

文章编号: 1005-8451 (2011) 03-0053-04

基于成对组合的遗传算法生成联锁测试用例

李娟, 陈光武, 范多旺

(兰州交通大学 光电技术与智能控制教育部重点实验室, 兰州 730070)

摘 要: 计算机联锁自动测试的质量往往取决于测试用例的质量。本文提出一种新的设计思想: 将基于成对组合覆盖的遗传算法用于计算机联锁测试用例的生成和优化, 并以道岔测试为例对该算法的应用过程进行详细分析, 重点介绍成对组合覆盖方法、算法的整体框架和遗传操作设计。采用遗传算法对成对组合的启发式算法 AETG 进行改进, 有效地减少了冗余的测试用例, 提高测试用例质量和联锁测试效率。

关键词: 成对组合; 遗传算法; 联锁; 测试用例

中图分类号: TP391

文献标识码: A

Application of Genetic Algorithm based on pair-wise for generation of interlocking test case

LI Juan, CHEN Guang-wu, FAN Duo-wang

(Key Laboratory of Optical-electronic Technology and Intelligent Control, Ministry of Education, Lanzhou Jiaotong University, Lanzhou 730070, China)

Abstract: The quality of computer interlocking automatic test depended on the quality of test cases. A new design concept was presented in this paper, which applied Genetic Algorithm based on pair-wise coverage to implement the generation and optimization of interlocking test case. Taking switch test for example, the full text carried on a detailed analysis of this algorithm's application process, which focused on introducing the method of pair-wise coverage, the overall framework of algorithm and the design of genetic operation. Using Genetic Algorithm to improve the heuristic algorithm AETG of pair-wise could reduce the redundant test cases effectively and enhance the quality and efficiency of interlocking test.

Key words: pair-wise; Genetic Algorithm; interlocking; test case

计算机联锁系统是保证铁路行车安全和效率的实时控制和防护系统。联锁软件是联锁系统的安全性关键软件, 通过自动测试可尽早发现并排除联锁软件潜在的缺陷, 使联锁软件运行时发生故障的几率降到最低。联锁软件测试的实质是针对所设计的一组测试用例, 执行联锁程序, 以发现联锁程序的缺陷。测试用例的设计与生成是联锁软件测试中的难点和关键, 它决定着软件测试的质量, 影响软件测试的效率和速度。

测试用例的自动生成是在特定的数据空间中寻找符合测试标准的输入数据, 执行将实际结果与预期结果比较的过程。本文研究的重点是测试用例质量的提高和优化。该问题具有非线性特点, 而遗传算法的优化计算和全局寻优搜索策略适合于处理高度复杂的非线性问题。本文提出一种新的设计方案: 应用基于成对组合的遗传算法进行联锁测试用例自动生成和优化, 以提高联锁测试

用例的质量。

1 成对组合覆盖

成对组合覆盖是软件测试方法之一, 同时也是衡量测试充分性的一种准则。它是一种面向应用的测试方案, 按照两两覆盖的原则产生测试用例。成对组合的测试目标是所设计的测试用例集合对系统的所有对象可能取值的两两组合至少覆盖一次。满足成对组合的最小测试用例集实际上是经典的 NP-Complete 问题, 目前主要采用启发式算法或人工智能方法产生测试用例的最优解。

测试用例最优解的生成模型分析如下。设 DC 为道岔测试用例集合, DZ 为其一个子集, 对应每个测试用例都有相应的测试代价, DR 为测试需求。测试用例最优解问题等价于如何从测试用例空间 DC 中选取最小数量的用例子集 DZ, 使 DZ 能够以最小的测试代价覆盖测试需求集 DR。

设道岔测试有 3 个输入参数 A、B、C。A 表示

收稿日期: 2010-06-01

作者简介: 李娟, 在读硕士研究生; 陈光武, 教授。

道岔,有2个值A1、A2, $V(A)=\{1/3,5\}$; B表示操纵,有4个值B1、B2、B3、B4, $V(B)=\{\text{定位单独操纵,反位单独操纵,单独解锁,单独锁闭}\}$; C表示条件,有4个值C1、C2、C3、C4, $V(C)=\{\text{一般状态,进路锁闭,区段锁闭,引导总锁闭}\}$ 。测试需求集DR可描述为 $DR=\{dr1,dr2,dr3,dr4,dr5\}$,其中需求dr1为道岔定、反位单独操纵,dr2为道岔单独锁闭、操纵、解锁,dr3为进路锁闭后道岔单操测试,dr4为区段锁闭后道岔单操测试,dr5为引导总锁闭后道岔单操测试。按照全覆盖准则,共需 $2 \times 4 \times 4=32$ 个测试用例。而应用成对组合覆盖可以只包含16个测试用例。

