

基于多核并行遗传算法并行生成测试用例

陈清媛¹, 钟 诚¹, 李 智^{1,2}

(1 广西大学 计算机与电子信息学院, 广西 南宁 530004; 2 广西科技信息网络中心, 广西 南宁 530012)

摘 要: 通过将种群划分为多个子种群, 对每个子种群执行遗传操作, p 个核心并行执行遗传算法搜索测试路径, 以加速测试用例的生成; 通过在处理核心之间迁移子种群的个体, 使得交叉变异后得到的种群个体变得丰富, 算法能更好地全局寻优, 可以寻找到较多能够覆盖全部路径的测试用例. 实验结果表明, 与基于串行遗传算法生成测试用例相比, 多核并行遗传算法并行生成测试用例能够生成较多覆盖全部路径的测试用例且运行时间少.

关键词: 测试用例生成; 并行遗传算法; 多核系统; 线程级并行

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2013)11-0149-05

Multi-core Parallel Generating Test Cases Using Parallel Genetic Algorithm

CHEN Qing-yuan¹, ZHONG Cheng¹, LI Zhi^{1,2}

(1 School of Computer and Electronics and Information, Guangxi University, Nanning 530004, China;

2 Guangxi Scientific Information Network Center, Nanning 530012, China)

Abstract: A population is partitioned into several sub-populations, the genetic operations for each sub-population are executed, and parallel genetic algorithms are executed by p cores to search the test paths to speed up generating test cases. The individualities for population are enriched by migrating the individualities for sub-populations among cores, the optimization solutions are searched globally better, and more test cases for covering all the test paths are found. The experiment results show that compared with the sequential generating test cases method using genetic algorithm, the multi-core parallel generating test cases method using parallel genetic algorithm can obtain more generated test cases and require less execution time.

Key words: generation test cases; parallel genetic algorithm; multi-core systems; thread-level parallelism

1 引言

软件测试在保证软件质量的过程中扮演着极其重要的角色. 面向路径的测试是测试中最常用的测试方法, 对于程序中不同的路径, 需要设计测试数据来覆盖这些路径以实现目标路径的测试. 文献[1-4]采用遗传算法等对测试用例进行了研究. 文献[5]将复杂类簇划分成多个测试块, 多个处理核并行处理测试块, 在每个测试块中, 将类与类之间存在的关系以关系对的形式表示, 各处理核以多线程方式并行处理这些关系对, 利用 μ Java 工具为各个关系对生

成变异程序, 将生成的变异体与模型检测器 JPF 相结合来自动生成反例, 并对反例进行分析以提取出被测试类簇的测试用例.

测试用例数据的产生是一个不可判定性问题, 测试准确率和效率是测试用例生成的两个重要方面. 本文提出基于线程级并行遗传操作的高效、可覆盖全部测试路径的测试用例生成算法.

2 多核并行遗传操作算法生成测试用例

2.1 串行遗传算法生成测试用例

运用遗传算法^[6]设计生成测试用例主要包括分

收稿日期: 2013-01-15; 修回日期: 2013-02-22

基金项目: 广西自然科学基金项目(2011GXNSFA018152); 广西自然科学基金(桂科基 0728033)

析待测试程序和运用遗传算法搜索两个阶段:将遗传算法的一个个体作为一个测试用例,每个个体的进化就是测试用例的优化,生成测试用例.在每一步进化计算过程中,测试用例自动生成程序使用当前群体驱动待测试程序的执行,记录个体执行路径(即测试用例覆盖的路径),以覆盖路径的最大化执行为目标进行计算,产生下一代群体;经过多代进化之后得到最优种群或超过特定限制条件而结束.

基于遗传算法生成测试用例的步骤如下:

(1)分析待测试程序,找出参数并分析得到的程序控制流.

(2)根据程序的控制流对待测试程序进行插装.

(3)使用产生的种群的个体作为输入,运行插装后的待测试程序.

(4)评估种群,判断该种群个体是否满足结束条件,若满足结束条件,则转步(7);否则,进入下一步产生新的个体.

(5)对个体执行适应度计算,根据适应度对个体执行选择操作.

(6)对执行选择操作后的个体做交叉、变异运算,以生成新的种群,转到步(3).

(7)输出结果.

其中,分析待测试程序是测试用例生成的基础.通过对程序的分析提取有用的参数,这些参数的个数以及范围是遗传算法编码的依据,是遗传算法和测试用例生成的一种映射.对待测试程序结构进行分析,得到控制流,根据控制流确定程序的执行路径以及分支,对待测试程序进行插装,以使得测试用例生成程序运行时可根据评价规则对个体进行评价.

