

NSGA-II 面向多目标测试数据模型的生成

周文晶,王占刚

(天津工业大学 计算机科学与技术学院,天津 300387)

摘要:测试数据的自动化生成的实现是软件测试自动化的重要研究项目。当前很多研究人员使用多种方法实现测试数据的自动生成,但生成的测试数据一般仅实现最大覆盖率的测试标准。在测试数据生成问题上,希望生成的测试数据能够达到最大的覆盖率,同时也希望生成的测试数据集越小越好,可以降低执行时间,同时提高执行效率。文中从覆盖标准和内存消耗两个方面对测试数据进行评估,采用多目标优化算法 NSGA-II,实现同时满足最大分支覆盖率和最大内存分配的测试数据的自动生成。实验表明,NSGA-II 算法生成的测试数据比其他多目标优化算法能更好地满足两个目标。

关键词:NSGA-II 算法;多目标测试数据;覆盖率;内存分配

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2014)04-0021-04

doi:10.3969/j.issn.1673-629X.2014.04.005

Generation of NSGA-II for Multi-objective Test Data Model

ZHOU Wen-jing, WANG Zhan-gang

(School of Computer Science and Software, Tianjin Polytechnic University,
Tianjin 300387, China)

Abstract: The realization of the test data automatic generation is an important research project of software test. At present many researchers use a variety of methods to realize the automatic generation of test data, but the generated test data are generally only to achieve maximum coverage rate. In test data generation, hope that generated test data can achieve maximum coverage rate and generated test data set is as small as possible, therefore reducing the execution time and improving the execution efficiency. To evaluate testing data from two aspects of coverage standard and memory consumption, adopt multi-objective optimization algorithm, NSGA-II, to achieve automatic generation of test data which satisfies the biggest branch coverage rate and the maximum memory allocation at the same time. The experimental results show that the NSGA-II to generate test data is better to meet the two goals than the other multi-objective optimization algorithms.

Key words: NSGA-II; multi-objective test data; coverage rate; memory allocation

0 引言

在软件工程领域,多目标优化方法的使用是迫在眉睫的研究领域^[1]。由于多目标优化可以使问题在多个目标之间取得最优解,它已经被研究用来减少项目开发时间和软件项目开发时间^[2]、软件的重构和设计^[3]、选择软件发布需求^[4]等多目标优化问题。在测试数据自动生成研究中,很多研究人员也在这方面不断的尝试。文献[5]中作者提出一种新的适应函数生成测试数据覆盖特定的单个路径,由于多数研究中测试数据生成器每次仅能搜索一个目标数据,使得许多中间有用的数据被浪费掉,该文章构造了另一种适应函数与单路

径函数相结合,驱动遗传算法完成多路径的覆盖,避免重复搜索,利用中间种群个体覆盖不同路径。该文章提高路径覆盖的同时减少了消耗的时间,达到多个目标的同时实现。文献[6]已经开始采用多目标进化算法来解决测试数据生成的多目标问题,文章通过加权将多个目标函数归为单个目标函数的计算,实现多路径覆盖及缺陷检测的目标。Ghiduk 等人使用遗传算法生成满足数据流覆盖的测试数据^[7],同时减少生成测试数据的数量,进而减少搜索时间。

文中介绍的方法与上述方法的不同之处在于,使用了高效的多目标^[8]优化算法 NSGA-II 来解决测试

收稿日期:2013-06-18

修回日期:2013-09-24

网络出版时间:2014-01-28

基金项目:教育部博士点基金项目(200800580004);天津市自然科学基金资助项目(043600711)

作者简介:周文晶(1986-),女,硕士研究生,研究方向为计算机软件测试;王占刚,副教授,研究方向为计算机检测应用、计算机网络安全、软件测试。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140128.1200.057.html>

数据的自动生成,使生成的测试数据满足最大分支覆盖率 and 最大内存分配的目标,得到最优化的测试数据集。并且该方法较之加权方法生成的测试数据虽然在单个目标上不是最突出的,但在多个目标之间却是最优秀的,它不需要考虑目标间的权值,因此更稳定。

1 面向多目标的测试数据自动生成模型

为解决这种多目标优化的测试数据自动生成问题,文中给出生成这种测试数据的框架。框架中引入 NSGA-II 算法,该算法用来生成测试数据并评估测试数据集是否满足设定的多个目标。框架分为三个模块:配置模块、主体模块、测试模块。

