基于否定选择遗传算法的路径覆盖测试数据生成

夏春艳[1],张 岩[1],万 里[2],宋 妍[1],肖 楠[1],郭 冰[1]
(1. 牡丹江师范学院计算机与信息技术学院,黑龙江牡丹江 157012;2. 天津大学智能与计算学部,天津 300350)

摘 要: 路径覆盖是软件测试领域重要的测试方法之一. 在搜索空间中,找到一组测试数据满足路径覆盖是一个具有挑战性的问题. 因此,自动生成测试数据是软件测试的关键问题. 文中提出一种基于否定选择遗传算法的路径覆盖测试数据生成方法,将否定选择策略融入遗传算法,动态优化遗传算法的种群数据,自动生成覆盖目标路径的测试数据. 多个基准程序和工业程序的实验结果表明,与随机方法和遗传算法比较,文中方法能够提高路径覆盖率,减少冗余测试数据的生成.

关键词: 软件测试; 遗传算法; 否定选择; 路径覆盖; 测试数据
中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2019)12-2630-09
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2019.12.024

Test Data Generation of Path Coverage Based on Negative Selection Genetic Algorithm

XIA Chun-yan[1], ZHANG Yan[1], WAN Li[2], SONG Yan[1], XIAO Nan[1], GUO Bing[1]
(1. *School of Computer and Information Technology, Mudanjiang Normal University, Mudanjiang, Heilongjiang* 157012, *China*;
2. *Tianjin University Division of Intelligence and Computing, Tianjin University, Tianjin* 300350, *China*)

Abstract: Path coverage is one of the most important testing methods in the field of software testing. It is a challenging problem to find a set of test data to satisfy the path coverage in the search space. Therefore, automatically generating test data is a key issue in software testing. In this paper, a generation method of test data based on the negative selection genetic algorithm is proposed. The negative selection strategy is integrated into the genetic algorithm, and the population data of the genetic algorithm is dynamically optimized, and the test data covering the target path is automatically generated. The experimental results show that compared with the random method and the genetic algorithm, the proposed method can improve the path coverage and reduce the generation of redundant test data.

Key words: software test; genetic algorithm; negative selection; path coverage; test data

1 引言

软件测试是软件生命周期的重要组成部分,是保证软件质量、提高软件可靠性的重要手段[1]. 研究资料表明,软件测试过程大约占软件开发总成本的一半以上[2]. 在结构测试中,路径覆盖是最有效的测试方法,可以生成满足覆盖充分性准则的测试数据[3]. 路径覆盖指尽可能生成覆盖所有路径的测试数据,是一个及其复杂、费力和耗时的过程[4]. 单锦辉等人[5]认为,许多软件测试问题都可以归结为路径覆盖测试数据生成问题. 因此,自动生成覆盖目标路径的测试数据是软件测试的关键任务,是简化软件测试过程的有效方法,能够降低软件测试成本,提高软件测试效率.

自动生成满足覆盖充分性准则的测试数据是软件测试的一个基本问题. 许多研究者针对这一问题提出了不同的测试数据生成技术,以提高被测程序的覆盖率,降低测试成本. 在诸多的测试方法中,遗传算法的应用最为广泛,如回归测试[6,7]、变异测试[8~10]和并行测试[11,12]中均有较好的应用. 其中,大多数的研究为测试数据的自动生成问题,如文献[13~16]研究的都是基

收稿日期: 2018-08-20;修回日期:2018-10-18;责任编辑:张龑翔
基金项目: 黑龙江省教育厅基本科研业务费(No. 1353MSYYB005);牡丹江师范学院科学技术研究(No. YB2018004, No. GP201602);黑龙江省自然科学基金(No. F2016039);黑龙江省教育厅(No. 1353ZD003, No. 1353MSYYB007, No. 1353MSYQN006);牡丹江市科技计划(No. Z2016s0027, No. Z2018s073);牡丹江师范学院大学生创新训练项目(No. 201910233006)

于遗传算法生成测试数据的方法. 遗传算法的缺点是容易过早收敛,优点是覆盖率高和时间消耗低. 在生成最优测试数据的过程中,如果能够避免遗传算法过早收敛,那么遗传算法可以更有效的提高程序的覆盖率,减少测试时间.

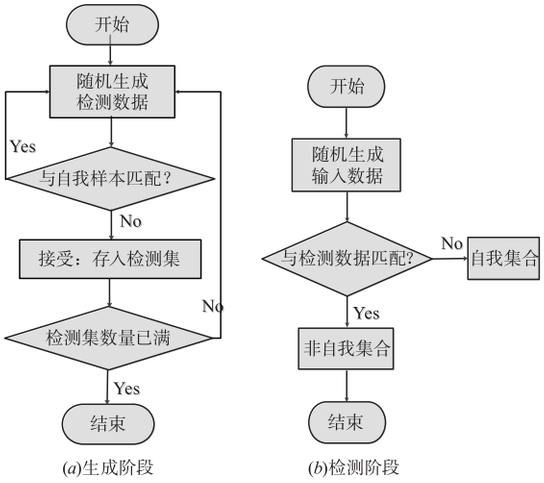
2　基本知识

为了便于说明本文工作,这里先介绍否定选择算法和遗传算法.

