

软件测试数据智能化生成的研究

傅博

(北京航空航天大学工程系统工程系, 北京 100083)

摘要: 软件测试数据自动生成是软件测试中的重要难题之一。测试数据自动生成问题可归结为测试数据的搜索或组合优化问题, 通常具有不连续、不可微和非线性等特征, 适合于采用遗传算法、神经网络等人工智能技术进行解决。国内外学者在此方面作了不少研究并取得一定的成果, 但也存在一些问题。该文系统地综述了近年来软件测试数据智能化生成的研究和存在的问题, 并对未来的发展进行了展望。
关键词: 软件测试; 神经网络; 遗传算法; 测试数据自动生成

Research on Software Intelligent Test Data Generation

FU Bo

(Dept. of System Engineering of Engineering Technology, Beijing University of Aeronautics and Astronautics, Beijing 100083)

【Abstract】The automatic software test data generation is one of the elementary problems in software testing. The problem of test data generation is reduced to a search or combinatorial optimization problem with discontinuity and nonlinearity. Therefore, it can be suitably resolved by using AI techniques such as neural networks and genetic algorithms. This paper summarizes and analyzes the application research of AI techniques on automatic test data generation. Future research directions and some of the researching problems are also addressed.

【Key words】 Software test; Neural networks; Genetic algorithms; Automatic test data generation

在软件开发过程中, 软件测试占有举足轻重的地位。为了缩短开发周期, 降低费用, 人们研究软件测试数据的自动生成问题。由于此问题极其复杂, 通常属于组合优化问题, 用常规方法解决遇到困难, 因此软件测试数据的智能化生成问题成为软件测试的研究热点。

1 基于GA的测试数据自动生成

基于遗传算法(GA)的测试数据自动生成可归结于测试数据的优化问题, 基本原理如图1所示。在被测软件(SUT)的输入域内, 以规定的测试准则, 采用GA搜索测试数据。按照测试准则, 它分为结构和功能两类。

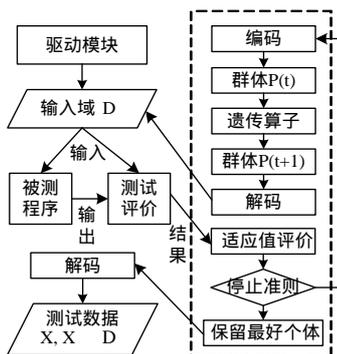


图1 基于GA的测试数据自动生成

1.1 基于GA的结构测试数据自动生成

生成原理是在输入空间中, 通过维持一定规模的输入数据群体, 按照选定的路径不断演化群体, 从而找到符合该路径的输入数据。GA的目标是搜索满足规定的结构覆盖率的测试数据集。

(1) 编码规则

GA生成测试数据的首要问题是通过编码规则将输入变量映射

成GA的染色体, 以便进行遗传操作。对输入空间D内某一输入变量 $\{x_1, x_2, \dots, x_N\}$, 通过编码规则C转换为位串形式的染色体 y_i , $C: \{x_1, x_2, \dots, x_N\} \rightarrow y_i$; 经GA操作G实现染色体的演化, $G: y_i \rightarrow y_i'$; 再通过解码 C^{-1} 转换为输入变量, $C^{-1}: y_i' \rightarrow \{x_1', x_2', \dots, x_N'\}$, 并输入到SUT。二进制编码规则具有表达简单、实施快捷等优点^[1], 文献[2]对浮点数变量实施二进制编码, 对其它数据结构实施格雷编码。

(2) 适应值函数

适应值函数设计依赖于程序路径的分支谓词, 采用程序插装技术, 按照分支谓词的特点插入分支判断函数。有两种方法确定分支判断函数: Korel分支函数法和海明距离法。

Korel分支函数法: 设分支谓词pr为关系表达式: $E_1 OP E_2$, E_1 和 E_2 为算术表达式, 关系运算符OP $\{<, >, =, \dots\}$ 。将pr转换为等价的F rel 0谓词形式, $rel \{<, =, \dots\}$, F即为Korel分支函数。该方法具有简单易行、最优值已知等优点, 应用较广^[2]。

