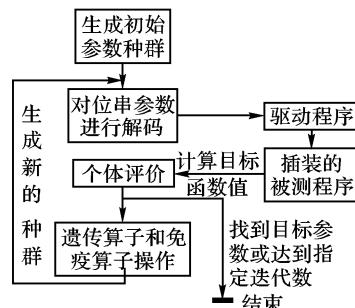


图 1 基于 IGA 的测试数据生成模型

测试环境的构造模块主要利用程序插装技术, 对被测程序模块进行插装, 通过“分支函数叠加法”^[1] 构造的方法来构造针对路径覆盖的目标函数。设被测程序的测试路径上有 m 个分支点, 对测试路径的分支点插入分支函数 f_1, f_2, \dots, f_m , 最后在被测程序模块结束处加上由分支函数叠加形成的目标函数表达式 $F = f_1 + f_2 + \dots + f_m$, 并将 F 的值返回到免疫遗传算法模块。给出如下表 1 的一个例子, 下面以对被测程序进行插装的示例来说明对被测程序的分支函数的插装和叠加。

收稿日期:2007-09-30;修回日期:2007-11-29。

作者简介: 夏芸(1978-), 女, 安徽芜湖人, 硕士研究生, 主要研究方向: 软件工程; 刘峰(1962-), 男, 安徽宿县人, 教授, 博士, 主要研究方向: 软件工程、并行分布计算、计算机网络。

被测原程序,测试路径是覆盖 printf 语句:

```

void RightTriangle( int a, int b, int d)
{
    if (a > 0 && a < 200)
        if (b > 0 && b < 200)
            if (d > 0 && d < 200)
                if (a * a + b * b == d * d)
                    printf ("This is right triangle. ");
}

分支函数插装和叠加后的被测程序模块:
float RightTriangle( int a, int b, int d)
{
    float f1, f2, f3, f4, F;
    f1 = f2 = f3 = MAX1; f4 = MAX2;
    f1 = max(0 - a, a - 200); /* 分支函数插装 */
    if (a > 0 && a < 200)
        {f2 = max(0 - b, b - 200); /* 分支函数插装 */
        if (b > 0 && b < 200)
            {f3 = maxhs(0 - d, d - 200); /* 分支函数插装 */
            if (d > 0 && d < 200)
                {f4 = fabs(a * a + b * b - d * d); /* 分支函数插装 */
                if (a * a + b * b == d * d)
                    printf("this is right angel. \n");
            }
        }
    }
    if (f1 < 0) f1 = 0;
    if (f2 < 0) f2 = 0;
    if (f3 < 0) f3 = 0;
    if (f4 == 0) f4 = 0;
    F = f1 + f2 + f3 + f4; /* 分支函数叠加得到目标函数 */
    return F;
}

```

在上例中不难看出,为了覆盖 printf 语句,其搜索目标就是使目标函数 $F = 0$ 的测试数据。

2 应用于测试数据生成的免疫遗传算法设计

遗传算法尽管是全局收敛的,但是由于两个主要的遗传算子(交叉、变异)都是在一定的概率下随机的、无指导的操作,因此在种群进化的同时也不可避免的出现了个体退化而导致盲目搜索。为了克服遗传算法盲目搜索以及收敛速度慢的缺点,在遗传算法中加入了免疫算子,免疫算子针对求解问题的特性先提取免疫疫苗,再对交叉、变异后的种群中部分个体注射免疫疫苗并进行相应的免疫检测,从而在一定程度上阻止个体退化,加快搜索速度,最后进行免疫选择。

2.1 免疫算子

2.1.1 提取疫苗

疫苗的提取是通过利用所求问题的一些特征信息或者对问题的一些先验知识来进行。疫苗不是一个成熟或完整的个体,它仅具备最佳个体局部基因位上的可能特征。本文中所给出的提取疫苗的方法是:对前 k 代中的两个不同的最优个体 x_1 和 x_2 的基因位上的共同特点信息作为疫苗。

2.1.2 接种疫苗

从当前种群中按一定比例选取 N_e 个个体,设个体为 $x = x_1x_2\cdots x_M$,给其接种疫苗是指按照先验知识,修改 x 的某些分量,使所得新个体以较大的概率适应最优解的取值范围,它须满足:若 x 已经是最优个体,则 x 以概率 1 转移为 x ;若 x 是最差个体,即 x 以概率为 0 转移为 x 。

2.1.3 免疫检测和选择

免疫检测就是对接种了疫苗的个体进行检测。若接种疫苗的个体的适应度不如父代,说明它出现严重的退化,则该个

体将被父代中相应的个体所替代,若其适应度优于父代,则该个体将进入下一代种群。它对加强接种疫苗有积极作用,避免退化。

在免疫选择中本文采用的个体选择概率为: $p = \alpha p_f + (1 - \alpha) p_d$, 其中 $0 < \alpha, p_f, p_d < 1, p_f$ 为适应度概率, p_d 为浓度概率, α 为亲和系数。它们的计算公式表示如下:

$$p_f = f / \sum_{i=1}^N f_i \quad (1)$$

其中, f 为个体适应度, N 为种群规模。在本文算法中, 个体适应度是对目标函数 F 进行评价: $f = \exp(-F/T_k)$, (其中, T_k 是当前 k 代的温度, 它是单调递减趋向于 0 的退火温度序列, 在本文中定义为 $T_k = T_0 \times 0.99^{k-1}$, T_0 为初始温度), 采用这样的适应度评价函数是通过温度的变化对适应度函数进行适当的拉伸。