2.1　否定选择算法

否定选择算法(Negative Selection Algorithm,NSA)是人工免疫系统的最重要的方法,是计算智能模型的一个分支[17]. NSA 的主要目标是区分自我样本和非自我样本,而只有自我样本是可用的.

NSA 的基本思想是在搜索空间中产生若干检测数据,然后应用检测数据对新数据进行自我集合或非自我集合分类. NSA 分为两个阶段:生成阶段和检测阶段. 在生成阶段,采用随机过程生成检测数据,并对此过程进行监督. 与自我样本匹配的候选数据被丢弃,不匹配的候选数据被存储到检测集. 当生成足够数量的检测数据时,生成阶段终止. 在检测阶段,将生成阶段产生的检测集用于检查输入数据是否对应自我样本或非自我样本. 如果输入数据与检测数据匹配,则被划分到非自我集合;否则,存入自我集合. 图 1 为 NSA 流程图,生成阶段如图 1(*a*)所示,检测阶段如图 1(*b*)所示.



(*a*)生成阶段　　(*b*)检测阶段

图1　NSA流程图

2.2　遗传算法

遗传算法(Genetic Algorithm,GA)是模拟生物在自然环境中的遗传和进化过程而形成的一种自适应全局优化概率搜索算法[18]. GA 的主要目标是在搜索空间中寻求问题的最优解或近似最优解.

GA 的基本思想是将要解决的问题模拟成一个生物进化的过程. 在搜索空间随机产生若干输入数据,并对其编码代表问题可能潜在的解集. 通过选择、交叉和变异操作产生下一代个体,计算个体的适应度,淘汰适应度低的个体,增加适应度高的个体,进化生成适应度最高的个体,解码为目标函数的最优解或近似最优解. GA 流程图如图 2 所示.

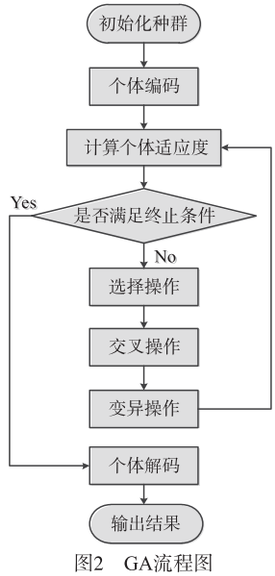


图2　GA流程图

3　进化方法

众所周知,路径测试是结构测试的主要策略. 解决路径测试的一个基本方法是在搜索空间中尽可能找到覆盖目标路径的测试数据. 本文提出了否定选择遗传算法(Negative Selection Genetic Algorithm,NSGA),用于进化生成测试数据. NSGA 是将否定选择策略融入遗传算法,对遗传算法的种群数据进行动态优化,进而使种群数据具有多样性,避免遗传算法过早收敛,在保证生成的测试数据具有较高覆盖率的基础上,达到以较少的测试数据覆盖较多的测试路径的目的,减少冗余测试数据的生成,实现目标函数进化优化的过程.

本文方法的主要实现过程分为两个阶段. 第一阶段初始化种群,依据 NSA 的生成策略,有监督的生成 NSGA 的初始种群. 该阶段的设计思想是随机产生输入数据,并与初始种群中已生成的数据进行匹配,如果匹配,则为冗余数据,存入非初始种群数据集;否则,存入初始种群数据集. 重复上述过程,直到生成初始种群的全部数据为止. 第二阶段进化生成覆盖目标路径的测试数据. 该阶段的设计思想是在迭代的过程中,依据 NSA 的检测策略,动态更新 NSGA 的进化种群数据. 首先,对初始种群数据进行个体编码,计算个体的适应度,将生成的覆盖目标路径的测试数据存入检测集. 然后,

进行选择、交叉和变异操作,重新计算个体的适应度,依据 NSA 的检测策略动态更新种群数据,保证进化数据多样化,避免算法过早收敛. 需要指出的是,在种群数据的更新过程中,主要是对进化生成的适应度较低的个体进行更新,同时补充种群数据使其达到初始种群规模,并计算新个体的适应度. 最后,重复迭代过程,直到满足终止条件(生成覆盖全部目标路径的测试数据或者达到最大迭代次数)为止,将生成的覆盖目标路径的测试数据进行个体解码输出. 图 3 给出了 NSGA 生成覆盖目标路径测试数据的流程图.

在 NSGA 的第一个阶段,关键技术是将 NSA 的生成策略应用在 GA 的种群初始化过程中,具体问题体现在随机产生的输入数据与初始种群数据如何进行匹配. 在 NSGA 的第二阶段,关键技术是将 NSA 的检测策略应用在 GA 的种群数据更新过程中,具体问题为如何进行种群数据更新. 在应用 NSGA 生成测试数据的过程中,适应度函数的设计也至关重要. 下面详细阐述这三方面内容.

