

文章编号: 2096-1618(2019)06-0595-05

# 基于改进的人工鱼群算法的软件测试数据自动生成算法

余嘉纯<sup>1</sup>, 刘 涛<sup>1</sup>, 梅倬玮<sup>1</sup>, 胡诗朋<sup>1</sup>, 李 静<sup>2</sup>, 徐自励<sup>2</sup>, 王国强<sup>2</sup>

(1. 成都信息工程大学电子工程学院, 四川 成都 610225; 2. 中国民航局第二研究所, 四川 成都 610041)

**摘要:**针对软件测试领域的测试数据自动生成问题,提出一种改进人工鱼群算法(AFSA)的方案。在对人工鱼群算法的觅食行为、群聚行为、追尾行为中引入动态步长的观点,让迭代次数和效率得到进一步的优化。用改进的人工鱼群算法解决直角三角形判别程序的测试数据生成问题,结果表明在不同数量的种群下改进的人工鱼群算法的收敛速度快,求解精度高,优于传统的遗传算法(GA)、粒子群优化算法(PSO)等方法。

**关键词:**人工鱼群算法;软件测试;测试数据;路径导向

**中图分类号:**TP311.55

**文献标志码:**A

**doi:**10.16836/j.cnki.jcuit.2019.06.006

## 0 引言

软件测试是软件开发中不可或缺的环节,可分为人工测试和自动测试,也可分为静态测试和动态测试。在实际应用中,动态测试应用较广,通过运行软件,动态测试检查软件的动态结果,确认结果的正确性。动态测试由测试软件和测试数据构成。测试数据的设计是指从数据域内找到一系列输入数据来覆盖相应的软件路径。测试数据的质量直接影响软件的开发周期、成本、质量和可靠性等。在动态测试中,设计测试数据的工作量很大。传统的测试数据设计方法无法满足测试覆盖的需求。因此,测试数据的自动生成方法成了研究热点<sup>[1]</sup>。

近年来,软件的自动化测试技术得到了深入的研究,提出一系列的测试数据的自动生成方法,特别是进化算法得到广泛的应用。其中,遗传算法是基于自然选择和群体遗传机理的搜索算法,通过预设的目标函数对每一个个体进行评估,基于一个适度值选择产生下一代的个体,在经过交叉和变异算子进行在组合生成新一代,这样逐步朝着最优解的方向进化。粒子群算法是模拟鸟群蜂群的觅食行为的一种算法,通过随机解出发,通过迭代寻找当前搜索到的全局最优解,是通过群体中个体之间的协作和信息共享寻找最优解的算法。人工鱼群算法作为一种新型的群体智能算法,在求解旅行商问题、路径优化问题、复杂函数优化问题等方面得到有效的应用。以测试数据的自动生成成为研究重点,提出了一种基于改进的人工鱼群算法的测试数据生成方法。

## 1 相关研究

测试数据自动生成方法可分为:随机测试数据生成、静态测试数据生成和动态测试数据生成。

随机测试数据生成是一种基于概率的简单生成方法,其优势在于适用于所有类型的程序。但是无法生成高质量的测试数据,在代码覆盖率方面表现不佳。即使在小型程序语句覆盖目标上,其他方法比随机测试数据生成的效果更好,在更加复杂的程序上可能会产生更大的问题<sup>[2]</sup>。

静态测试数据通过直接扫描程序代码,提取程序的关键语句并分析其语义,根据预设的安全规则、漏洞特征等检测代码的漏洞。符号执行是其中一种比较经典的方法<sup>[3]</sup>。

动态测试数据生成的核心思想是将程序视函数来处理。文献[2]提出了一种使用局部方向搜索技术的动态方法。因为这存在陷入局部最优的风险,这种方法不能应用在所有类型的函数上。为克服这一局限而采用全局优化技术,如遗传算法<sup>[4]</sup>、蚁群算法等<sup>[5]</sup>。

在动态测试数据生成方法中,将测试数据生成问题转化为函数优化问题,构造所需路径的分支函数,应用改进的人工鱼群算法生成所需路径的测试数据。

## 2 改进的人工鱼群算法

### 2.1 改进的人工鱼群算法

受到鱼群行为的启发,李晓磊等<sup>[6]</sup>提出人工鱼群算法(AFSA),是行为主义的典型应用。AFSA是一种随机搜索算法,模拟鱼类的行为,如觅食、成群结队、跟随和移动,它基于个体行为获得全局最优解。