遗传算法搜索部分是测试用例生成算法的核心.搜索算法根据程序的插装输出覆盖路径,判断测试数据是否满足覆盖标准,如果测试数据满足程序的覆盖判断标准,得到最优解,那么输出结果;否则,一直反复进行遗传搜索,通过选择、变异、交叉操作改变染色体,以形成更好的种群,直到得到最优解或者达到限定的最大运行代数而结束.

遗传算法采用实数编码,它将个体测试用例的每个基因值用某一范围内的实数值表示,个体的编码长度等于其变量的个数.

适应度计算方法是:根据分支函数,在待测程序目标分支点前插入相应的适应度函数 $bf_i(x)$,若种群个体覆盖此分支路径, $bf_i(x) = 1$,否则 $bf_i(x) = 0$.在待测程序的结束处插入每个个体 i 的适应

度: $f_i(x) = \sum_{i=1}^n bf_i(x)$, n 为总分支数.

遗传算法选择个体的方法是:采用赌轮法选择个体,设群体的大小为 m ,个体 i 的适应度为 $f_i(x)$,则个体 i 被选择的概率为

$$Pr_i = f_i(x) / \sum_{j=1}^m bf_j(x) [6].$$

交叉操作是指对被选择的个体测试用例进行单点交叉.在个体串中随机设定一个交叉点,进行交叉操作时,该点的前或后的两个个体的部分结构进行互换形成两个新的个体.在交叉个体中,个体的交叉概率大小直接影响着遗传算法的性能.选择较大的交叉概率可以增强遗传算法的搜索能力,但是,此时好的种群个体容易被破坏,从而导致可能得不到覆盖全部路径的测试用例;交叉概率小,那么即使搜索迭代多次,种群中的个体变化也不大,使得满足覆盖全部路径的测试用例基本没有变化.

变异操作是指对个体的某些基因进行变异,也就是对染色体上的位串产生随机的变化^[6].通过变异可以增强种群个体的多样性,改善局部的搜索能力,防止没有找到覆盖全部路径的测试用例而出现收敛现象.但变异率大小的选择必须适度,变异率过大会引起算法生成的数据变多,使得为了搜索能覆盖全部路径的测试用例而需要增加大量的搜索时间;变异率过小则会影响迭代搜索区的范围,从而会导致寻找到能覆盖全部路径的测试用例减少,生成的测试用例质量降低.

2.2 多核并行遗传算法并行生成测试用例

在多核系统上基于并行遗传算法并行生成测试用例方法的关键是:一方面将串行遗传算法并行化,设计出线程级并行遗传算法;另一方面,将种群划分成多个子种群,将子种群分配给多核系统的 p 个核心,每个处理核心负责处理一个子种群, p 个核心并行执行相关遗传操作, p 个核心之间相互交互数据、并行搜索生成测试用例.

遗传算法算法内部并行性主要采用并行调度策略,将循环迭代均匀分布到各个处理核心执行多线程并行处理;同时,对数据相关性进行分析,如果循环部分数据相关且计算量大,那么分析是否可以改写程序以消除数据相关性.通过将种群划分成若干个子种群分配给 p 个核心, p 个核心同时执行遗传操作,并行执行适应度计算、选择、交叉和变异产生子种群的过程.

在多核系统上基于并行遗传算法并行生成测试

用例算法形式描述如下:

算法 1 多核并行遗传算法并行生成测试用例
begin

(1)分析待测试程序,寻找出被测程序参数及执行路径控制流;

(2)根据控制流对待测试程序进行插装;

(3)产生初始种群并将种群划分成 p 个子种群;

(4)for $i = 0$ to $p-1$ do in parallel

(4.1)处理核心 $core_i$ 利用产生的子种群中的个体运行待测试程序;

(4.2)处理核心 $core_i$ 评估子种群,判断该子种群个体是否满足结束条件,若满足结束条件则转步(6);否则,执行下一步以产生新的个体;

(4.3)处理核心 $core_i$ 对子种群中的个体执行选择、交叉、变异运算,对交叉变异后的个体执行适应度计算;

(4.4)处理核心 $core_i$ 判断是否达到指定迁移的代数,若达到指定迁移的代数则进行迁移,转至(4.5),如果没有达到指定的迁移代数,则对新子种群,转至步(4)进行迭代处理;

(4.5)处理核心 $core_i$ 从其子种群中随机提取一个个体;

(4.6)处理核心 $core_i$ 从其子种群中随机选出 1 个被替代个体并删除;

(4.7) 处理核心 $core_i$ 从其他子种群中随机地选取 1 个个体作为 $core_i$ 自己对应种群的一个新个体;

end for

(5)循环执行步(4)~(5);

(6)输出结果,算法结束.

多核并行遗传算法并行生成测试用例算法执行过程如图 1 所示.