海明距离法: 将输入变量 $\{x_1, x_2, \dots, x_N\}$ 映射为一对谓词值 $\{x_{pk}, x_{pk}'\}$, 分别对应谓词k两边的函数值。海明距离算子H将谓词值对 $\{x_{pk}, x_{pk}'\}$ 对应的位串对 $\{b_{pk}, b_{pk}'\}$ 映射为数值, 即为海明距离函数, $H: \{b_{pk}, b_{pk}'\} \rightarrow f_k$ 。该方法能够部分检查分支谓词边界, 代价是适应值函数的最优值为未知^[1]。

(3) 遗传算子的改进

交叉算子: 当输入变量较多、个体位串较长时, 单点交叉对位串结构的改变不大, 不利于新信息的引入, 导致搜索效率降低, 所以采用多点交叉。在多点交叉时, 考虑交叉均匀性。

变异算子: 针对输入变量个数和位串长度, 实现变异均匀性。在编码规则的位串组合方式中, 引入位权重系数, 控制位置变异。

(4) 模拟退火GA

GA生成结构测试数据虽然能够快速逼近目标值, 但从逼近目标值到最终找到目标值花费的代价甚至比逼近的过程还要大。为解决

作者简介: 傅博(1964—), 男, 高工、博士生, 主研方向: 软件测试, 软件可靠性, 人工智能应用

收稿日期: 2005-09-18 **E-mail:** fu_bo@371.net

此问题,克服 GA 局部搜索能力弱的缺点,可将模拟退火作为一种遗传算子嵌入到 GA 中,根据 GA 进化成熟度对个体进行模拟退火。

1.2 基于 GA 的功能测试数据自动生成

在软件输入域内,GA 搜索功能错误或功能极限的测试数据。

(1)功能错误的适应值函数

相对适应值组合函数:考虑过去的测试数据因素,由新颖度、相似度和严重度构成^[3]。新颖度指测试数据与过去测试数据相比的相异性。相似度指测试数据与发现SUT错误的测试数据的相似程度。严重度指发现错误的测试数据导致SUT故障的严重程度。

绝对适应值组合函数:不考虑过去的测试数据因素,由发生率、故障密度和故障严重度构成。对于命令(逻辑)序列和数量序列的测试数据,适应值函数由逻辑错误和数量错误组成。

(2) 功能极限的适应值函数

对于实时软件,响应时间是一项重要功能,搜索边界响应时间(WCET)以判定实时功能的正确性是功能测试的任务之一。采用 GA 生成 WCET 的测试数据显示出较大的优势,适应值函数可以采用响应时间设计。

2 基于神经网络的测试数据生成

2.1 预测测试数据的揭错能力

人工神经网络(ANN)作为分类器预测测试数据原理如图 2 所示。通过ANN预测得到测试数据的揭错能力选用测试数据,以减少测试数据数量^[4]。

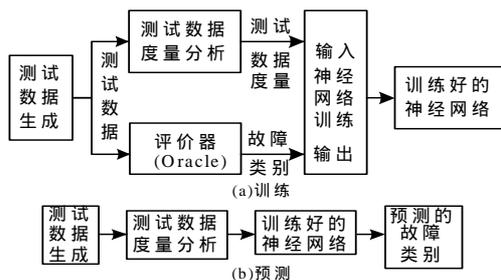


图 2 ANN 预测测试数据的揭错能力

在训练阶段,测试数据生成器生成的测试数据一方面通过度量分析提取测试数据指数,另一方面通过评价器(Oracle)得到测试数据揭示的故障类别。由一系列测试数据指数及对应的故障类别组成 ANN 的学习样本,采用学习算法(如 BP 算法)实施 ANN 训练,训练完毕得到训练好的 ANN。在预测阶段,由 Oracle 生成测试数据,经过度量分析提取测试数据指数,将其输入到训练好的 ANN,得到测试数据的揭错等级,并据此对测试数据进行取舍处理。训练阶段是对 Oracle 的学习概括,预测阶段是对 Oracle 的推广。