配置模块接收来自测试人员的参数,参数包括最大执行时间或最大代数,交叉和变异概率、种群大小,评价适应度的目标函数;主体模块是应用 NSGA-II 算法生成测试数据,并对生成的测试数据评价,该模块要接收配置模块处理后传递过来的参数,并将生成的测试数据集传递给测试模块;测试模块接收主体模块传递过来的测试数据集,并将测试数据转化为要求的格式,执行测试数据,最后返回测试数据覆盖信息给主体模块。

2 目标函数设定

种群中的每个个体可以根据两个或多个目标函数进行优化,文中将测试数据的分支覆盖率、动态内存分配作为目标函数,对种群个体进行评定。

对于测试数据对代码覆盖率的评定有很多标准,如语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、多重条件覆盖^[9]。文中主要针对分支覆盖对测试数据进行考量,这主要是由于分支覆盖能够更多地发现程序中存在的 bug。完成目标分支覆盖就是要生成充足的测试数据,使得测试数据运行被测程序时,能够覆盖被测程序控制流程图中的分支。设计分支覆盖的适应度函数,就是计算实际测试用例覆盖的路径与目标路径的距离,具体适应度函数如下:

$$\text{FitnessFunc} = \text{Approlev} + S(d) \quad (1)$$

$$S(d) = 1 - 1.001^{-d} \quad (2)$$

其中,Approlev 表示实际路径与目标路径的相符程度,用被测程序的控制流程图表示,Approlev = 目标路径长度 - 实际执行路径与目标路径重叠长度; d 表示实际路径与目标路径不同的第一个分支节点处,实际路径分支与目标路径分支的距离。

分支覆盖率的计算公式为:

$$\text{分支覆盖率} = \frac{\text{TestCov}}{\text{TestReq}} \times 100\% \quad (3)$$

文中生成的测试数据在满足分支覆盖的同时,希

望生成的测试数据能够执行被测程序的动态内存分配。这种方案主要用在测试人员知道内存被高度限制或者测试人员相信存在内存泄漏或者存在废弃的空指针的情形。为了计算内存分配,为源码设置一个全局变量,用来计算函数执行时所分配的所有字节数。废弃指针得到的自由内存不被计算在内,因为释放的全局指针所分配的所有内存空间不在被测函数的范围内。

3 参数设置

改进的非支配遗传算法(NSGA-II)与普通遗传算法的遗传操作是一样的,同样需要经过选择、交叉、变异,来生成新一代种群。本节主要对 NSGA-II 算法生成测试数据过程中所采用的遗传算子的方法进行设置,NSGA-II 算法对个体采用的编码方式为实数编码方式^[10]。实数编码是为了弥补二进制编码在高维、连续优化问题上的不足提出的。实数编码中,染色体的表示是用参数输入域内的浮点数表示的,染色体的长度是由参数个数决定的。实数编码使用的是参数的真实值。实数编码的优点是计算精确度高,便于和经典连续优化算法结合,适用于数值优化问题。但其缺点是适用范围有限,只能对连续变量问题使用。

选择算子,采用精英策略和拥挤锦标赛选择法。精英选择就是将当前最优个体直接复制到下一代种群中,以确保优秀个体不会在进化过程中丢失,同时也加快了算法收敛的速度。锦标赛选择法基本思想就是每一轮中随机选取 n 个个体,计算每个个体的适应度值进行比较,比较之后选择适应度值最高的个体直接作为下一代种群的父本。由于遗传算法对种群个体实施遗传操作的同时,也会将一部分优秀个体的优秀基因给破坏掉,使得这部分个体在计算适应度时不能被优先选择下来,甚至可能会被筛选掉,导致算法的运行效率降低。而锦标赛选择法恰好可以解决此问题,避免该情况的发生。

交叉算子,采用 SBX(Simulated Binary Crossover,模拟二进制杂交)方法,该方法对实数编码的父代个体进行交叉操作,根据给定的随机交叉点,交换父代个体位于交叉点两侧部分的编码。

变异算子,采用多项式变异法^[11],该变异方法是由 Deb 等提出的,目前研究多目标优化的研究人员多采用该方法。多项式变异的变异形式为:

$$v_i = x_i \pm r_i \cdot \delta \quad (4)$$

$$\delta = \sum_{j=0}^{15} a_j \cdot 2^{-j} \quad (5)$$

其中, a_j 等于 1 或等于 0,等于 1 时的概率为 $1/16$; x_i 是输入参数; r_i 是每个输入参数的变异区间范围。

4 加权遗传算法 (WGA)