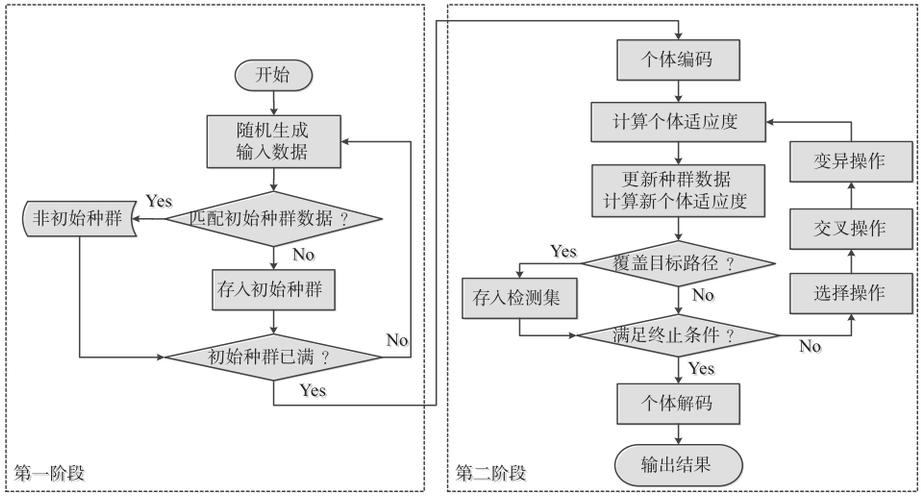


图3 NSGA流程图

3.1 数据匹配

被测程序记为 G,输入搜索空间为 S,个体 x 代表输入数据,以二进制编码表示,即 $x \in S$.

数据匹配方法为计算采样数据与检测集中检测数据的海明距离,如果小于阈值 ε,则为非自我数据;否则,为自我数据. 这里,海明距离为个体不同二进制编码的个数,计算公式为

$$d_i(x) = x \oplus x_i, i = 1,2,\cdots,n \tag{1}$$

其中,n 为检测集含有检测数据的个数,x 表示采样数据,x_i 表示检测集的第 i 个检测数据,$d_i(x)$ 表示 x 与 x_i 的海明距离.

在生成初始种群数据时,测试数据为随机产生的输入数据,检测数据为初始种群中已生成的初始数据. 因此,由式(1)知,如果 $d_i(x) \geq \varepsilon, i = 1,2,\cdots,n$,表明输入个体与初始种群中的每个个体的海明距离都不小于阈值 ε,存入初始种群. 否则,存入非初始种群.

匹配阈值 ε 决定了采样数据与检测集数据之间的相似程度,选择合适的阈值直接影响 NSGA 的性能. ε 太大,检测集覆盖的与自我空间不相邻的非自我空间就会过大. ε 太小,可能又不会从有价值的自我空间中生成合理大小的检测集. 一般情况,通过实验数据及算法的最佳性能确定 ε.

以三角形分类程序为例,说明海明距离的计算方法. 假设采样数据 $x = (22,16,7)$,检测集的第 i 个数据 $x_i = (24,20,8)$,则 x 与 x_i 的海明距离 $d_i(x) = 8$,计算过程如图 4 所示.

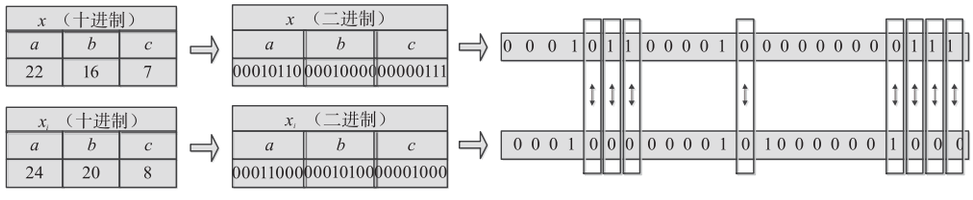


图4 海明距离计算过程

3.2 数据更新

种群数据更新指的是每次迭代后,对生成的适应度较低的种群数据进行更新. 包含两方面内容,一是依据 NSA 的检测策略,检查进化后适应度较低的个体与检测集的检测数据是否匹配,匹配丢弃,不匹配保留. 特别的,更新种群数据用到的匹配方法与初始化种群数据的数据匹配方法相同,计算进化数据与检测集的检测数据的海明距离. 二是如果更新种群数据时,由于移除覆盖目标路径的数据或者丢弃冗余数据产生当前种群含有个体的数量小于初始种群规模的情况,应采用与生成初始种群同样的方法,随机产生输入数据,生成足够数量的种群数据. 需要说明的是,种群初始化与数据更新过程都需要生成种群数据,区别在于第二阶段随机产生的输入数据不仅要与已有种群数据进行匹配,还要与检测集中生成的测试数据进行匹配,同时满足两种匹配条件的数据才能存入当前种群,直到生成满足种群规模的数据为止. 以三角形分类程序为例,说明种群数据更新的过程,如图 5 所示.