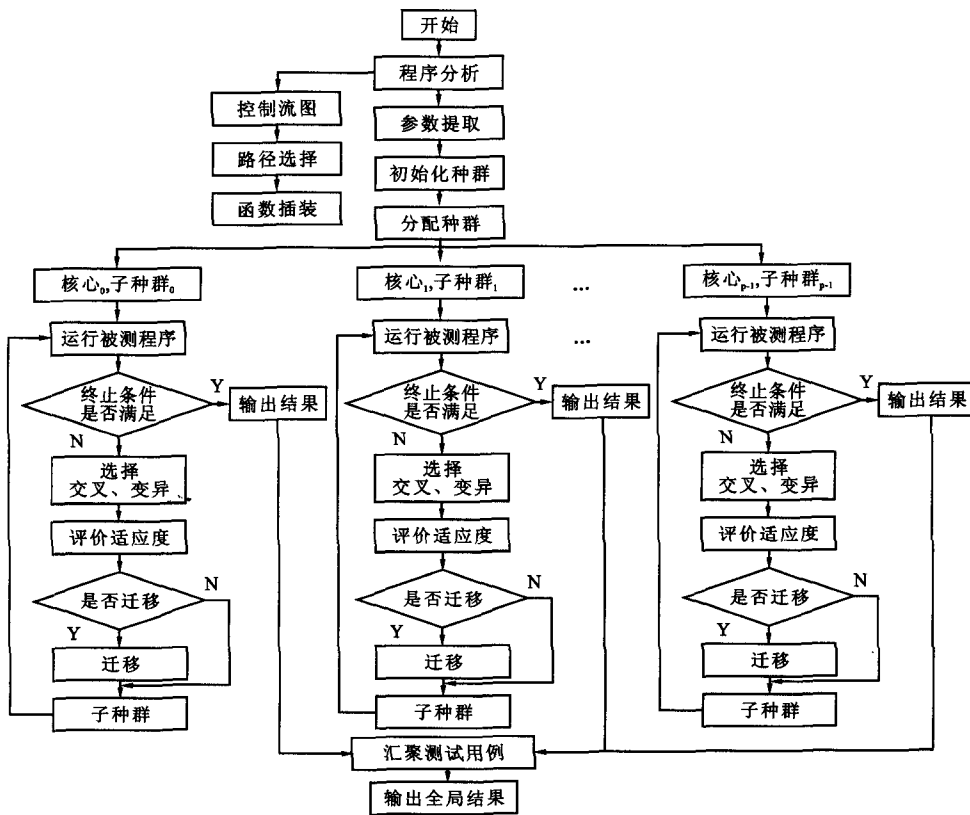


图 1 基于并行遗传算法并行生成测试用例过程

在多核计算机系统中,通过将种群划分为多个子种群,对每个子种群执行遗传操作,由 p 个核心并行执行遗传算法搜索测试路径,可以显著加速测试用例的生成.另一方面,通过迁移子种群的个体,丰富了每个核心处理的种群个体,交叉变异后得到的种群个体变得丰富,能更好地全局寻优,使得算法

可以找到较多能够覆盖全部路径的测试用例.

3 实验

实验硬件环境为 AMD Athlon(tm) CPU 主频 3.00 GHz、内存容量 3.00 GB 的四核计算机,软件环境为 Windows XP 操作系统、Microsoft Visual

Studio2008,采用 C 语言和 OpenMP^[7]混合编程实现本文提出的多核并行遗传算法并行生成测试用例算法,并与基于串行遗传算法生成测试用例算法进行实验性能比较.实验测试以生成覆盖全部路径的测试用例为目标.

本文以判断三角形程序为待测试程序,通过对输入三角形的三边判断三角形的类型.实验对三角形三边进行实数编码、采用赌轮选择方法选择个体、精华模型来保留上一代最好的个体、单点交叉和均匀变异.

待测试判断三角形程序及其控制流分别如图 2 和图 3 所示.

```
double triangle(double a,double b,double c)
{
  ① string Triangle="Not a Triangle"
  ② if((a+b>c)&&(b+c>a)&&(c+a>b))
  {
    ③ if((a!=b)&&(b!=c)&&(c!=a))
    {
      Triangle="Scalene"; //一般
    }
    ④ else
    {
      ⑤ if(((a==b)&&(b!=c))||((b==c)&&(c!=a))||((c==a)&&(a!=b)))
      {
        Triangle="Isosceles"; //等腰
      }
      ⑥ else
      {
        Triangle="Equilateral"; //等边
      }
    }
  }
  ⑦ else {
    Triangle="No triangle!"; //不是三角形
  }
}
```

图 2 待测试判断三角形程序

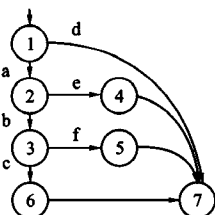


图 3 测试判断三角形程序控制流

从图 3 可看出,所有可能覆盖的判断三角形程序执行的路径分别为 $d, a \rightarrow e, a \rightarrow b \rightarrow f, a \rightarrow b \rightarrow c$.