为了更好地突出 NSGA-II 在测试数据自动生成中的优势,在实验仿真部分将 NSGA-II 的实验数据与加权遗传算法 (WGA) 和随机法进行对比。实验结果证明,基于 NSGA-II 算法生成的测试数据在分支覆盖和动态内存分配上确实比 WGA 算法和随机法有优势。

WGA 算法在解决多目标问题时,每个目标同时有效,将多个目标通过加权合成单个目标,因此加权遗传算法仅能找到一个最优解。当两个目标存在矛盾时,为了使 WGA 有效,目标要按某种方式进行优先排序。优先级高的,赋予的权重就大,相反,优先级低的目标赋予的权重就小。文中,目标最大化内存分配,不是明确定义或明确计量的,因为内存消耗没有一个明显的最佳效果。如果加大它的权重,可能会抑制分支覆盖度;若过分减少它的权重,可能导致它的不重要性,使得动态内存分配最大化的目标降低至随机搜索。

对于目标分支覆盖和动态内存分配,WGA 算法的适应度函数定义为:

$$1.001^{-b} \times w_b^{-1} + 1.0 \times d$$

(6)

其中, b 是原始分配的字节数; w_b 是目标内存分配的权重; d 是分支距离,分支距离的权重设为 1.0。

解决多目标优化问题实际就是为多个存在矛盾冲突的目标找到同时满足这些目标的最优解,这个最优解不是一个解,而是一组最优解集 Pareto^[12]。NSGA-II 对原 NSGA 算法进行了改进。在 NSGA 进行非支配排序时,种群中每个个体都要与其余个体进行比较,那么一个规模为 N 的种群,目标函数个数为 m ,则其运算复杂度为 $O(mN^2)$,即进行一次 Pareto 分层的时间复杂度为 $O(mN^2)$ 。如果每一层只含有一个个体,则需要对种群个体进行 N 次分层,那么这种情况下的时间复杂度为 $O(mN^3)$ 。改进后的快速非支配排序方法使得算法的计算复杂度 $O(mN^3)$ 降低到 $O(mN^2)$ 。

5 实验结果分析

5.1 NSGA-II、WGA 和随机法的比较

三种算法在实验中各运行 10 次,观察三种算法生成的测试数据的分支覆盖率和内存分配情况。得到的实验结果如图 1 所示。

第一个实例是一个实际应用中的程序,它含有 32 个分支,从图 1 可以看出,三种算法生成的测试数据对分支的覆盖率最高的是 WGA 算法,最低的是随机法,内存分配最高的是 NSGA-II 算法,它分配到的字节要比 WGA 算法高出 66%,最低的是随机法。NSGA-II 获得的动态内存要比其他两种方法高出很多,它对分支的覆盖率 85% 满足测试的需求,在两个目标中得到的是最优解集。

图 2 显示在第二个实例中,随机法对容易覆盖的分支很容易达到较高的覆盖率,在第二个实例中它对分支的覆盖率达到 98%,但运行 10 次后它分配到的字节仅为优化内存分配的 33%。WGA 算法的权重设置为 3:2,因此获得较好的内存分配,但是分支覆盖率却是最差的。NSGA-II 的覆盖率达到 95%,获得的内存为 10 000,三种算法中两个目标均获得最优的测试数据。

5.2 NSGA-II 算法和 WGA 算法的比较

综合上述分析,NSGA-II 和 WGA 算法得到的数据结果较之随机方法生成的测试数据更优秀,随机方法在生成测试数据时存在很大的不稳定性,不适合直接应用到实际中。下面单独对 NSGA-II 和 WGA 算法进行分析比较,图 3 和图 4 为实例一中单从分配的字节数和分支距离上比较 NSGA-II 算法和 WGA 算法的优劣。

从图 3 中可以看到,NSGA-II 算法生成的测试数据在覆盖分支的同时,能为单个分支分配较多的内存,即使在分支较难分配的内存时,如第 16、17、21 分支,NSGA-II 算法为其争取分配的内存相较于 WGA 算法