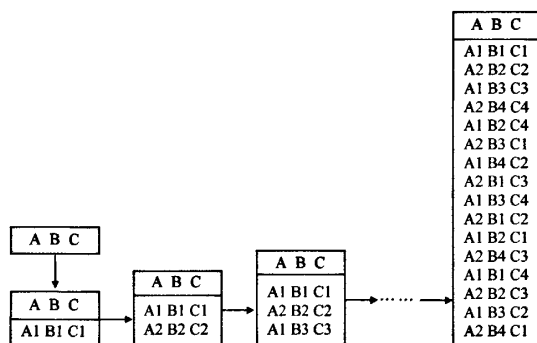


图1 道岔测试用例生成过程

测试用例生成过程如图1,采用的是AETG启发式算法。AETG(Automatic Efficient Test Generator,自动高效测试生成器)开始于一个空的测试用例集,每次根据贪心算法产生一组候选用例,从中选取能够覆盖最多未被覆盖的成对组合的用例为新的测试用例。最初未被覆盖的成对组合集 $WZ=\{(A1,B1),(A1,B2),(A1,B3),(A1,B4),(A1,C1),(A1,C2),(A1,C3),(A1,C4),(A2,B1),(A2,B2),(A2,B3),(A2,B4),(A2,C1),(A2,C2),(A2,C3),(A2,C4),(B1,C1),(B1,C2),(B1,C3),(B1,C4),(B2,C1),(B2,C2),(B2,C3),(B2,C4),(B3,C1),(B3,C2),(B3,C3),(B3,C4),(B4,C1),(B4,C2),(B4,C3),(B4,C4)\}$ 。首先选择测试用例 $DC1=(A1,B1,C1)$,然后去掉WZ中的(A1,B1)、(A1,C1)和(B1,C1),以剩下的组合中出现次数最多的A2为中心选择 $DC2=(A2,B2,C2)$ 。按照同样的方法依次递归产生新的测试子集,直到集合WZ为空。测试用例集和测试需求间的覆盖关系如表1。

表1 测试用例与测试需求覆盖表

测试需求	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
dr1	x	x			x			x		x	x		x	x		
dr2			x	x		x	x		x			x			x	x
dr3	x						x	x		x					x	
dr4			x										x		x	
dr5				x	x				x					x		

此例中利用16个交互测试用例组合覆盖测试需求,能有效地避免测试用例集之间产生的组合爆炸问题。采用传统的贪心搜索方法求得的测试用例集并非最小化的测试用例子集,其中包含大量的冗余测试用例,影响了测试效率。

2 遗传算法的设计

本文在设计遗传算法时主要考虑3个因素:可靠性,收敛性,稳定性。

解决满足覆盖需求的软件测试用例集生成问题时,通常先以随机方法产生初始种群,然后将种群中的不可行个体通过选择、交叉、变异转化为可行个体,再根据种群中个体的适应度函数值选取最优的测试用例子集。基于上述思路,本文将GA嵌入到AETG中,并从贪心算法改进为遗传算法生成新的测试用例集。以道岔测试为例,所设计的算法框架如图2。

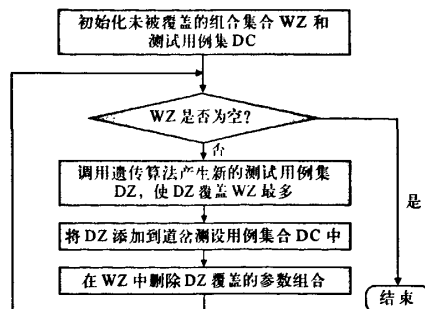


图2 算法的整体流程

其中遗传算法的代码结构如下所示:

Genetic Algorithm Program

Begin

t<-0;

initialize(); // 随机产生N个个体进行种群初始化

execute solution(); // 运行N个初始个体

```

    evaluate();           // 评价初始种群，即计算
                           // 计算N个初始个体的适应度函数值
    while (not termination-condition)
    {
        t<-t+1;
        select();         // 选择复制操作，选择
                           // 个体到下一种群
        crossover();       // 确定杂交算子，进行
                           // 交叉融合
        mutate();          // 不可行个体的变异操作
        execute solution(); // 运行N个新产生的个体
        evaluate();        // 计算N个新个体的适
                           // 应度函数值
    }
    End