图 4 给出了种群大小变化、最大代数 1 000、交叉率为 0.8、变异率为 0.07 时,分别运行基于串行遗传算法生成测试用例算法^[4](简称为 SerialGATestCase 方法)、基于并行遗传算法并行生成测试用例算法(简称为 ParaGATestCase 方法)生成覆盖全部路径的测试用例所需的时间.

从图 4 中可以看到,多核并行遗传算法并行生成测试用例算法较之基于串行遗传算法生成测试用例算法在所需运行时间上具有明显的优势.

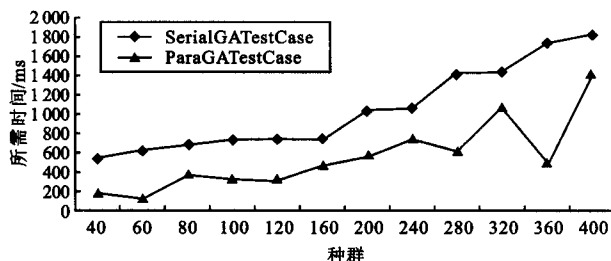


图 4 基于串行和并行遗传算法生成测试用例算法生成覆盖全部路径的测试用例所需的时间

实验给出了种群大小分别为 40、100 和 400,以最大代数 1 000 为作为结束条件时,基于串行遗传算法生成测试用例算法、多核并行遗传算法并行生成测试用例算法生成覆盖全部路径的测试用例数目的对比.结果表明:基于并行遗传算法并行生成测试用例算法可以找到更多覆盖全部路径的测试用例,特别是对于构成等边三角形的情形,基于并行遗传算法并行生成测试用例的方法可以找到更多满足覆盖条件的测试用例,使得该路径测试充分;而基于串行遗传算法生成测试用例的方法,只能找到满足覆盖条件的 1 个测试用例.

综上所述,多核并行遗传算法并行生成测试用例的方法在运行时间上和寻找到满足覆盖全部路径的测试用例数量上都有明显的优势.

4 结束语

本文设计实现了线程级并行遗传算法,通过将种群划分为多个子种群,对每个子种群执行遗传操作, p 个核心并行执行遗传算法搜索测试路径,可以显著加速测试用例的生成;通过迁移子种群的个体,丰富了每个核心处理的种群个体,交叉变异后得到的种群个体变得丰富,能更好地全局寻优,使得算法可以寻找到较多能够覆盖全部路径的测试用例.下一步将研究在多核系统上融合遗传算法和蚁群算法并行生成测试用例的方法.

参考文献:

- [1] Debasis Mohapatra, Prachet Bhuyan, Durga P Mohapatra. Automated test case generation and its optimization for path testing using genetic algorithm and sampling[C]//Proc of 2009 International Conference on Information Engineering. Piscataway, New Jersey: IEEE Press, 2009:643-646.
- [2] Dong Yuehua, Peng Jidong. Automatic generation of software test cases based on improved genetic algorithm[C]//Proc of 2011 International Conference on

- Multimedia Technology. Piscataway, New Jersey: IEEE Press, 2011:227-230.
- [3] 王丹,高丰,朱鲁华. 基于遗传算法的 Fuzzing 测试用例生成模型[J]. 微电子学与计算机,2011,28(5):130-135.
- [4] Wang Xibo, Su Na. Automatic test data generation for path testing using genetic algorithms [C]//Proc of 2011 Third International Conference on Measuring Technology and Mechatronics Automation. Los Alamitos, CA: IEEE Computer Society Press, 2011: 596-599.
- [5] Zhong Cheng, Wu Xi-hua. Thread-level parallel generating inter-class test cases [C]//Proc of 4th International on Parallel Architectures, Algorithms and Programming. Los Alamitos, CA: IEEE Computer Society Press, 2011:42-46
- [6] 陈国良,王煦法,庄镇泉,等. 遗传算法及其应用[M]. 北京:人民邮电出版社,2001.
- [7] 多核系列教材编写组. 多核程序设计[M]. 北京:清华大学出版社,2007.

作者简介:

陈清媛 女,(1987-),硕士研究生. 研究方向为可信软件、网络与并行分布计算.

钟 诚 男,(1964-),博士,教授,博士生导师. 研究方向为可信软件、网络与并行分布计算.

李 智 男,(1967-),硕士,研究员. 研究方向为网络与并行分布计算.

(上接第 148 页)

- [8] 蒋建春,汪同庆. 一种异构多核处理器嵌入式实时操作系统构架设计[J]. 计算机科学,2011, 38 (6):298-303.

作者简介:

刘清涛 男,(1986-),硕士研究生. 研究方向为计算机体系结构.

于立新 男,(1967-),硕士,研究员. 研究方向为计算机体系结构、VLSI 和 SoC 设计.