

测试数据生成技术在软件故障诊断中的应用

王 杰,周 婕,慕晓冬

(第二炮兵工程学院,陕西 西安 710025)

摘 要:随着软件在武器装备中应用范围的日益扩大,软件故障诊断问题日益突出。文中针对软件测试中测试数据生成问题作了重点研究,采用了遗传算法解决按路径生成测试数据的问题,构建了基于遗传算法的测试数据生成系统模型,并对系统模型进行了实例验证。通过数据证明了遗传算法在测试数据生成中的优越性。

关键词:遗传算法;软件测试;测试数据;分支函数

中图分类号:TP311.52

文献标识码:A

文章编号:1673-629X(2006)09-0091-02

Application of Testing Data Technique in Software Fault Diagnosis

WANG Jie, ZHOU Jie, MU Xiao-dong

(The Second Artillery Engineering Institute, Xi'an 710025, China)

Abstract: According to development of software in the area of weaponry, the diagnosis of software failure becomes more and more important. The paper focuses on the generation of software testing, and uses the method of genetic algorithm to build a model of software testing data generation. From the experimentation for some testing models, it testifies that the genetic algorithm is certainly a good method for software testing.

Key words: genetic algorithm; software testing; testing data; branch function

0 引言

随着计算机技术在武器系统和自动化信息系统中的广泛应用,软硬件的结合越来越紧密,大量软件装备在武器系统中的应用使软件规模越来越大,实现的功能也越来越复杂。作为保证软件质量和可靠性的重要手段,软件测试在这方面发挥着其它方法不可替代的作用。而软件测试技术中关键的一项就是如何产生合适的测试数据。目前,测试数据的自动生成技术是研究的热门问题^[1]。

1 系统模型的构建

软件测试数据生成通常可分为功能测试数据生成和结构测试数据生成。文中将只就结构测试数据生成技术进行讨论。对软件结构测试而言,一般采用基于“路径覆盖”的方式来生成测试数据。目前,按路径生成测试数据的方法主要有符号执行和程序直接执行技术,而程序执行技术中的难点在于选取合适的搜索寻优算法。

遗传算法作为一种基于自然选择原理和自然遗传机制的通用搜索算法,基于遗传算法生成测试数据系统模型中的核心部分是遗传算法包,首先随机地产生原始种群,

然后按照输入参数的编码方式将种群中的个体位串解码成实际参数值,并传递给被测程序,驱动被测程序的运行。被测程序运行时计算出的评价函数值返回给遗传算法包,算法包据此来评价种群中每个个体位串的优劣,并通过遗传算子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 软件测试数据的生成

遗传算法自身的寻优特点,决定了它能在较短时间内得到近似最优解,因此很适合应用于面向路径的测试数据生成中^[2]。图1所示为基于遗传算法测试数据生成的系统模型。

2.1 参数的编码

针对程序单元的结构测试,被测单元一般有一个至多个数据类型相同或不同的参数,而遗传算法在求解问题时,要求把问题的解空间表示为有限常数的位串。因此,必须首先将被测单元的一系列参数进行编码。因此用遗传算法生成测试数据时,要寻求适当的映射方式 f ,使得被测程序的输入参数 $\{x_1, x_2, \dots, x_n\}$ 能够表示成一种适当的编码形式: $f(x_1, x_2, \dots, x_n) \rightarrow (c_1, c_2, \dots, c_n)$;遗传算法在运行时,遗传算子 g 对参数编码进行操作,改变其结构,形成新的编码: $g(c_1, c_2, \dots, c_n) \rightarrow (c_1', c_2', \dots, c_n')$; f 逆映射 f^{-1} 解码得到输入参数的当前值 $\{x_1', x_2', \dots, x_n'\}$ 即: $f^{-1}(c_1', c_2', \dots, c_n') \rightarrow (x_1', x_2', \dots, x_n')$ 。如果