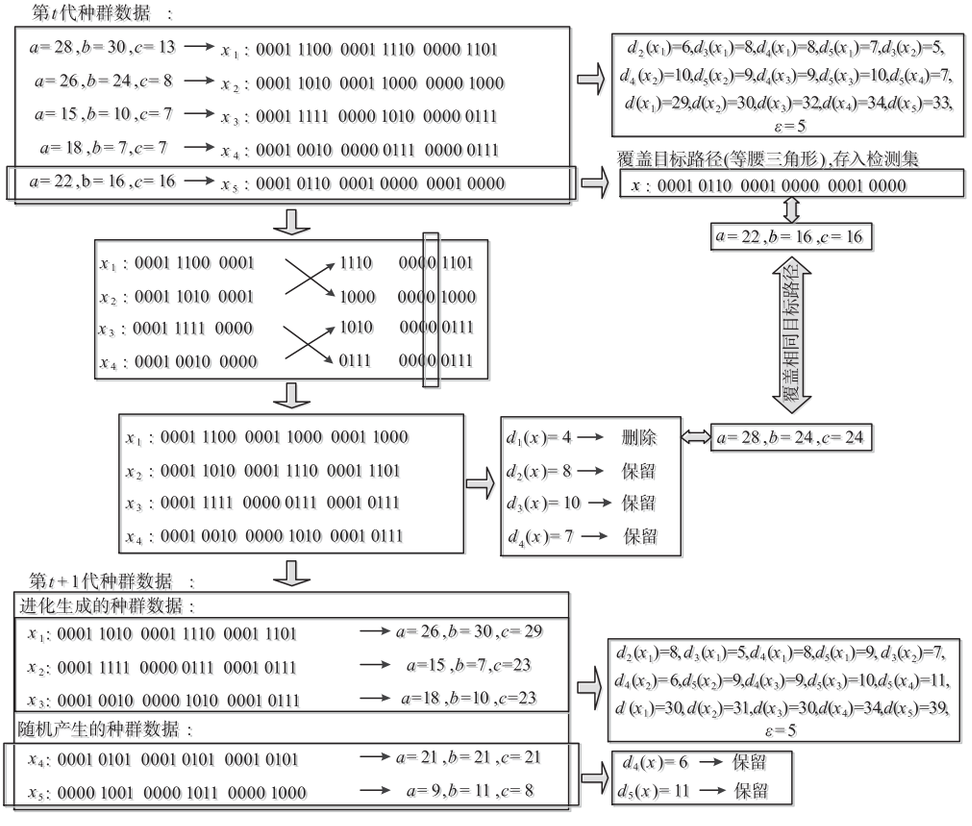


图5　种群数据更新过程

在本例中,设初始种群 $m=5$,个体编码后对应 5 条染色体,阈值 $\varepsilon=5$,这里第 t 代种群数据是依据种群初始化过程产生的,则在第 t 代种群数据生成覆盖目标路径的测试数据前,检测集为空. 首先,计算任意两个个体的相似距离,生成满足初始种群要求的第 t 代种群数据. 其次,计算个体的适应度,将覆盖目标路径的测试数据x_5存入检测集. 再次,将没有覆盖目标路径的个体进行选择、交叉和变异操作,重新计算个体的适应度. 然后,计算适应度低的个体与检测集中个体的相似距离,发现x_1与检测集中 x 的相似距离为 4,小于 ε,将其删除. 此处值得注意的是,丢弃的数据(28,24,24)与检测集中的检测数据(22,16,16)穿越的目标路径相同,说明丢弃的数据为冗余数据,这也证实了本文方法能够动态优化种群数据,保证进化种群的多样性,进而减少了冗余测试数据的生成. 最后,补充种群数据,使用与种群初始化相同的方法生成个体x_4和x_5,其中任意两个个体的相似距离均大于 ε,则得到第 $t+1$ 代种群数据(26,30,29)、(15,7,23)、(18,10,23)、(21,21,21)和(9,11,8).

3.3 适应度函数设计

在采用遗传算法生成路径覆盖的测试数据时,适应度函数体现了自然进化过程的优胜劣汰原则. 因此,适应度函数的设计是求解优化问题的关键.

本文采用分支距离与层接近度结合的方法设计适应度函数. 设个体 x 的层接近度为 approach_level(x),表示 x 的穿越路径与目标路径的接近程度,常见的计算

方法是:x 的穿越路径与目标路径不匹配的节点个数除以目标路径的节点总数. 设个体 x 的分支距离为 branch_distance(x),反映 x 的穿越路径与目标分支的偏离程度,详细内容参见文献[19]. 一般情况,分支距离的值远大于层接近度的值. 因此,为了权衡分支距离与层接近度的大小,并统一为最小化运算,将其规范化为 $1.001^{-\text{branch\_distance}(x)}$,其值越小,个体越优. 因此,个体的适应度函数的计算公式可表示为

$$\text{fit}(x)=\text{approach\_level}(x)+1.001^{-\text{branch\_distance}(x)} \quad (2)$$

4 实验

为了评估所提方法的性能,本文选择多个基准程序和工业程序进行实验. 实验条件:Windows 7 操作系统,计算机主频 2.80GHz,内存 2GB,所有程序均用 java 语言编写,在 eclipse 环境下运行.

4.1 基准程序

基准程序如表 1 所示,包含选择结构、循环结构以及复杂的嵌套结构,含有算术运算符、关系运算符和逻辑运算符,还涵盖了整型、浮点型、字符型以及字符串等数据类型,这些基准程序都被广泛的应用于软件测试研究领域.