Oracle 是 ANN 预测测试数据的关键。Oracle 能正确判断一个测试数据是否导致 SUT 出现故障,若出现故障属于何种类别的故障。Oracle 可以通过软件故障模式分析,构建测试数据故障模型库实现。构建 Oracle 是一项重要且困难的工作。

2.2 表达输入输出(I-O)关系

基于 ANN 的 I-O 关系分析使用随机法生成大量测试数据,输入到 SUT 得到相应的输出结果。将此大量 I-O 结果组成学习样本,通过对 ANN 进行构建、训练、裁减和规则抽取,析取出输入和输出的相互影响,得到精简的测试数据集。

2.3 模拟真实软件系统

GA 生成测试数据需要不断运行 SUT。单元测试时,驱动单元的运行比较可行,但对复杂的系统测试,不断运行目

标系统,存在效率低、成本高、甚至不可行的问题。为解决该问题,可采用 ANN 构建系统模型,替代目标系统,评价测试数据。学习样本集通过随机法生成的功能测试数据及其输出获得。

3 基于专家系统的测试数据生成

基于专家系统的测试数据生成研究较少,图 3 是一种基于规则的测试数据生成模型^[5]。该模型由测试数据的生成规则和分支谓词的符号信息,通过规则解释器生成测试数据。驱动模块用于控制SUT的执行,通过插装方式得到相应的结构覆盖信息。由测试数据及其覆盖信息输入到规则解释器生成新的测试数据,该循环直到达到规定的结构覆盖率或规定的测试数据数量为止。

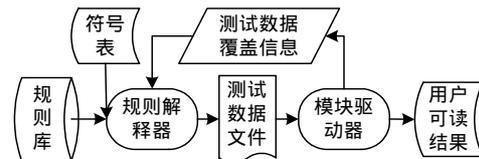


图 3 基于规则的测试数据生成模型

4 基于智能规划的测试数据生成

比较典型的两种类型是:针对图形用户接口(GUI)和命令语言接口(CLI)的测试数据生成^[6]。生成GUI测试数据包括设置阶段和规划阶段;生成CLI测试数据分为 3 步:建立对问题的描述,生成解决问题的规划,将规划转换成测试数据。智能规划生成测试数据的核心是智能规划器,但是设计有效的通用规划器存在一定的困难。生成的测试数据的正确性依赖于系统初始状态、测试目标和操作符。智能规划器假定操作符的结果均为已知,但是在软件系统中操作符的结果是不可知或不可预测的,所以基于智能规划的测试数据生成受到一定的制约。

5 讨论与展望

基于 GA 的结构测试数据自动生成的研究较为活跃,技术趋于成熟。但对于包含标志、非结构控制流等特性的复杂程序结构,或者遇到不可达路径,由于缺乏进化信息,使得搜索效率降低。为解决此问题,可以结合程序分析技术,将尽可能多的信息引入 GA。另外与 SUT 不同,GA 所需的最佳算法控制参数也不同,为避免反复实验确定控制参数的缺点,可以采用动态自适应技术,在 GA 进化过程中,自动调整控制参数和编码粒度,保证 GA 收敛性、稳定性和实用性。

基于 GA 的功能测试数据自动生成研究主要包括两个方面:(1)搜索 SUT 的边界响应时间,以便揭示响应错误;(2)采用适应值组合函数,生成功能故障的测试数据。在搜索边界响应时间的过程中,缺乏达到最优解的程度信息随着 SUT 结构复杂度的提高,搜索难度不断增加、搜索效率不断降低。在适应值组合函数的设计中,确定测试数据的发生概率、故障密度和故障严重度等组合因素及其合适的权重系数是一项比较困难的工作。在搜索功能故障中,缺乏达到发现故障的程度信息,当故障密度较小时,GA 退化为按发生概率随机搜索,这些问题需要进一步研究。

ANN 在软件测试中的应用主要体现在两个方面:(1)将 ANN 作为分类器,预测测试数据的揭错能力,据此选用、精简测试数据集。(2)ANN 用作系统逼近器,代替真实的 SUT 实施软件测试。ANN 无论是作为分类器,还是作为逼近器,

(下转第 207 页)