```

2.1 参数编码方式

遗传算法处理过程中参数编码对算法的效率、性能有很大的影响。为了将问题的潜在解使用适合遗传算法的基因编码表示，本文将组合测试中的输入参数映射到位串空间 $B^l = \{0,1\}^l$ 上，然后在其上进行遗传操作，得到的结果再通过解码过程还原成其表现型以进行适应度函数值的评估。根据各个参数的取值范围确定其编码长度。若一个参数的最大取值为 L_i ，且 $2^{n-1} < L_i \leq 2^n$ ，则该参数的编码长度为 n 。道岔测试项中 DC 由 3 个输入参数组成 (A,B,C)，参数取值个数分别为 2、4、4，采用参数取值的序号表示参数的实际值，DC 可用长度为 5 的二进制编码表示，如图 3。测试用例 (A2, B2, C3) 的编码表示为 10110。

2.2 适应度函数

适应度函数是用以区分种群中个体好坏的演进过程。利用遗传算法生成的测试用例集，在满足成对组合覆盖的前提下，其例数尽可能的少。关于进化个体的适应度可按以下公式计算：

$$F(DC_j) = \text{Cov}(DC_j) / \text{Cost}(DC_j) \quad (1)$$

其中 $\text{Cov}(DC_j)$ 是用例集 DC_j 的测试覆盖度， $\text{Cost}(DC_j)$ 是用例集 DC_j 的测试运行代价。 $\text{Cov}(DC_j)$ 可按式 (2) 计算， $\text{Cost}(DC_j)$ 可按式 (3) 计算。

$$\text{Cov}(DC_j) = \sum_{DZ \in DC_j} \text{Cov}(DZ) \quad (2)$$

$$\text{Cost}(DC_j) = \sum_{DZ \in DC_j} \text{Cost}(DZ) \quad (3)$$

测试用例中的覆盖信息由表 1 给出，在第 j 列中有多少个“X”，就代表测试用例 DC_j 覆盖了哪些

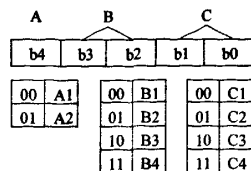


图3 道岔测试用例的编码表示

测试需求。设向量 $X_j = [x_{j1}, x_{j2}, \dots, x_{jk}]$ 表示表 1 中第 j 列的信息，向量 $W = [w_1, w_2, \dots, w_k]$ 表示每个测试需求的权重，用例子集 DZ 覆盖度可按下式计算：

$$\text{Cov}(DZ) = \sum_{i=1-k} (x_{ji} \times w_i) \quad 1 \leq i \leq k \quad (4)$$

从以上计算方法可以看出，F 值越高表示单位测试代价下测试用例集 DZ 的覆盖度越大。

2.3 遗传操作设计

遗传算法的基本操作包括 3 种：选择、杂交和变异操作。

不同的选择策略将导致不同的选择压力，从而影响算法的收敛速度。选用轮赌盘模型，选择算子从当前群体中按照某一概率选择一些个体作为新生种群的父辈，某个体 X_i 被选择的概率 P_i 与其适应度函数值成正比。设累积函数 $L_j = \sum_{i=1-j} P_i$ ，每个个体 DC_j 处于选择区间 $[L_j, L_j+1)$ ，计算时每一轮产生一个 $[0,1]$ 均匀分布的随机数，以该随机数为指针来确定对应选择区间的个体。个体的适应值越大，它被选择到的机会越多，其基因被遗传到下一代的可能性越大。

杂交运算是遗传算法区别于其他进化算法的重要特征。它的设计与所研究的问题密切相关，本文采用单点杂交操作，能较少的破坏优良个体的性状，同时有效地产生较好的新个体。单点杂交是指在个体编码串中只随机设计一个杂交点，然后在该点相互交换两个配对个体的部分染色体。两个长度为 n 的个体编码串进行单点杂交的过程如图 4。