表 1 基准程序

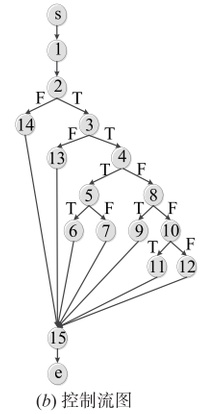
序号	程序描述	代码行数	目标路径数
P1	三角形分类	17	7
P2	计算 x 除以 y 的余数	10	2
P3	最大值最小值	26	4
P4	冒泡排序	32	4
P5	学生成绩	18	5
P6	查找第二天的日期	97	13
P7	查找两个日期之间的天数	118	25

三角形分类程序是软件测试领域中典型的基准实验程序,通过 3 个输入数据判定三角形类型. 如果 3 个输入变量是 16 位的二进制,则搜索空间需要执行 $2^{16}\times 2^{16}\times 2^{16}=2^{48}$ 个测试用例. 考虑搜索空间内的测试数据,仅有较少的输入数据能够满足等边三角形的路径覆盖. 比如,在上述搜索空间中,有 $2^{16}-1$ 个等边三角形,其概率仅为 $(2^{16}-1)/2^{48}\approx 1/2^{32}$. 因此,本文选择三角形分类程序为例,说明实验过程,其程序源代码和控制流图如图 6 所示.

三角形分类程序实验参数设置,初始种群为 100,交叉概率为 0.8,变异概率为 0.15,最大迭代次数为 200. 下面应用随机方法(Random)、遗传算法(GA)和本文提出的方法(NSGA)生成覆盖目标路径的测试用例. 每种算法运行 50 次,取平均值作为实验结果. 当实验终止时,统计生成覆盖各个目标路径的测试用例数

```
s  void Triangle(int a, int b, int c)
1  {string type;
2  if (a>0 && b>0 && c>0)
3    if (a<b+c && b<a+c && c<a+b)
4      if (a==b)
5        if (b==c)
6          type="Equilateral";
7        else type="Isosceles";
8      else if (a==c)
9        type="Isosceles";
10     else if (b==c)
11       type="Isosceles";
12     else type="Scalene";
13   else type="Not Triangle";
14 else type="Not Triangle";
15 return type;
e  }
```

(a) 源代码



(b) 控制流图

图6 三角形分类程序

量;当生成覆盖所有目标路径的测试用例时,记录迭代次数,统计生成的测试用例总数量,核算覆盖率,实验结果相关数据如表 2 所示.

表 2 三角形分类程序实验结果

	Random	GA	NSGA
等边三角形	0	1	4
等腰三角形	213	268	381
一般三角形	9731	9212	8861
非三角形	10056	10519	10754
迭代次数	200	163	5
测试用例总数	20000	14853	419
覆盖率	75%	100%	100%

表 2 实验结果表明,GA 和 NSGA 能够生成覆盖率为 100% 的测试用例,而 Random 没有生成覆盖等边三角形路径的测试用例,覆盖率仅为 75%. 根据表 2 实验结果,作三种算法生成覆盖所有目标路径时的测试用例总数和迭代次数的对比直方图,如图 7 和图 8 所示. 明显可见,Random 生成的测试用例总数最多,迭代次数最大. GA 进化到 163 代时才生成覆盖所有目标路径

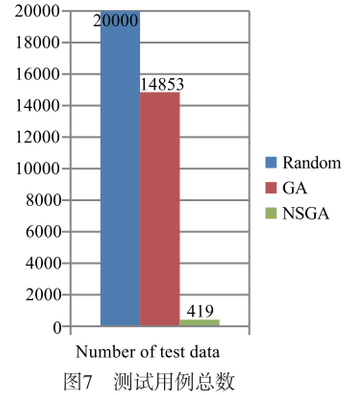


图7 测试用例总数

的测试用例. NSGA 生成覆盖所有目标路径的测试用例总数仅为419,而且只需要迭代 5 次,减少了冗余测试数据的生成,降低了迭代次数. 因此,NSGA 具有更好的有效性.

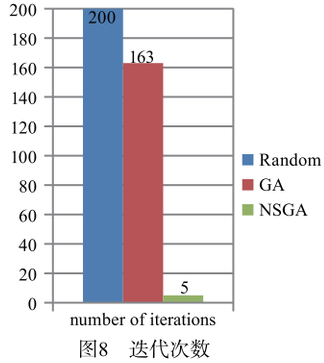


图8　迭代次数

依托三角形分类程序实验参数设置以及实验过程,对表 1 所示的基准程序进行实验,所得的相关实验结果如表 3 所示.

表 3 实验结果表明,在基准程序实验中,NSGA 生成的 测试用例数量和迭代次数均最少,且达到了100% 的覆盖率. 为了更明显的表明 NSGA 的有效性,做出上述三种方法的迭代次数和测试用例数量的折线图,如图 9 和图 10 所示. 显而易见,NSGA 在提高了路径覆盖率的基础上,减少了冗余测试数据的生成,降低了迭代次数. 进而说明,NSGA 具有更好的有效性.

