基于遗传算法的业务流程测试

张 磊,王晓军
(南京邮电大学,江苏 南京 210003)

摘 要:随着软件在各个领域的广泛应用,人们对软件可靠性的要求不断提高。作为保障软件可靠性最重要的手段,软件测试所受到的关注也日益增加。在传统的黑盒功能性测试当中,单个模块的功能测试得到了很好的解决。然而在大型软件的集成测试时,由于软件所涉及的业务流程较多,模块较多,如何在黑盒功能性测试当中尽可能地完全覆盖所有的业务流程以及所对应的功能模块,常常需要软件测试人员人工定义。文中通过对遗传算法的研究,提出在集成测试时借助于遗传算法来产生测试用例,最大程度地覆盖所有的业务流程以及应用模块关联。将此前集成测试时,需要测试人员人工定义的大量业务流程以及模块关联转变为自动化产生,极大地提高了软件测试的自动化水平,提高了软件测试的效率。

关键词:软件测试;遗传算法;业务流程测试
中图分类号:TP311.56 **文献标识码**:A **文章编号**:1673-629X(2010)03-0155-04

Test of Business Process Based on Genetic Algorithm

ZHANG Lei, WANG Xiao-jun
(Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: Software has been widely used in various fields, demands of reliability of software become higher and higher. As the most important means of assurence of software reliablity, sofeware testing has got more and more attentions. The traditional black-box functional testing is a good solution of single module function testing. However, in the integration testing of large-scale software, there are so many processes and funciont modules, how to completely cover all the processes and the corresponding function modules as much as possible, often require software testing staff to define it. In this paper, based on the research of genetic algorithm, put forword a method that in integration testing using genetic algorithm to generate test cases, that to cover all the processes and the associated function modules in the greatest degree. In the integration testing with prior method, don't need test staff to manually define a large number of processes and associated modules but automatically generate test cases, which have greatly increased the level of automation in software testing and improved the efficiency of software testing.

Key words: software testing; genetic algorithm; test of business process

1 概 述

1.1 软件测试

软件测试是软件开发过程中的重要组成部分,通常占开发时间的30%到50%[1]。在测试活动中,测试案例的编写是一项极具挑战性的工作。研究测试案例的自动生成,提高测试的质量和测试的效率,具有十分重要的意义。

传统的测试方法主要关注于白盒测试的源代码本身,但是由于现今软件的结构复杂性以及软件规模的不断增大,使人们缺少一种有效的对于业务流程的测试案例自动生成的方法。自动化业务流程测试的方法是通过随机选取业务流程模块抽取必要的工作流,尽可能多地覆盖此软件的所有业务流程路径,执行所选取的业务流程,尽可能发现更多的错误。

黑盒测试[2](Black-box testing)是通过使用整个软件或某种软件功能来严格地测试,而并没有通过检查程序的源代码或者很清楚地了解该软件或某种软件功能的源代码程序具体是怎样设计的。测试人员通过输入数据然后看输出的结果从而了解软件怎样工作。通常测试人员在进行测试时不仅使用肯定出正确结果的输入数据,而且还会使用有挑战性的输入数据以及可能结果会出错的输入数据以便了解软件怎样处理各种类型的数据。黑盒测试的测试用例设计通常可用等价类划分法。所谓等价分类(Equivalence Partition-

收稿日期:2009-06-23;**修回日期**:2009-09-05
基金项目:国家科技支撑计划(2007BAH17B04)
作者简介:张 磊(1983-),男,硕士研究生,研究方向为计算机网络与分布式计算;王晓军,硕士研究生导师,研究方向为面向服务的软件体系结构、业务流程管理和协同,特别是对Web服务以及基于Web服务的应用支撑环境的研究等。

ing),是指把所有可能的输入数据(有效的和无效的)划分成若干个等价的子集(称为等价类),使得每个子集中的一个典型值在测试中的作用与这一子集中所有其它值的作用相同。因此可从每个子集中选取一组数据来测试程序。换句话说,如果从某一等价类中任意选出一个测试用例未能发现程序的错误,就可以合理地认为在该类中的其它测试用例也不会发现程序的错误。这样,就把漫无边际的随机测试变成有针对性的等价类测试,有可能用少量有代表性的例子来代替大量内容相似的测试,借以实现测试的经济性。

1.2 遗传算法简介

遗传算法[3](Genetic Algorithm)是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型,是一种通过模拟自然进化过程搜索最优解的方法,它最初是由美国 Michigan 大学 J. Holland 教授于1975年首先提出来的,并出版了颇有影响的专著《Adaptation in Natural and Artificial Systems》,GA 这个名称才逐渐为人所知。GA 通常包含 5 个基本要素[4]:①参数编码;②初始种群设定;③适应度函数的设定;④遗传操作设计;⑤控制参数设定。