收稿日期:2019-09-19

基金项目:四川省教育厅科研资助项目(18ZA0111);四川省科技厅-四川省省院省校科技合作研发项目(2018JZ0030)

### 2.1.1 人工鱼的行为

设  $X_i$  是人工鱼  $AF_i$  当前所在位置,  $f(X_i)$  是位置  $X_i$  的食物浓度,  $d(X_i, X_j)$  表示  $X_i$  和  $X_j$  之间的距离, Visual 和  $\delta$  分别代表人工鱼的视觉范围和人工鱼的群体因子, Step 表示人工鱼的移动步长。

(1) 觅食行为(Praying)。发现食物后,人工鱼游到食物更加丰富的地方。人工鱼的当前位置是  $X_i$ , 然后人工鱼随机在其视觉范围内选择一个新的位置  $X'$ , 即:

$$X' \leftarrow X_i + \text{rand} \cdot \text{Visual} \quad (1)$$

如果  $f(X')$  优于  $f(X_i)$ , 那么  $AF$  向  $X'$  游一步, 此时更新  $X_i$  的位置, 即:

$$X_i \leftarrow X_i + \frac{X' - X_i}{d(X_i, X')} \cdot \text{rand} \cdot \text{Step} \quad (2)$$

否则,  $AF_i$  会随机尝试另一个位置, 判断是否有更高食物浓度而游向新的位置。多次尝试后, 如果  $AF_i$  没有游向新的位置, 那么它就游向一个随机的位置, 此时更新位置  $X_i$ , 即:

$$X_i \leftarrow X_i + \text{rand} \cdot \text{Visual} \quad (3)$$

(2) 聚群行为(Swarming)。为分享食物和避免危险, 鱼在游动过程中自然成群。聚群行为定义为:  $AF_i$  在其视觉范围内寻找其他  $AF_c$ , 位置为  $X_c$ 。如果  $f(X_c)$  优于  $f(X_i)$ , 并且  $X_c$  的群体因子小于  $\delta$ , 则  $AF_i$  游向  $X_c$ , 此时更新位置  $X_i$ 。否则实施觅食行为。

$$X_i \leftarrow X_i + \frac{X_c - X_i}{d(X_i, X_c)} \cdot \text{rand} \cdot \text{Step} \quad (4)$$

(3) 追尾行为(Following)。当  $AF_i$  发现食物时, 附近的鱼游过来并获得食物。追尾定义为:  $AF_i$  在其视野范围内找到其他  $AF$ 。计算邻近鱼的  $f(X)$ , 得到最大化的  $f(X)$ , 且位置为  $X_m$ 。如果  $f(X_m)$  优于  $f(X)$ , 并且  $X_m$  的群体因子小于  $\delta$ , 那么  $AF_i$  向  $X_m$  游近一步, 更新位置  $X_i$ 。

$$X_i \leftarrow X_i + \frac{X_m - X_i}{d(X_i, X_m)} \cdot \text{rand} \cdot \text{Step} \quad (5)$$

否则, 实施觅食行为。

(4) 游动(Moving)。鱼群在水中随机的游动。游动定义为:  $AF$  在视觉范围内随机选择一个新位置并向其游一步, 更新位置  $X_i$ 。

$$X_i \leftarrow X_i + \text{rand} \cdot \text{Visual} \quad (6)$$

### 2.1.2 人工鱼群算法(AFSA)的步骤

步骤1: 初始化  $AF$  的数量、视觉范围、 $\delta$ 、尝试次数和迭代次数限制。先随机初始化所有  $AF$  的位置, 即: 初始化  $X = (X_1, X_2, \dots, X_i)$ 。

步骤2: 计算所有  $AF$  初始位置的食物浓度, 获得最佳食物浓度和位置, 并存放在公告板上。

步骤3: 对每个  $AF$ , 都要模拟执行聚群行为和追尾行为, 并且评估结果。每个  $AF$  选择更高的食物浓度位置。  $AF$  的默认行为是觅食行为。