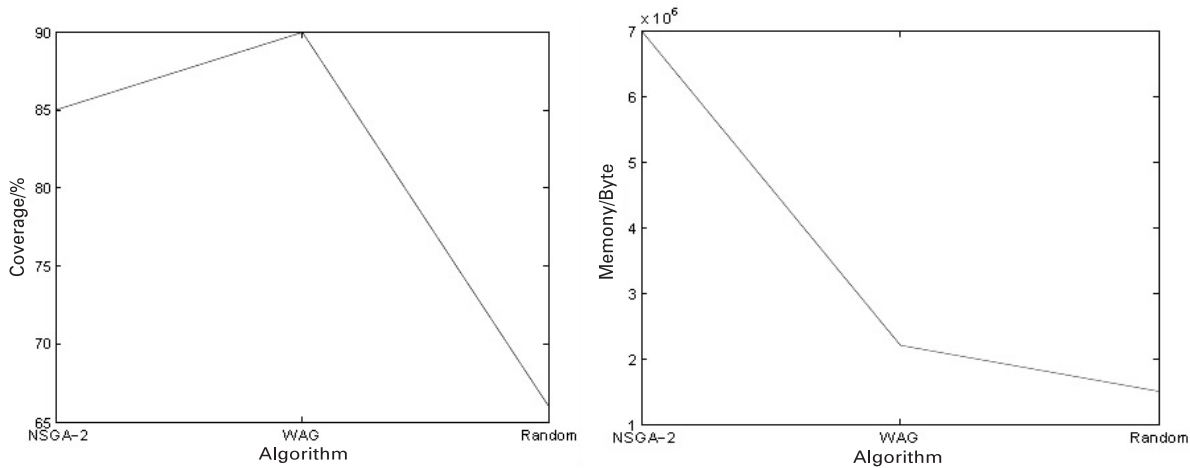


图 1 实例一中三种算法生成测试数据的对比结果

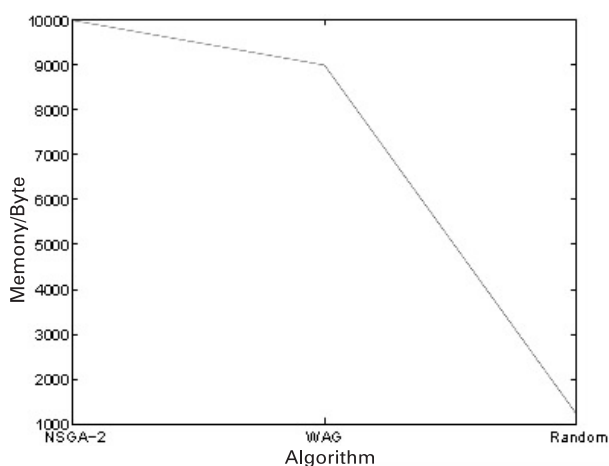
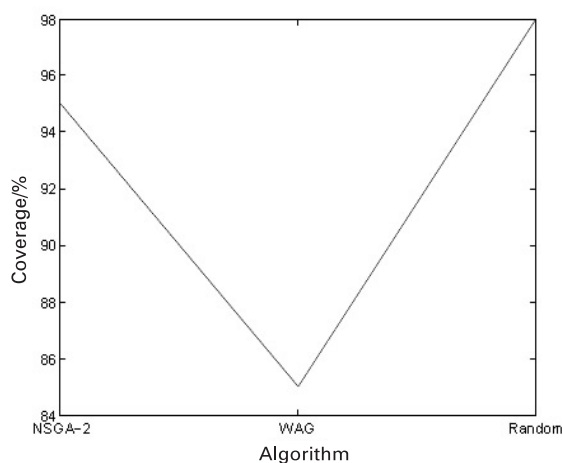