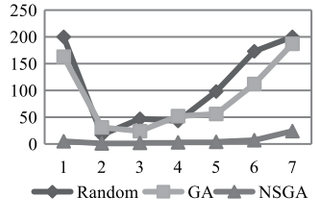


图9　基准程序迭代次数

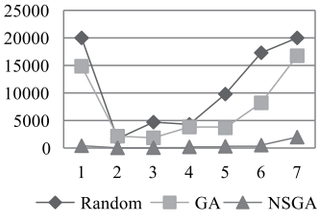


图10　基准程序测试用例数量

表 3　基准程序实验结果

序号	Random 方法			GA 方法			NSGA 方法		
	迭代次数	测试用例数量	覆盖率	迭代次数	测试用例数量	覆盖率	迭代次数	测试用例数量	覆盖率
P1	200	20000	75%	163	14853	100%	5	419	100%
P2	17	1700	100%	31	2156	100%	1	36	100%
P3	47	4700	100%	24	1860	100%	2	64	100%
P4	43	4300	100%	52	3819	100%	3	210	100%
P5	98	9800	100%	56	3670	100%	4	323	100%
P6	173	17300	100%	112	8210	100%	7	531	100%
P7	200	20000	63%	187	16756	100%	24	1985	100%

4.2　工业程序

工业程序,选择的是 Replace、Space、Gzip、Sed 和 Flex,这些工业程序均被广泛的应用于软件测试研究领域. 对于每个被测程序选择部分子函数进行实验,表 4 分别列出了被测程序的基本信息.

工业程序实验参数设置,初始种群为 100,交叉概率为 0. 8,变异概率为 0. 15,最大迭代次数为 500. 工业程序的相关实验结果如表 5 所示.

表 5 实验结果亦表明,NSGA 生成的测试用例数量和迭代次数均最少,路径覆盖率最高,达到了减少冗余测试数据的生成和降低迭代次数的目的. 因此,本文提出的 NSGA 具有更好的有效性. 图 11 和图 12 给出了 Random、GA 和 NSGA 三种方法的迭代次数和测试用例数量的折线图,由此可以更好的说明,本文方法在保证覆盖率的基础上,比其他两种方法更有效.

表 4　工业实验程序

程序	程序代码数	函数个数	函数代码数	目标路径数
Replace	550	3	98	10
Space	6199	3	195	15
Gzip	7329	2	239	47
Sed	8063	2	367	125
Flex	13225	2	1103	233

表 5 工业程序实验结果

程序	Random 方法			GA 方法			NSGA 方法		
	迭代次数	测试用例数量	覆盖率	迭代次数	测试用例数量	覆盖率	迭代次数	测试用例数量	覆盖率
Replace	56	5600	100%	28	1607	100%	4	232	100%
Space	200	20000	73%	123	9564	100%	32	2678	100%
Gzip	163	16300	100%	108	8214	100%	24	1956	100%
Sed	200	20000	58%	200	15614	79%	200	13564	97%
Flex	200	20000	42%	200	17862	57%	200	14521	86%

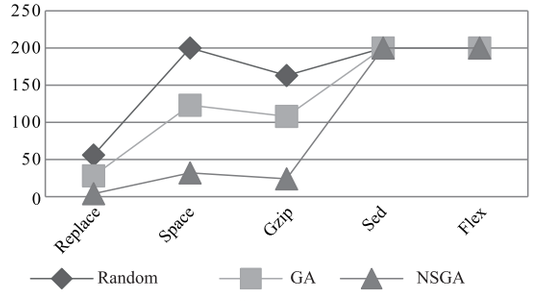


图11 工业程序迭代次数

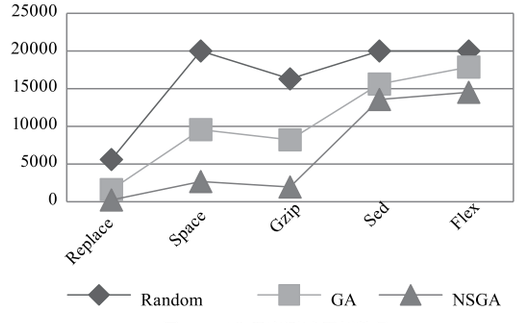


图12 工业程序测试用例数量

4.3 实验结果的统计分析

为了更好的验证实验结果的可靠性,这里借助统计分析对实验结果进行显著性检验. 本文采用的假设检验方法为 Z 检验,设置显著性水平 $\alpha=0.01$,则 $-Z_\alpha=-2.325$. 检验的假设是对比方法之间的平均生成测试用例的数量没有显著差异.

设随机变量 X 表示生成的测试用例数量. 在实际问题中,这些随机变量是由大量的相互独立的随机因素的综合影响所形成的,而其中每一个别因素在总的影响中所起的作用都是微小的,因此,随机变量 X 近似地服从正态分布 $X\sim N(\mu,\sigma^2)$. 本文比较每种方法相应随机变量均值 μ 的大小. 其值越小,表明该方法生成的冗余测试数据越少,该方法的有效性也就越高.

以三角形分类程序为例,给出 NSGA、GA 和 Random 三种方法实验结果的对比过程. 考虑到样本方差是总体方差的无偏估计,用样本方差的值作为总体方差的估计值. 根据三角形分类程序的实验结果得到如下数据:样本容量 $n_1=n_2=n_3=50$,样本均值 $\bar{X}_1=419$,$\bar{X}_2=14853$,$\bar{X}_3=20000$,样本方差 $\sigma_1^2=329600$,$\sigma_2^2=29576995$,$\sigma_3^2=0$.