步骤4: 计算所有  $AF$  当前位置的食物浓度, 获得最大食物浓度及相应位置, 更新公告板。

步骤5: 如果迭代次数等于极限, 则输出当前最大食物浓度及相应位置。否则, 转到步骤3。

### 2.1.3 改进的人工鱼群算法

在人工鱼群算法中, 步长是影响迭代效果的一个关键参数。但是固定的参数会导致早期迭代缓慢而后期迭代过快。在改进方法中, 步长是动态变化的, 定义为

$$\text{Step} = \delta \cdot \text{iteration\_times} \quad (7)$$

其中  $\text{iteration\_times}$  为迭代次数,  $\delta$  为比例系数。

动态步长通过迭代次数的增加而发生变化。迭代计算是一个推算过程, 靠逐步逼近最优点获得近似解, 次数越大越接近真实的解, 但是也耗费更长的时间。于是比例系数的值应当分级设置, 令迭代次数较低时能有较大的步长, 而迭代次数较高时能有较小的步长, 提高算法的有效性与精确性。

## 2.2 基于人工鱼群算法生成测试数据

### 2.2.1 测试鱼群算法生成模型

改进的人工鱼群算法生成测试数据的流程如图1所示。该流程由测试环境和改进的人工鱼群算法组成。在测试环境部分, 分析待测试的软件结构, 并用桩信息进行编写。改进人工鱼群算法的迭代计算过程中, 被测试软件作为个体适应度评价的基础。在改进的人工鱼群算法部分, 执行该算法以获得能覆盖目标路径的测试数据。

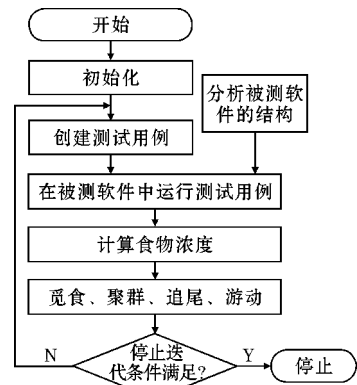


图1 改进的人工鱼群算法生成测试数据模型

### 2.2.2 分支函数与适应度函数

程序  $f$  可以视为一个函数, 即  $f: I \rightarrow P$ , 该函数中,  $I$  是所有可能输入的集合,  $P$  是所有可能的执行路径的集合。如果  $p \in P$  是目标路径, 那么应该产生  $i \in I$  以覆盖路径  $p$ 。为了得到测试数据, 有必要构造一个适应度函数。具体的方法是: 在分支中插入函数, 这些函数用分支谓词计算, 如表1所示。

改进人工鱼群算法自动生成测试数据的实质是通过改进人工鱼群算法迭代计算获得最佳测试数据。实现最佳测试数据的关键设计如下所述。

表 1 分支函数的计算

Element	Value
Boolean	If true then 0 else $k$
$A=B$	If $A=B$ then 0 else $\text{abs}(A-B)+k$
$A \neq B$	If $A \neq B$ then 0 else $k$
$A \neq B$	If $A \neq B$ then 0 else $k$
$A < B$	If $A < B$ then 0 else $(A-B)+k$
$A \leq B$	If $A \leq B$ then 0 else $(A-B)+k$
$A > B$	If $A > B$ then 0 else $(B-A)+k$
$A \geq B$	If $A \geq B$ then 0 else $(B-A)+k$
$A \vee B$	$\text{Min}(\text{fit}(A), \text{fit}(B))$
$A \wedge B$	$\text{fit}(A)+\text{fit}(B)$

插桩和适应度函数设计:为自动测试数据生成,使用函数最小值构造适应度函数。

测试软件有  $m$  个分支,有  $n$  个参数  $x_1, x_2, \dots, x_n$ 。利用函数的最小值,  $m$  分支分别插入分支函数  $f_1, f_2, \dots, f_m$ 。分支函数的累加为适应度函数:

$$\text{fitness} = \sum_{i=1}^m f_i(x_1, x_2, \dots, x_n) \tag{8}$$

函数最小值可以用于直角三角形判别程序:

```
void RightTriangle(int a,int b,int c)
{
    if(a>0 && a<500)
        if(b>0 && b<500)
            if(c>0 && c<500)
                if(a * a + b * b == c * c)
                    printf("It is right triangle.");
}
```