图 2 实例二中三种算法生成测试数据的对比结果

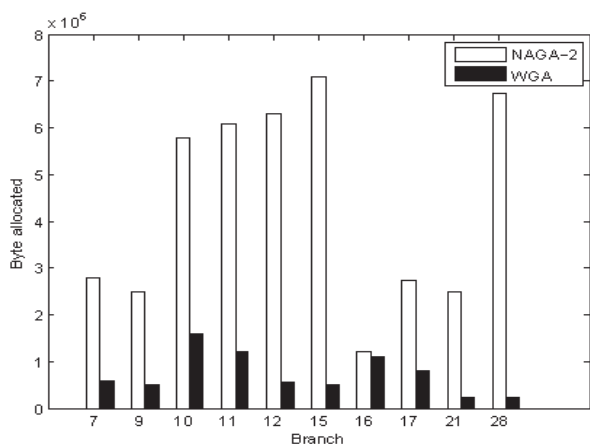


图 3 NSGA-Ⅱ和 WGA 为特定分支分配的内存

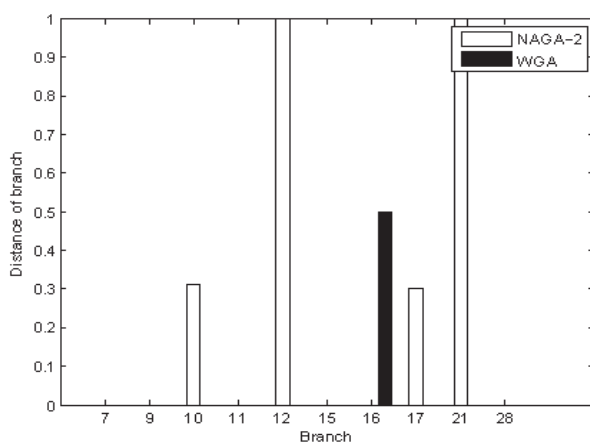


图 4 NSGA-Ⅱ和 WGA 中特定分支的分支距离多。且从图中可以看到 NSGA-Ⅱ算法在为分支分配内存上与 WGA 算法的差距很大,从 12、15、28 分支可以看到,这种算法分配到的内存不在一个量级上。

图 4 是从分支距离上显示 NSGA-Ⅱ算法与 WGA 算法生成的测试数据在分支覆盖标准中,测试数据执行的分支与预期分支的偏离程度。从图中可以看到 WGA 算法生成的测试数据能更好地满足分支覆盖标准,实例分支中大部分的分支距离为 0,也就是能完全覆盖。相较于 WGA 算法,NSGA-Ⅱ算法在分支覆

盖程度上就稍显逊色,但是它的分支距离范围也控制在(0,1)之间,在多目标优化问题上,对于单个目标可能不是最好,但是对于多个目标它却是最优的。

6 结束语

通过实验的对比可以看出,NSGA-Ⅱ算法生成的测试数据能使两个目标同时达到最优,它能很好地权衡两个目标,进而为两个目标找到非劣最优解集。而 WGA 算法由于需要对目标进行加权,权值的大小直接影响两个目标的侧重点,因此 WGA 算法得到的测试数据不能使两个目标同时达到最优,并且要找到最有效的权值也是一项比较困难的工作。三个算法中,随机法最简单,但是随机法生成的测试数据对分支的覆盖率和内存的分配都比较低,不能满足人们对测试数据的要求,因此该方法一般很少使用。文中在多目标优化方面,主要针对测试数据的覆盖率、内存消耗进行评估,而影响测试数据生成的因素不仅仅只是这几方面,比如缺陷检测能力、执行时间、其他覆盖标准等。未来的工作可以对其他目标展开研究。

参考文献:

- [1] Harman M, Mansouri S A, Zhang Y. Search based software engineering: A comprehensive analysis and review of trends techniques and applications[R]. London: King's College London, 2009.
- [2] Alba E, Chicano F. Software project management with GAs [J]. Information science, 2007, 177(11): 2380-2401.
- [3] Bowman M, Briand J C, Labiche Y. Multi-objective genetic algorithm to support class responsibility assignment[C]//Proc of IEEE international conference on software maintenance. [s. l.]: [s. n.], 2007: 124-133.
- [4] Zhang Y, Harman M, Mansouri S A. The multi-objective next release problem[C]//Proc of Conference on genetic and evo-

短优化周期,提高优化效率。

4 结束语

人工蜂群算法作为一种新型的群智能优化算法,已有的研究表明其在优化精度及速度上比遗传算法、粒子群算法等更具优势。但标准的蜂群算法仍受早熟收敛的困扰。文中通过修改觅食蜂和观察蜂的搜索策略,提升了算法的收敛精度和收敛速度,同时保持了算法的简洁。收敛速度和精度的提高对全局类智能算法十分重要,具有很高的工程实用价值。

参考文献:

- [1] Karaboga D. An idea based on honeybee swarm for numerical optimization [R]. Turkey: Erciyes University, 2005.
 - [2] Karaboga D, Akay B. A comparative study of artificial bee colony algorithm [J]. Applied mathematics and computation, 2009, 214(1): 108-132.
 - [3] Ye Zhiwei, Zeng Mengdi. Image enhancement based on artificial bee colony algorithm and fuzzy set [C]//Proc of international symposium on information engineering and electronic commerce. Hubei, China: [s. n.], 2011: 127-130.
 - [4] 肖永豪,余卫宇. 基于蜂群算法的图像边缘检测[J]. 计算机应用研究, 2010, 27(7): 2748-2750.
 - [5] 何志明,马 苗. 基于灰色关联分析和人工蜂群算法的图像匹配方法[J]. 计算机技术与发展, 2010, 20(10): 78-81.
 - [6] 胡中华,赵 敏. 基于人工蜂群算法的 TSP 仿真[J]. 北京理工大学学报, 2009, 29(11): 978-982.
 - [7] Omakar S N, Senthilnath J, Khandelwal R, et al. Artificial bee colony for multi-objective design optimization of composite structures [J]. Applied soft computing, 2011, 11(1): 489-499.
 - [8] 刘 勇,马 良. 函数优化的蜂群算法[J]. 控制与决策, 2012, 27(6): 886-890.
 - [9] 李林菲,马 苗. 基于 ABC 算法的逻辑推理题快速求解方法[J]. 计算机技术与发展, 2011, 21(6): 125-127.
 - [10] Wang Hsin-Chih, Wang Yucheng. Performance comparisons of genetic algorithm and artificial bee colony algorithm applications for localization in wireless sensor networks [C]//Proc of 2010 international conference on system science and engineering. Wuhan, China: [s. n.], 2010: 469-474.
 - [11] 吴 斌,钱存华,崔志勇. 具有社会认知策略的人工蜂群算法研究 [C]//第 24 届中国控制与决策会议. 太原: 出版者不详, 2012: 2681-2684.
 - [12] 王 辉. 改进的蜂群算法[J]. 计算机工程与设计, 2011, 32(11): 3869-3872.
 - [13] Tsai P W, Pan J S, Liao B Y. Enhanced artificial bee colony optimization [J]. International journal of innovative computing, information and control, 2009 (12): 5081-5092.
 - [14] 罗 钧,李 研. 具有混沌搜索策略的蜂群优化算法[J]. 控制与决策, 2010, 25(12): 1913-1916.
 - [15] Gao Weifeng, Liu Sanyang, Huang Lingling. A novel artificial bee colony algorithm based on modified search equation and orthogonal learning [J]. IEEE journals & magazines, 2013, 43(3): 1011-1024.
 - [16] Karaboga D, Gorkemli B. A quick artificial bee colony - qABC - algorithm for optimization problems [C]//Proc of 2012 international symposium on innovations in intelligent systems and applications. [s. l.]: [s. n.], 2012: 1-5.
-
- (上接第 24 页)
- lutionary computation. [s. l.]: [s. n.], 2007: 1129-1137.
 - [5] Cao Y, Hu C, Li L. Search-based multi-paths test data generation for structure-oriented testing [C]//Proc of ACM/SIG-GEVO summit on genetic and evolutionary computation. [s. l.]: [s. n.], 2009: 25-32.
 - [6] Zhang Y, Gong D W. Evolutionary generation of test data for multiple paths coverage with faults detection [C]//Proc of IEEE fifth international conference. [s. l.]: [s. n.], 2010: 406-410.
 - [7] Ghiduk A S, Harrold M J, Girgis M R. Using genetic algorithms to aid test-data generation for data-flow coverage [C]//Proc of 14th Asia-Pacific software engineering conference (APSEC). [s. l.]: [s. n.], 2007: 41-48.
 - [8] 马光娟. 基于新模型的多目标遗传算法 [D]. 西安: 西安电子科技大学, 2008.
 - [9] White L J, Cohen E I. A domain strategy for computer program testing [J]. IEEE transactions on software engineering, 1980, 6(3): 247-257.
 - [10] Korel B. Dynamic method of software test data generation [J]. Journal of software testing, verification and reliability, 1992, 2(4): 203-213.
 - [11] Deb K, Goyal M. A combined genetic adaptive search (GeneAS) for engineering design [J]. Computer science and informatics, 1996, 26(4): 30-45.
 - [12] 郑金华,蒋 浩,邝 达,等. 用擂台赛法则构造多目标 Pareto 最优解集的方法 [J]. 软件学报, 2007, 18(6): 1287-1297.

NSGA-II面向多目标测试数据模型的生成

作者：[周文晶](#)，[王占刚](#)，[ZHOU Wen-jing](#)，[WANG Zhan-gang](#)
作者单位：[天津工业大学 计算机科学与技术学院, 天津, 300387](#)
刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2014(4)

本文链接：http://d.wanfangdata.com.cn/Periodical_wjfz201404005.aspx