GA 解决问题的一般流程是:①随机产生一定数目的初始种群,每个个体表示为染色体的基因编码;②计算每个个体的适应度,并判断是否符合优化准则,若符合,输出最佳个体及其代表的最优解并结束计算,否则转向第3步;③依据适应度选择再生个体,适应度高的个体被选中的概率高,适应度低的个体可能被淘汰;④执行交叉和变异操作,生成新的个体;⑤得到新一代的种群,返回到第②步。

遗传算法运算过程:

1)选择(复制):根据各个个体的适应度,按照一定的规则或方法,从第 t 代群体 P(t)中选择出一些优良的个体遗传到下一代群体 P(t+1)中;

2)交叉:将群体 P(t)内的各个个体随机搭配成对,对每一对个体,以某个概率(称为交叉概率)交换它们之间的部分染色体;

3)变异:对群体 P(t)中的每一个个体,以某一概率(称为变异概率)改变某一个或某一些基因座上的基因值为其他基因值。

遗传算法有下列显著特点:

①遗传算法从问题解的中集开始搜索,而不是从单个解开始;②遗传算法求解时使用特定问题的信息极少,容易形成通用算法程序;③遗传算法有极强的容错能力;④遗传算法中的选择、交叉和变异都是随机操作,而不是确定的精确规则;⑤遗传算法具有隐含的并行性,显著提高了搜索效率。

遗传算法和传统的搜索寻优算法相比有以下几点不同:

(1)遗传算法工作在参数集的编码上而非参数集本身上。遗传算法是一类与领域无关的通用搜索算法,属于弱搜索算法。

(2)遗传算法从一个点的群体而不是从一个单一点进行搜索。遗传算法搜索保持一个潜在解的群体,而许多传统的方法处理搜索空间的一个单点。

(3)遗传算法采用概率转移规则去引导搜索,也就是采用概率随机选择做为工具引导朝向有可能改进的区域进行搜索。

(4)遗传算法用目标或评价函数信息,无需其他辅助知识,这使得遗传算法成为更为规范的方法。

遗传算法作为人工智能领域的后起之秀,近年来已崭露头角,作为一种强健的搜索方法,它在解决大空间、多峰、非线性、全局优化等高复杂度问题时显示了独特的优势和高效性。遗传算法的引入,也为解决测试流程自动生成这一软件测试领域的难题带来了新的思路[5]。

2 遗传算法在业务流程中的应用

使用遗传算法实现测试用例的自动生成可描述为:应用遗传算法来求解一组优化的测试用例,在每一步进化计算过程中,测试用例自动生成器使用当前群体(测试用例)驱动被测试程序的执行,每一个测试用例的执行路径被跟踪和记录,以最大化程序执行路径的覆盖为适应性目标函数进行计算,产生出下一代群体。在多代进化之后,得到最优种群或超过特定的循环限制条件而结束[6]。

用遗传算法生成测试用例的系统模型基于遗传算法软件测试用例生成的系统模型如图1所示。

该模型可分为两部分:算法执行部分和算法评价部分。算法执行部分是系统的核心,它随机地产生第一代种群,然后按照输入参数的编码方式将种群中的个体位串映射成实际参数值,并传递给被测程序,驱动被测程序的运行。算法评价部分主要利用程序插装技术,在被测程序源代码级插入用以评估当前输入参数值的评价函数,并将评价函数值返回给遗传算法。算法执行部分据此来评价种群中每个个体位串的优劣,并通过遗传算子(选择、杂交、突变)的操作改变个体位串的结构,形成新一代更优种群,如此往复,直至找到覆盖选定路径的目标参数值。

2.1 编 码

由于遗传算法是工作在参数集的编码上,而非参数集本身上,所以我们要对我们的业务流程进行编码。

编码方法在很大程度上决定了如何进行群体的遗传进化运算以及遗传进化运算的效率。因此 Goldberg 提出了三条编码评估规范[7]:

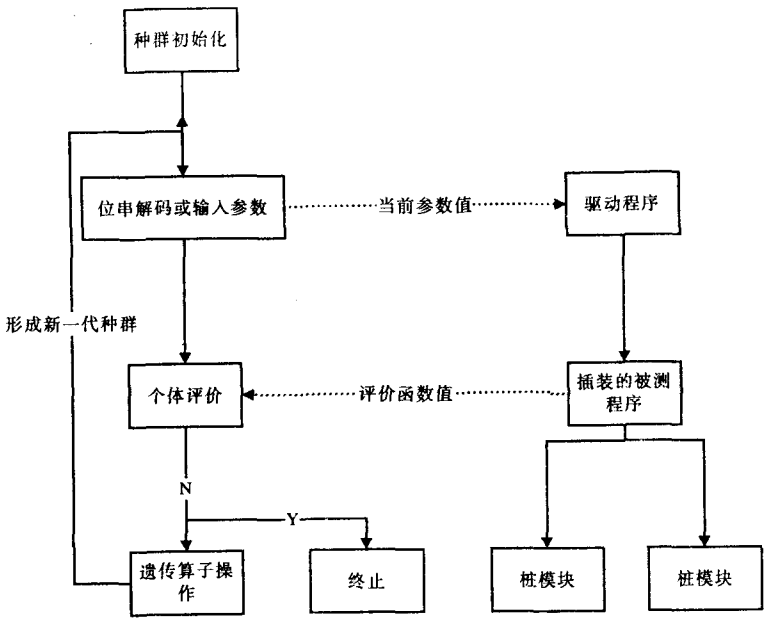

图 1　测试用例生成系统模型图

(1)完备性:问题空间中的所有点都能做为 GA 问题空间中的染色体表现。

(2)健全性:GA 空间中的染色体能对应所有问题空间中的候选解。

(3)非冗余性:染色体和候选解一一对应。在此首先根据软件的需求抽象出所要执行的业务流程模块的信息流图表,结点表示中间状态,边表示走向的下一流程。然后对所有的业务流程指向依次进行字符标示如图 2 所示。

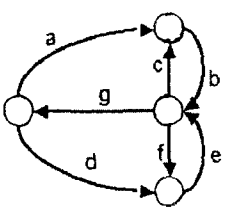

图 2　业务流程信息流图

其次,要将此图表转化为另外一种可方便编码的图表,具体步骤如下:对于任意一结点,将该结点的入度边转化为一个新图的一个结点指向该结点的出度边结点。例如对于最左边结点,该结点的入度为 g 边,出度为 a、d 条两条边,转化以后如图 3 所示。

对图 2 进行这种完全的转化为如图 4 所示。

然后对所产生的新图得出此图的邻接矩阵,两结点直接相连接的表示为 1,不直接相连接的表示为 0,可得矩阵如图 5 所示。

取此矩阵主对角线以下的每行的代码对图中顶点进行编码为(1 01 000 0011 01001 11011)

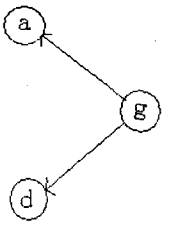

图 3　最左结点的转化图

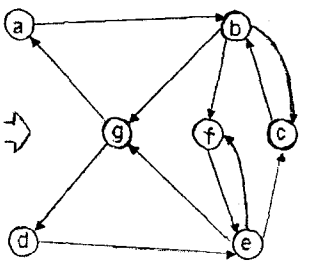

图 4　业务流程信息图完全转化图

	a	b	c	d	e	f	g
a	0	1	0	0	0	0	1
b	1	0	1	0	0	1	1
c	0	1	0	0	1	0	0
d	0	0	0	0	1	0	1
e	0	0	1	1	0	1	1
f	0	1	0	0	1	0	0
g	1	1	0	1	1	0	0

图 5　转化后的邻接矩阵

2.2　适应度函数的选择

如何定义适应度函数,是遗传算法的关键步骤,函数的优劣将直接影响到解决问题的效率。在此由于是对业务的流程进行测试,为了更大程度地对业务流程进行覆盖,在对流程中的各模块进行编码之后,选取每个个体中的"1"的个数在 N 段中位数总和的百分比作为整个的适应度函数。适应度越大表示这个个体越好,根据适应度的大小顺序对群体中的个体进行排序。显然适应度越大,表示业务流程间的工作交互与联系越多,随机进行个体选择之后,适应度大的选中的概率就大,然后去替换适应度小的个体,适应度高的个体一直向下保存。这样就保证了主要的业务流程的路径覆盖程度的增大,防止在测试中丢失一些业务模块。

2.3　交叉和变异

交叉:遗传算法包含两个主要的操作:交叉和变异[8]。所谓交叉算子也称重组、配对算子。交叉是产生新个体的主要手段,在遗传算法中起核心作用。它仿照生物学中杂交的原理,将两个父代染色体的部分基因相互交换,由此产生新的个体。交叉操作有三种主要的方法:单点交叉,双点交叉,均匀交叉。单点交