收稿日期:2005-11-17

基金项目:中国人民解放军总装备部十五预研基金项目(413270204)

作者简介:王 杰(1982-),男,山东滕州人,硕士研究生,研究方向为软件工程、网络信息安全等;慕晓冬,教授,博士生导师,研究方向为多媒体技术、软件工程、信息融合等。

当前参数值 $\{x_1', x_2', \dots, x_n'\}$ 不满足终止条件, 遗传算子 g 将重复上述过程, 直到找到目标参数值。

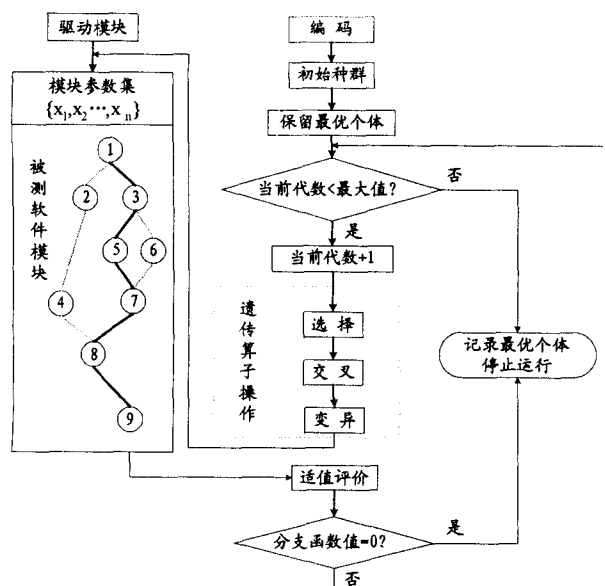


图 1 基于遗传算法测试数据生成模型

2.2 遗传算子的改进

遗传算法通过其遗传算子的操作来不断产生新子代, 再从新子代中选择优化的子代, 从而实现种群的优化。其中的遗传算子操作主要有交叉算子和突变算子等, 文中以交叉算子为例, 通过改进遗传算子从而提高整个种群的进化效率。

简单的单点交叉是在种群里随机选择两个参数将其编码成位串形式, 并在个体位串上随机选取的交叉点, 将两个位串交叉点后的部分交换从而形成新的位串, 即为新的个体。但是这种单点交叉在参数较多、位串较长的个体中显得力不从心了。因为仅仅是单点交叉使得个体的变异不明显, 不利于新信息的引入, 从而降低了搜索效率。因此, 考虑引入多点交叉来解决以上的问题。但是仅是通过增加交叉点无法解决随机插入的交叉点的分布情况, 就可能出现交叉点分布过于集中的情况, 这种状况下对于整个种群的进化也是不利的, 因为这样的变异仅仅局限于几个参数的位串上, 导致种群的进化产生较大的局限性。为此有意识地将交叉点均匀地分布在参数上, 使得每个参数实现单点交叉。这种对交叉算子的改进, 既避免了某些参数变异过大, 也解决了位串信息变化较小的弊端, 从而充分发挥了交叉算子的变异效率。图 2 所示为改进后的遗传算子结构。

2.3 适值函数的构造

适值函数是遗传算法与实际问题的惟一接口, 因此适值函数的构造同实际问题结合紧密。在 Korel“分支函数”概念的基础上, 提出了“分支函数正相叠加法”^[3]。分支函数是一个实值函数, 是分支谓词到实值的一个映射, 可以量化地反映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。在此基础上, 采用了 J. C. Huang 提出的程序插装理论进行分支函数插装^[4,5]。具

体做法是在选定路径上各分支点前插入相应的分支函数 (假定选定路径上有 m 个分支点, 参与编码的“参数”个数为 n 个):

$$\varphi_1 = f_1(x_1, x_2, \dots, x_n), \varphi_2 = f_2(x_1, x_2, \dots, x_n), \dots, \varphi_m = f_m(x_1, x_2, \dots, x_n)$$