此程序的控制流程图如图 2 所示。程序有 3 个输入和 5 个路径,要生成满足直角三角形要求的测试数据,测试数据应通过路径 {1,3,5,7,9}。

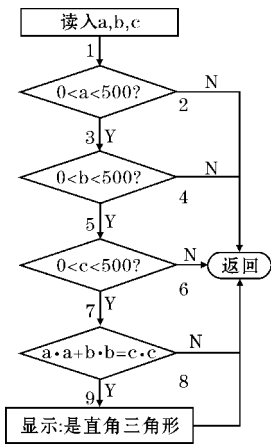


图 2 直角三角形的判定程序流程图

为得到目标测试数据,在直角三角形判别程序中加入了分支函数和适应度函数。具有分支函数和适应度函数的程序如下:

```
int RightTriangle(int a,int b,int c)
{
    int f1,f2,f3,f4,fitness;
    f1=f2=f3=f4=MAX;
    f1=max(0-a ,a-500); //分支函数
    if(a>0 && a<500)
    {
        f2=max(0-b ,b-500); //分支函数
        if(b>0 && b<500)
        {
            f3=max(0-c ,c-500); //分支函数
            if(c>0 && c<500)
            {
                f4=abs(a * a + b * b - c * c); //分支函数
                if(f4 == 0) printf("It's right triangle.");
            }
        }
    }
    if(f1 <0 ) f1=0;
    if(f2<0) f2=0;
    if(f3<0) f3=0;
    if(f4<0) f4=0;
    fitness=f1+f2+f3+f4; //适应度函数
    return fitness;
}
```

重写代码可以看出目标测试数据必须使适应度函数值等于 0,0 是最小的适应度函数。利用改进的人工鱼群算法对测试数据进行优化,使适应度函数达到最小值。设计分支函数和适应度函数是非常重要的,后者将达到目标测试数据的最小值。

2.2.3 方法的执行情况

对于目标测试数据的生成和优化,提出的方法执行步骤描述如下:

- (1)初始化:AF 的随机初始位置产生的,其中每一个位置代表一个测试数据集。
- (2)评价:利用适应度函数对每个 AF 进行评估,更新当前最佳适应度和相应位置。
- (3)行为:每个 AF 朝着更高的适应度执行聚群行为和追尾行为。
- (4)重复步骤(2)~(3)以生成新的测试数据集。
- (5)当测试数据集达到最优,或超过预设迭代的次数后,停止优化过程。

3 实验与讨论

实验中将改进的人工鱼群算法、遗传算法、粒子群优化算法 PSO<sup>[7]</sup>用于解决直角三角形判别程序的测试数据生成问题,并对比 3 种算法的应用效果。目标测

试数据集能构成一个直角三角形的3个输入,输入的范围均为(0,500)。

改进的人工鱼群算法参数设置为:视觉范围设置为3.1,初始步长设置为0.35,群因子设置为0.041,试数设置为3,迭代次数限制设置为500。

遗传算法设计如下:每个个体编码36位。它使用比例选择、单点交叉、简单变异。交叉概率设为0.9。突变概率设为0.15。

PSO设计如下:惯性权重设为0.9。加速度系数 $c_1$ 设为0.12, $c_2$ 设为1.2。

这里实施了3个比较实验。实验中分别将3个优化技术的种群大小分别设置为100、200和1000,并将最大迭代次数设置为500。为了避免随机性,改进的人工鱼群算法、GA和PSO均分别执行了10次。比较实验结果如表2所示。

表2 利用不同测试数据生成算法在直角三角形判别程序中的比较结果

实验方法	鱼群数量	成功率/%	搜索最优解的迭代次数		
			最小	最大	平均
本算法	100	100	201	479	335
	200	100	70	191	297
	1000	100	49	73	53
遗传算法	100	0	-	-	-
	200	0	-	-	-
	1000	0	-	-	-
粒子群优化算法	100	50	74	-	-
	200	70	58	-	-
	1000	100	46	75	56

图3、图4和图5显示了该方法中的适应度值随迭代次数的增加而逐渐收敛的情况,其中种群大小分别为100、200和1000。

在这个例子中,改进的人工鱼群算法在小于500次迭代的情况下发现目标测试数据;使用GA在小于500次无法获得目标测试数据;当PSO算法中种群大小为1000时,PSO才能在不超过500次迭代的情况下找到目标测试数据。