第 1 步,建立原假设 $H_0:\mu_1\geq\mu_2$ 和 $H_0:\mu_1\geq\mu_3$,以及备择假设 $H_1:\mu_1<\mu_2$ 和 $H_1:\mu_1<\mu_3$.

第 2 步,构造统计量 $Z_1=\dfrac{\bar{X}_1-\bar{X}_2}{\sqrt{\dfrac{\sigma_1^2}{n_1}+\dfrac{\sigma_2^2}{n_2}}}$ 和 $Z_2=\dfrac{\bar{X}_1-\bar{X}_3}{\sqrt{\dfrac{\sigma_1^2}{n_1}+\dfrac{\sigma_3^2}{n_3}}}$.

第 3 步,给出拒绝域 $Z_1\leq -Z_\alpha$ 和 $Z_2\leq -Z_\alpha$.

第 4 步,计算统计量的值,即

$$Z_1=\frac{\bar{X}_1-\bar{X}_2}{\sqrt{\dfrac{\sigma_1^2}{n_1}+\dfrac{\sigma_2^2}{n_2}}}=\frac{419-14853}{\sqrt{\dfrac{329600}{50}+\dfrac{29576995}{50}}}\approx -18.66,$$

$$Z_2=\frac{\bar{X}_1-\bar{X}_3}{\sqrt{\dfrac{\sigma_1^2}{n_1}+\dfrac{\sigma_3^2}{n_3}}}=\frac{419-20000}{\sqrt{\dfrac{329600}{50}+\dfrac{0}{50}}}\approx -241.17.$$

第 5 步,给出结论:$Z_1\leq -Z_\alpha$ 和 $Z_2\leq -Z_\alpha$. 因此,拒绝原假设 $H_0:\mu_1\geq\mu_2$ 和 $H_0:\mu_1\geq\mu_3$,接受备择假设 $H_1:\mu_1<\mu_2$ 和 $H_1:\mu_1<\mu_3$. 因此,NSGA 与 GA 和 Random 两种方法的对比结果有显著差异. 这说明,NSGA 生成测试用例的数量明显少于 GA 和 Random.

依据上述实验结果的对比过程,对基准程序和工业程序的实验结果进行假设检验,检验结果如表 6 所示. 由表 6 可以看出,NSGA 与 GA 和 Random 两种方法的对比结果有显著差异,即本文方法生成测试用例的数量明显少于其他两种方法.

表 6 检验结果

被测程序	统计量的值		检验结果
	Z_1	Z_2	
P1	-18.66	-241.17	
P2	-12.95	-10.68	
P3	-11.75	-13.12	
P4	-11.33	-11.80	
P5	-10.27	-14.10	
P6	-10.63	-78.66	接受$H_1:\mu_1<\mu_2$
P7	-45.48	-113.39	接受$H_1:\mu_1<\mu_3$
Replace	-9.64	-11.25	
Space	-8.93	-76.60	
Gzip	-8.36	-40.52	
Sed	-3.42	-11.13	
Flex	-6.20	-12.81	

5 总结

本文提出一种基于否定选择遗传算法的路径覆盖测试数据自动生成方法,将否定选择策略融入遗传算法,动态优化遗传算法的种群数据,解决了遗传算法过早收敛的问题,进化生成覆盖目标路径的测试数据. 多个基准程序和工业程序的实验结果以及统计分析表明,与随机方法和遗传算法比较,文中方法能够提高路径覆盖率,减少冗余测试数据的生成,降低迭代次数,具有更好的有效性.

后续工作,执行更复杂的并行程序来衡量本文方法的有效性. 进一步,研究测试数据优先级技术,对生成的测试数据按照优先级排序,更好的提高软件测试的效率.

致谢　在此,向对本文工作给予支持和建议的评审专家表示衷心的感谢.

参考文献

[1] Mei Jia, Wang Sheng-yuan. An improved genetic algorithm for test cases generation oriented paths[J]. Chinese Journal of Electronics, 2014, 23(3): 494 - 498.

[2] Omur Sa-hin, Bahriye Akay. Comparisons of metaheuristic algorithms and fitness functions on software test data generation[J]. Applied Soft Computing, 2016, 49: 1202 - 1214.

[3] Ghiduk A S. Automatic generation of basis test paths using variable length genetic algorithm[J]. Information Processing Letters, 2014, 114(6): 304 - 316.