在介绍浓度概率 p_d 的求解之前, 先介绍相关名词:

多样度 它是抗体的多样性测度。设有 N 个抗体, 每个抗体长度为 M , 采用的符号集大小为 S , 则抗体基因座 j 的信息熵 $H_j(N)$ 可定义为:

$$H_j(N) = \sum_{i=1}^S -p_{ij} \log p_{ij} \quad (2)$$

其中, p_{ij} 为第 i 个符号出现在基因座 j 上的概率, 可表示为:

$$p_{ij} = \frac{\text{在基因座 } j \text{ 上出现第 } i \text{ 个符号的总个数}}{N} \quad (3)$$

由信息熵可得平均熵 $H(N)$:

$$H(N) = \frac{1}{M} \sum_{j=1}^M H_j(N) \quad (4)$$

两个抗体 u 和 v 之间的亲和度定义为 $A_{u,v}$, 则 $A_{u,v} = \frac{1}{1 + H(2)}$, $A_{u,v}$ 取值范围为 $0 \sim 1$, $A_{u,v}$ 越大, 表示两个抗体越亲和或类似。

抗体浓度 C_i , 表示群体中相似抗体所占的比重, 即:

$$C_i = \frac{\text{和抗体 } i \text{ 具有很大亲和度的抗体数}}{\text{抗体总数 } N} \quad (5)$$

再由计算所得的抗体浓度, 找出浓度最大的个体 $1, 2, \dots, t$, 则定义这 t 个个体的浓度概率 p_d 为 $\frac{1}{N}(1 - \frac{t}{N})$, 其他 $N - t$ 个个体的浓度概率 p_d 为 $\frac{1}{N}(1 + \frac{t^2}{N^2 - N_t})$ 。

由上文的分析介绍, 采用 $p = \alpha p_f + (1 - \alpha) p_d$ 为个体选择概率有如下优点: 个体适应度越大, 则选择概率越大; 个体浓度越大, 则选择概率越小, 这样既可保留适应度高的个体, 又可确保个体多样性, 改善未成熟收敛。

2.2 免疫遗传算法执行步骤

1) 初始化: 随机生成初始种群。

2) 对当前种群 A_k 中的每个个体解码并进行评价, 从当前种群中提取疫苗, 并记录当前种群中的最优个体 x 。

3) 若当前种群 A_k 中已包含最优解或 k 已大于算法指定的遗传迭代代数, 则算法结束, 否则转 4)。

4) 对 A_k 种群按交叉概率 P_c 选择成对个体进行交叉, 得到新的种群 B_k 。

5) 对 B_k 种群按变异概率 P_m 选择个体进行变异, 得到新的种群 C_k 。

6) 在 C_k 种群中选取 N_e 个个体进行疫苗注射

7) 对注射疫苗的个体进行检测, 若发现个体退化, 则将父代中对应的个体引入得到种群 D_k 。然后计算 D_k 种群中个体的适应度、亲和度、浓度, 从而得出每个个体的适应度概率和

浓度概率,进而求出每个个体的选择概率,采用赌轮法得到新一代种群 A_{k+1} ,若发现 A_{k+1} 中最优个体没有2)中的个体 x 优秀,则将 A_{k+1} 中的最差个体由 x 来替代,转2)。

3 实验结果

选用第一节示例中的直角三角形判断程序做基本的测试用例生成,该示例目的是找到达到printf语句的测试用例。遗传算子中用到的初始量:由于该被测程序有三个输入参数,因此,遗传编码为3参数级联编码,编码长度共有36位,每个参数12位,每个输入参数范围设定为0~4095,编码精度为1。参数均匀杂交率为0.85,每个参数的变异率为0.1,种群规模为200,遗传最大迭代数为1000。免疫算子中用到的初始量:初始温度 T_0 为 $2e+5$,接受疫苗注射的个体数 N_e 为10,亲和系数 α 为0.95。

按照上面的初始量设置后,分别用遗传算法和免疫遗传算法做10次实验。实验结果如表1所示。

表1 实验结果

算法	迭代代数
传统遗传算法	1000代,无结果
免疫遗传算法	289代

对实验过程中生成的种群数据分析发现,传统遗传算法生成的测试用例易陷入局部最优解。而免疫遗传算法通过疫苗注射使得解能更快速收敛,并保持一定的多样性,在计算适应度时采用温度对目标函数值进行适当拉伸,从而加快收敛速度,并在计算个体选择概率时除了考虑通常的适应度概率