最后在待测单元结束处插入适值函数:

$$F = \Psi(\varphi_1) + \Psi(\varphi_2) + \dots + \Psi(\varphi_m)$$

$$\Psi(x) = \begin{cases} 0 & x < 0 \text{ 或 } x = 0 \\ x & x > 0 \end{cases}$$

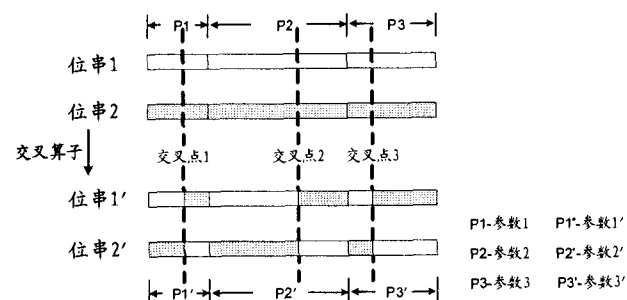


图 2 增加一均分交叉点

2.4 算法在单元测试中的应用

下面举例说明为寻求覆盖测试单元中所有真分支的测试数据而进行的分支函数和适值函数的插装:

Procedure TEST_Unit(I: in out INTEGER; F: in FLOAT) is

Fit: long_float := Fit_initial; constant long_float := 1.0E+3;

f1: long_float := Fit_initial;

f2: long_float := Fit_initial;

f3: long_float := Fit_initial;

O: INTEGER;

package INT_IO is new INTEGER_IO(INTEGER);

use INT_IO;

begin

null; \ 输出语句“NULL”

O := TCAG_VAR_PACK.IO_VAR01; \ 输出语句重定向

I := 1;

f1 := long_float(max((O) - (o) + 1, (o) - (2) + 1)); \ 分支函数插装

if O > 0 and O < 2 then

f2 := long_float(max((3.0) - (f) + float SAFE_SMALL, (f) - (4.0) + float SAFE_SMALL)); \ 分支函数插装

if F > 3.0 and F < 4.0 then

f3 := long_float(max((O) - (e) + 1, (e) - (50) + 1)); \ 分支函数插装

if E > 0 and E < 50 then

null; \ 输出语句“NULL”

end if;

end if;

end if;

if f1 < 0.0 then f1 := 0.0; end if;

if f2 < 0.0 then f2 := 0.0; end if;

if f3 < 0.0 then f3 := 0.0; end if;

(下转第 96 页)

管理的测试框架,如文献[7]。这些都满足了软件易测试性分析技术的要求,也将为改进和提高软件测试的过程乃至软件开发的过程提供帮助。此外,测试数据的生成也是 Web 服务测试的重要内容,文献[8]实现了使用 TTCN-3 描述的测试数据的自动生成,文献[9]则是基于合约变异的技术,实现一种有效测试数据的自动生成方法。文献[10]对 Web 服务测试报告的管理方法进行了研究,提出了一种实现方法和框架。

6 结束语

由于 Web 服务是一种包含大量运行行为的分布式应用,因而不可能完全沿用传统软件测试技术对其进行测试。文中基于 Web 服务测试的一些新特点,进行了一些比较和归纳,并给出了一种测试的框架,说明整个测试的过程。

目前,随着 SOA 的不断成熟,Web 服务处理自动化工具变得流行起来,Web 服务在运行的时候能够被动态描述。以前 Web 服务使用者只能动态地发现服务并使用它们,但是现在这些服务在被使用时候可能时刻都在变化。一个服务提供者只提供其他使用者一些类似产品目录的服务,这种 Web 服务实际上是大量的、动态的其他 Web 服务的集合。这时需要一种标准语言来描述不同的 Web 服务是如何集成在一起的,即描述 Web 服务流的语言。当前有两种常见的 Web 服务流描述语言,即 WSFL 和 XLANG。如果让用户自己来描述 Web 服务流,很可能导致错误。在发现错误之前,流中的许多 Web 服务已经被调用了,其后果会导致事务回滚困难、引起网络拥塞,从而造成资源浪费。文献[11]和[12]对 Web 服务流进行了研究。因此在以后的研究工作中,更多研究将是 Web 服务组的测试、Web 服务版本测试。

参考文献:

[1] W3C note. SOAP(Simple object access protocol)[EB/OL].