叉的步骤是在群体中随机选择两个染色体，然后随机再选择这两个染色体的某一位置，二者互换从该位置起的末尾部分基因，因此长度为N的码串，可以有的交叉点是N-1个，单点交叉可以实现N-1个不同的结果。由于单点交叉的交叉点是随机的，当参数个数增多，位串长度加长时，仅使用单点交叉技术，有可能会造成搜索概率的降低。在此我们对单点交叉技术进行改进：在随机得到交叉点时，也随机得到一个从编码起始点到交叉点长度的一个随机屏蔽字Pi，在交叉点之后位串依然是尾部互换，在两个染色体交叉点之前的两个等长的位串各位进行比较，如果两染色体某位编码相同，则与相应位的屏蔽字相交换，依次操作直到交叉点处。在我们的测试方法中这样做的目的在于使得种群的多样化，因为是对业务流程进行测试，要尽可能多地对业务的流程路径进行最大可能的覆盖。交叉操作的随机概率选择也有着重要的作用，如果概率较大，交叉的能力越强，但是对优秀种群的破坏也越大。交叉操作的概率越小，搜索可能陷入停滞状态，所以一般建议选择Pc是在0.5～0.99之间。

变异：变异是按一定的变异概率Pb将位串的某一位或某几位的1变成0，将0变成1。由于我们的编码属于是动态型编码，所以变异概率也采用一种动态确定变异率的方法如下：先将软件的业务流程图画出，将整个软件的所有业务流程数(对应于流程图的边)设为A，将业务流程模块(对应于流程图的结点)设为B，变异率=A/(A+B)；

流程图中如果边的个数较多，则说明流程越复杂和难以完全覆盖，采用此变异率可以在流程过多时显著地提高位串上发生突变的机会，这种动态的突变率的确定，可以比较好地针对于我们的业务流程测试系统的尽可能完全覆盖的目的以及动态编码的特性。

3 结束语

文中主要介绍了遗传算法在大型软件的业务流程方面的测试，采用了一种基于路径覆盖的方法，尽可能地生成所有的业务流程路径，动态地产生和更新工作流。遗传算法所具有的传统方法不具备的强壮性和高效性等特点，使得在解决此问题时显示了其独有的优势。

经过实验证明，采用此改进的遗传算法对业务流程进行测试时，可以使我们在集成测试时，尽最大可能地自动生成和选取各个模块之间的所有的组合流程，不再需要测试人员进行人工的记录以及大量繁琐的工作。但是也存在着一定的问题，就是在整个的软件系统中，不是每个模块和每个模块都有一种直接的联系和信息交互的，当面临不同模块的间接交互或依赖关系时，如果对算法进行改进？为了保证自动化产生的业务流程为有效流程，在具体的实际操作中可能还有一些细节需要进行考虑和研究，期望通过此文与有关方向的专家学者共同的探讨，力图使此方法尽快地实用化。

参考文献：

[1] 许向阳.遗传算法与结构自动测试[J].计算机工程与应用,1998(4):34-36.

[2] 朱良学.基于遗传算法的软件测试用例研究[J].软件导刊,2008,7(5):37-38.

[3] 姚 尧.一种基于遗传算法的软件测试用例生成新方法[J].计算机与数字工程,2009,37(1):18-21.

[4] 易云辉,汪国六.基于遗传算法的软件测试用例生成系统模型[J].科技广场,2005(6):18-20.

[5] 荚 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.

[6] 毛 颖,罗 蕾.基于遗传算法和爬山算法的测试用例生成研究[C]//2006年全国第六届嵌入式系统学术年会.西安:计算机技术与发展编辑部,2006.

[7] Rajappa D V,Biradar A,Panda S. Efficient Software Test Case Generation Using Genetic Algorithm Based Graph Theory [C]//First International Conference on Emerging Trends in Engineering and Technology. [s. l.]:[s. n.],2008:298-303.

[8] 赵 明,张毅坤,沈建雄,等.基于遗传算法的测试用例生成工具的研究[J].计算机工程,2005,31:13-15.

(上接第124页)

Journal of Human-Computer Studies,1998,48:125-141.

[3] Oliver J R. A machine learning approach to automated negotiation and prospects for electronic commerce[J]. Journal of Management Information Systems, 1996,13(3):83-112.

[4] 牛晓太,王 洋,邓其军.一种基于GA的自动谈判问题解决方案[J].计算机工程与应用,2005,41(24):197-200.

[5] 韩 伟.基于模糊约束规划的自动协商[J].计算机工程与应用,2008,44(3):94-97.

[6] 李海晨,冯玉强.面向Agent的供应链谈判模型研究[J].哈尔滨工业大学学报,2007,39(8):1305-1308.

[7] FIPA Communicative Act Library Specifi-cation. Foundation for Intelligent Physical Agents[EB/OL]. 2003. http://www.fipa.org/specs/fipa00037/index.html.

[8] 邓方安,周 涛,徐 扬.软计算方法[M].北京:科学出版社,2008:153-154.