以外,又加入了浓度概率,这样不但可以保留适应度高的个体,又可以确保种群的个体多样性,有效的改善未成熟收敛。

4 结语

在测试数据的自动生成方法中,遗传算法虽然取得了较好的效果,但是这种算法存在缺陷和局限性,而免疫遗传算法改善了其不足之处。实验结果表明免疫遗传算法的测试用例生成效果高于遗传算法。

参考文献:

- [1] KOREL B. Automated software test data generation[J]. IEEE transactions on Software Engineering, 1990, 16(8): 870~879.
- [2] 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.
- [3] 葛红. 免疫算法与遗传算法比较[J]. 暨南大学学报: 自然科学版, 2003, 24(1): 22~25.
- [4] 韩学东, 洪炳熔, 孟伟. 基于疫苗自动获取与更新的免疫遗传算法[J]. 计算机研究与发展, 2005, 42(5): 740~745.
- [5] 肖人彬, 王磊. 人工免疫系统: 原理、模型、分析及展望[J]. 计算机学报, 2002, 25(12): 1281~1293.
- [6] 葛红, 毛宗源. 免疫算法的实现[J]. 计算机工程, 2003, 29(5): 62~63.
- [7] 刘芳, 梁雪峰. 一种基于免疫算子的SVM算法[J]. 计算机科学, 2004, 31(2): 109~110, 119.
- [8] 钟治平, 徐拾义. 程序插装技术在软件内建自测试中的应用[J]. 计算机工程与应用, 2004, 40(17): 117~118, 229.
- [9] 吕军, 冯博琴, 李波. 免疫遗传算法及其应用研究[J]. 微电子学与计算机, 2005, 22(6): 221~224.

(上接第722页)

结果和最差结果均优于SMES,其他4个函数两种算法的平均结果和最差结果均相等。

6 结语

通过对ODE算法的改进,引入一种简单的多样性规则,提出SIDE算法。由于采用了混合自适应交叉变异算子,增强了算法的非凸区域搜索和局部搜索的能力,加速了算法的收敛速度。同时,由于一种简单多样性规则的使用,允许不可行个体能在一定程度上保持在群体中,增加了算法处理带约束全局最优问题的能力。由于SIDE算法并没有采用罚函数法来处理约束函数,因此不需要所求解问题的任何先验知识,可以方便工程人员使用。通过13个标准测试函数的测试,表SIDE算法能够较好的解决带约束的全局最优化问题,具有收敛快速,求解精度高和鲁棒性等特点,其性能要优于所比较的算法。当然,SIDE算法在处理带多个等式约束的全局最优化问题时,还存在一定的不足,这是需要做进一步改进。

参考文献:

- [1] 潘正君, 康立山, 陈毓屏. 演化计算[M]. 北京: 清华大学出版社; 南宁: 广西科学技术出版社, 2000.
- [2] BÄCK T, SCHWEFEL H-P. An overview of evolutionary algorithms for parameter optimization[J]. Evolutionary Computation, 1993, 1(1): 1~23.
- [3] MICHALEWICZ Z. Evolutionary algorithms for constrained parameter optimization problems[J]. Evolutionary Computation, 1996, 4(1): 1~32.
- [4] MEZURA-MONTES E, CARLOS A, COELLO C. A simple evolution strategy to solve constrained optimization problems[J]. IEEE Transactions on Evolutionary Computation, 2005, 9(1): 1~17.
- [5] STORN R, PRICE K. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 11(4): 341~359.
- [6] COELLO C A C. Theoretical and numerical constraint handling techniques used with evolutionary algorithms: a survey of the state of the art[J]. Computer Methods in Applied Mechanics and Engineering, 2002, 191: 1245~1287.
- [7] SMITH A E, COIT D W. Constraint handling techniques-penalty functions[M]. [S. l.]: Oxford University Press, 1997.
- [8] RUNARSSON T P, YAO X. Stochastic ranking for constrained evolutionary optimization[J]. IEEE Transactions on Evolutionary Computation, 2000, 4(3): 284~294.
- [9] DEB K. An efficient constraint handling method for genetic algorithms[J]. Computer Methods in Applied Mechanics and Engineering, 2000, 186(2/4): 311~338.
- [10] MEZURA-MONTES E, COLLO C A C, TUN-MORALES I. Simple feasibility rules and differential evolution for constrained optimization [C]// Proceedings of the 3rd Mexican International Conference on Artificial Intelligence, LNCS 2972. Heidelberg: Springer-Verlag, 2004: 707~716.
- [11] GUO TAO, KANG LI-SHAN. A new evolutionary algorithm for function optimization[J]. Wuhan University Journal of Natural Sciences, 1999, 4(4): 409~414.
- [12] KOZIEL S, MICHALEWICZ Z. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization[J]. Evolutionary Computation, 1999, 7(1): 19~44.
- [13] HAMIDA S B, SCHOENAUER M. ASCHEA: New results using adaptive segregational constraint handling[C]// Proceedings of the 2002 Congress on Evolutionary Computation. Piscataway: IEEE Service Center, 2002, 1: 884~889.