[4] 夏春艳,张岩,宋丽. 基于节点概率的路径覆盖测试数据进化生成[J]. 软件学报,2016,27(4):802 - 813.
XIA Chun-yan, ZHANG Yan, SONG Li. Evolutionary generation of test data for paths coverage based on node probability[J]. Journal of Software, 2016, 27(4): 802 - 813. (in Chinese)

[5] 单锦辉,王戟,齐治昌. 面向路径的测试数据自动生成方法述评[J]. 电子学报,2004,32(1):109 - 113.
SHAN Jin-hui, WANG Ji, QI Zhi-chang. Survey on pathwise auto-matic generation of test data[J]. Acta Electronica Sinica, 2004, 32(1): 109 - 113. (in Chinese)

[6] 吴川,巩敦卫,姚香娟. 基于分支覆盖的回归测试路径选择[J]. 软件学报,2016,27(4):839 - 854.
WU Chuan, GONG Dun-wei, YAO Xiang-juan. Selection of paths for regression testing based on branch coverage [J]. Journal of Software, 2016, 27(4): 839 - 854. (in Chinese)

[7] 杨波,吴际,刘超. 基于变量影响分析与数据变异的回归测试用例生成[J]. 计算机学报,2016,39(11):2372 - 2387.
YANG Bo, WU Ji, LIU Chao. An approach of regression test case generation based on variable impact analysis and data mutation[J]. Chinese Journal of Computers, 2016, 39(11): 2372 - 2387. (in Chinese)

[8] Chang-ai Sun, Fei-fei Xue, Huai Liu, Xiang-yu Zhang. A path-aware approach to mutant reduction in mutation testing[J]. Information and Software Technology, 2017, 81: 65 - 81.

[9] Pedro Reales Mateo, Macario Polo Usaola. Reducing mutation costs through uncovered mutants[J]. Software Testing, Verification and Reliability, 2015, 25: 464 - 489.

[10] 张功杰,巩敦卫,姚香娟. 基于统计占优分析的变异测试[J]. 软件学报,2015,26(10):2504 - 2520.
ZHANG Gong-jie, GONG Dun-wei, YAO Xiang-juan. Mutation testing based on statistical dominance analysis [J]. Journal of Software, 2015, 26(10): 2504 - 2520. (in Chinese)

[11] Gwan Hwan Hwang, Heng-yi Lin, Shao-yan Lin, Chesheng Lin. Statement-coverage testing for concurrent programs in reachability testing[J]. Journal of Information Science and Engineering, 2014, 30: 1095 - 1113.

[12] Tian Tian, Dun-wei Gong. Test data generation for path coverage of message-passing parallel programs based on co-evolutionary genetic algorithms[J]. Automated Software Engineering, 2016, 23: 469 - 500.

[13] Ali Shahbazi, James Miller. Black-box string test case generation through a multi-objective optimization[J]. IEEE Transactions on Software Engineering, 2016, 42(4): 361 - 378.

[14] Yeresime Suresh, Santanu Ku Rath. A genetic algorithm based approach for test data generation in basis path testing[J]. The International Journal of Soft Computing and

Software Engineering,2013,3(3):326 – 332.

[15] 张岩,巩敦卫. 基于搜索空间自动缩减的路径覆盖测试数据进化生成[J]. 电子学报,2012,40(5):1011 – 1016.
ZHANG Yan,GONG Dun-wei. Evolutionary generation of test data for path coverage based on automatic reduction of search space[J]. Acta Electronica Sinica,2012,40(5):1011 – 1016. (in Chinese)

[16] 姚香娟,巩敦卫. 基于路径比较的变异测试方法[J]. 电子学报,2012,40(1):103 – 107.
YAO Xiang-juan,GONG Dun-wei. Mutation testing based on comparison of paths[J]. Acta Electronica Sinica,2012,40(1):103 – 107. (in Chinese)

[17] Mohi-Aldeen S M,Mohamad R,Deris S. Application of negative selection algorithm(NSA) for test data generation of path testing[J]. Applied Soft Computing,2016,49:1118 – 1128.

[18] 周明,孙树栋. 遗传算法原理与应用[M]. 北京:国防工业出版社,1999.

[19] 张岩,巩敦卫. 基于稀有数据拓扑的路径覆盖测试数据进化生成方法[J]. 计算机学报,2013,36(12):2429 – 2440.
ZHANG Yan,GONG Dun-wei. Evolutionary generation of test data for paths coverage based on scarce data capturing[J]. Chinese Journal of Computers,2013,36(12):2429 – 2440. (in Chinese)

作者简介

夏春艳 女,1980 年 3 月出生,黑龙江桦川人. 副教授,CCF 会员. 2007 年毕业于长春理工大学计算机应用技术专业,获得硕士学位. 现为牡丹江师范学院教师,主要从事软件工程、数据挖掘等方面的研究工作.
E-mail:xia-chun-yan@163.com

张 岩(通讯作者) 女,1972 年 5 月出生,辽宁本溪人. 教授,CCF 会员. 2012 年毕业于中国矿业大学,获得博士学位. 现为牡丹江师范学院计算机学院院长,主要从事软件工程等方面的研究工作.
E-mail:zhangyan@mdjnu.cn

万 里 男,1994 年 4 月出生,湖北松滋人. 2018 年进入天津大学智能与计算学部. 现为硕士研究生,主要从事人工智能、软件工程等方面的有关研究.
E-mail:1151843852@qq.com

宋 妍 女,1977 年出生,黑龙江牡丹江人. 现为牡丹江师范学院教师,硕士,研究方向为软件工程、大数据技术.

肖 楠 女,1983 年出生,黑龙江牡丹江人. 现为牡丹江师范学院教师,硕士,研究方向为基于搜索的软件工程.

郭 冰 女,2000 年出生,河北省唐山市人. 现为牡丹江师范学院学生,本科. 研究方向为基于搜索的软件工程.