(上接第 92 页)

TCAG. VAR. PACK. global. fit; = f1 + f2 + f3; \ 将适值评价函数返回给遗传算法包

end TEST. Unit;

通过以上的实例可以看出,运用“分支函数正相叠加法”生成的适值评价函数既能反映测试数据的优劣,又能引导遗传算法最终找到目标参数值。从而对于寻找覆盖测试路径的最优值起到极大的促进作用,也使得面向路径测试数据的生成更加方便、可靠。

3 结论

着重分析了面向路径软件测试中测试数据生成问题,遗传算法凭借其优秀的强壮性和高效性,在解决这一问题上体现了其独具的优势。从整个系统模型的构建到最后

http://www.w3.org/TR/SOAP/,2003-06.

- [2] W3C note. WSDL (Web Service Description language)[EB/OL].http://www.w3.org/TR/wsdl.html,2001-03.
- [3] OASIS. Universal Description, Discovery, and Integration[EB/OL].http://www.uddi.org,2005-02.
- [4] Sprott D, Wilkes L. Understanding Service-Oriented Architecture[EB/OL].http://msdn.microsoft.com/architecture/soa/default.aspx,2004-02.
- [5] Nguyen H Q, Johnson B, Hackett M. Web 应用测试: test planning for mobile and internet-based systems[M].北京:电子工业出版社,2005.183-191.
- [6] Tsai W T, Paul R, Wang Y, et al. Extending WSDL to Facilitate Web Services Testing[A]. Proceedings of the 7th IEEE International Symposium on High Assurance Systems Engineering[C]. Tokyo:[s. n.],2002.171-172.
- [7] Tsai W T, Paul R, Song W, et al. Coyote: An XML-based Framework for Web Services Testing[A]. Proceedings of the 7th International Symposium on High Assurance Systems Engineering[C]. Tokyo:[s. n.],2002.173-174.
- [8] Schieferdecker I, Stepien B. Automated Testing of XML/SOAP based Web Services[A]. Proceedings of the GIFachtagung "Kommunikation in Verteilten Systemen", KIVS[C]. Leipzig, Germany:[s. n.],2003.43-54.
- [9] 姜 瑛,辛国茂,单锦辉,等.一种 Web 服务的测试数据的自动生成方法[J].计算机学报,2005,28(4):568-577.
- [10] Luo ling, BAI Xiaoying. Web services-Based Test Report Generation[J]. Tsinghua science and technology,2005,10(3):282-287.
- [11] Nakajima S. On Verifying Web Service Flows[A]. Proceedings of the Symposium on Applications and the Internet Workshops[C]. Nara:[s. n.],2002.223-224.
- [12] Nakajima S. Verification of Web Service Flows with Model2 Checking Techniques[A]. Proceedings of the 1st International Symposium on Cyber Worlds[C]. Tokyo:[s. n.],2002.378-385.

算法的应用都很好证明了该方法的实用性和有效性。

参考文献:

- [1] 郑人杰. 计算机软件测试技术[M].北京:清华大学出版社,1992.
- [2] 英 伟,高仲仪.基于遗传算法的软件结构测试数据生成技术研究[J].北京航空航天大学学报,1997,23(1):36-40.
- [3] 傅 博.基于模拟退火遗传算法的软件测试数据自动生成[J].计算机工程与应用,2005(12):81-83.
- [4] Korel B. Automated software test data generator[J]. IEEE Trans on Software Eng,1990,16(8):870-879.
- [5] Huang J C. Detection of Data Flow Anomaly Through Program Instrumentation[J]. TSE,1979,5(3):226-236.