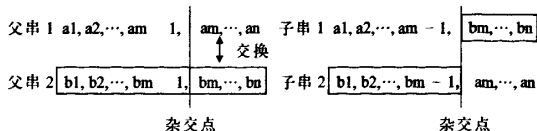


图4 单点杂交示意图

变异操作可改善遗传算法的局部搜索能力，维持群体的多样性，防止出现过早收敛。本文中二值基因链模型，采用基本位变异操作，即对指定的

(下转 P58)

dump。很多时候没有必要关闭数据库来停止 hanging,若一定要关闭,应在关闭前获取查看这些诊断信息,以便找出错误原因。

3.4 管理表空间

定期通过 Oracle Enterprise Manager Console 应用,检查数据库表空间的使用情况。当表空间利用率超过 90%,通过扩充数据文件的方法扩大数据库相应的表空间。一般监控表空间在 80% 左右,就可以对数据库表空间进行一些处理,释放部分空间,避免潜在故障发生。

4 六号线 AFC 系统数据库的优化

Oracle 数据库的性能设计在实际应用中,仍需要优化。主要的工作内容有:优化、恢复和备份。从调整 AFC 中央主机内存分配、调整操作系统参数,管理数据库备份数据、优化自由结构 OFA (Optimal flexible Architecture)、使用数据库连接池和 SQL 语句优化方面操作,比较容易优化功能。

4.1 优化自由结构 OFA (Optimal flexible Architecture)

优化自由结构 OFA,在数据库中可以高效自由地分布逻辑数据对象,数据库逻辑设计时,系统数据和用户数据分开、一般数据和索引数据分开、低活动表和高活动表分开等。

4.2 使用数据库连接池和 SQL 语句优化

(1) 使用直接的 OLE DB 数据库连接方式,提高处理速度。(2) 使用 Connection Pool 机制,提高系统的响应速度。(3) SQL 语句优化。减少对数据库的查询次数,即减少对系统资源的请求,使用快照和显形图等分布式数据库对象可以减少对

数据库的查询次数,同时尽量使用相同的或非常类似的 SQL 语句进行查询,这样不仅充分利用 SQL 共享池中的已经分析的语法树,要查询的数据在 SGA 中命中的可能性也会大大增加。

系统设计要充分考虑通用性和易用性。通过采用 2 台 HP rp4640 小型机做主机互备,以及 RAIDS 冗余磁盘阵列体系来预防数据库的安全性和容灾性。以上优化措施,在提高数据处理速度、系统稳定性上达到不错效果。

5 结束语

改善 AFC 系统数据库功能和优化性能需要提高信息利用率、增强系统的分析能力,强化系统的整理分析原始数据和信息的处理能力,把票务系统与线网的信息管理系统相结合,找出票务收益规律,为城市公共交通服务和管理提供及时、准确的决策分析意见,是 AFC 系统的发展主要方向之一。

参考文献:

- [1] 刘伟. UNIX 基础教程[M]. 北京:清华大学出版社, 2004.
- [2] 张红光,李福才. UNIX 操作系统教程[M]. 北京:机械工业出版社, 2010.
- [3] 王彬. Oracle 10g 入门与提高[M]. 北京:清华大学出版社, 2007.
- [4] 上海轨道交通票务中心. 城市轨道交通自动售票系统专用技术说明-1.01[Z]. 上海轨道交通票务中心, 2006.
- [5] 孟宪虎,马雪英,邓绪斌. 大型数据库系统管理、设计与实例分析[M]. 北京:电子工业出版社, 2008.
- [6] 陈俊杰. 大型数据库 Oracle 实验指导教程[M]. 北京:科学出版社, 2010.

(上接 P55)

变异点进行取反运算。若测试用例子集 A 编码为: 10111010, 对选取的第 5 位基因值取反后产生新的 A 编码为: 10110010。杂交算子和变异算子的相互配合,共同完成对搜索空间的全局搜索和局部搜索,实现测试用例最优解的寻优过程。

3 结束语

遗传算法因其自身强大的鲁棒性和全局寻优

搜索能力,在解决非线性、大空间的测试数据自动生成中显示出独特的优势和高效性。将其嵌入到成对组合的启发式算法 AETG 中,可有效的减少冗余测试用例,更快的收敛到全局近似最优解,提高测试效率。本文以道岔测试为例,着重介绍了应用基于成对组合覆盖的遗传算法进行计算机联锁测试用例生成和优化的新想法,并对其应用过程进行了分析探讨,给出了具体的设计思路。