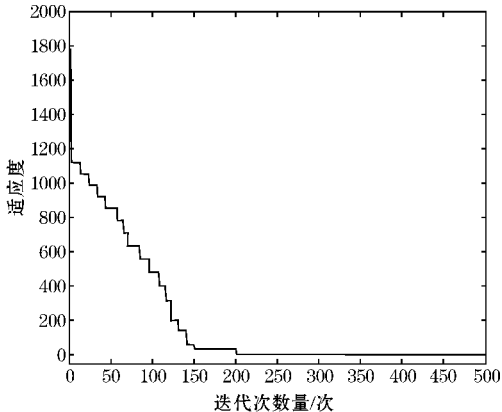


图3 100个AF的优化性能曲线

无论种群大小为100、200还是1000,无论是从最小的、最大的和平均迭代次数来看,改进的人工鱼群算法比GA和PSO有更好的效果。

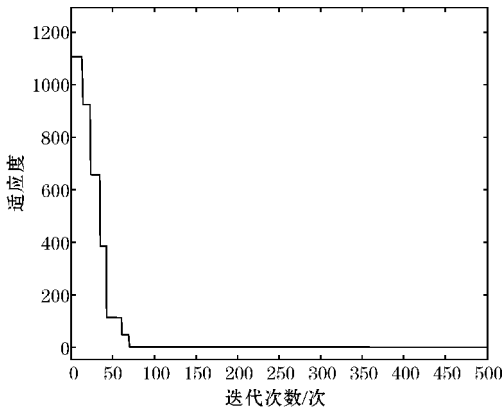


图4 200个AF优化性能曲线

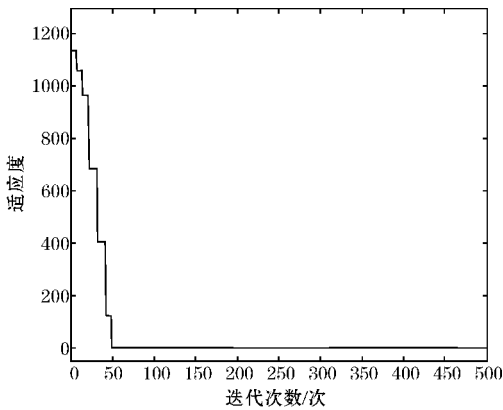


图5 1000个AF优化性能曲线

表2表明,当种群大小设置为100、200和1000时,GA不能达到目标测试数据。当种群大小设置为100和200时,PSO不能达到目标测试数据。同时,改进的人工鱼群算法每次都能找到目标测试数据。当种群大小设置为1000时,改进的人工鱼群算法也比PSO具有更好的效果。

文献[8]采用免疫遗传算法求解直角三角形判别程序。测试用例的3个输入的范围为(0,200)。

当群体大小设置为200时,方法在平均289次迭代中找到最优解。结果表明,改进的人工鱼群算法比免疫遗传算法有更好的效果。

4 结束语

提出了一种用于测试数据自动生成的改进人工鱼群算法。该方法将适应度函数作为目标路径选择的准则。实验结果表明,改进的人工鱼群算法达成了目标测试数据,比其他算法更具一定的优势。结果表明,将



人工鱼群算法应用于测试数据的自动生成是有效的。

致谢:感谢成都市科技项目产业集群协同创新项目(2016-XT00-00015-GX)对本文的资助

参考文献:

[1] Korel B. Automated software test data generation [J]. IEEE Transactions on Software Engineering, 1990,16:870-879.

[2] Korel B. Automated test data generation for programs with procedures, in Proceedings of the 1996 International Symposium on Software Testing and Analysis[J]. ACM Press,1996:209-215.

[3] Christophe Meudec. ATGen: automatic test data generation using constraint logic programming and symbolic execution [J]. Software testing, verifica-

tion and reliability,2001,11:81-96.

[4] 马臻,张毅坤,梁荣,等. 基于免疫遗传算法的构件化软件测试用例生成[J]. 计算机工程,2006,32(23):64-67.

[5] 陈明师,刘晓洁,李涛. 基于多态蚁群算法的测试用例自动生成. 计算机应用研究,2009,26(6):2347-2348.

[6] 李晓磊,邵之江,钱积新. 一种基于动物自治体的寻优模式:鱼群算法[J]. 系统工程理论与实践,2002,22(11):32-38.

[7] 袁爱平,唐一韬,万灿军. 基于粒子群优化算法的软件测试数据生成研究[J]. 计算机与数字工程,2013,41(2):163-164.

[8] 夏芸,刘锋. 基于免疫遗传算法的软件测试数据自动生成[J]. 计算机应用,2008,28(3):723-725.

Test Data Automatic Generation based on Modified Artificial Fish Swarm Algorithm

SHE Jiachun<sup>1</sup>, LIU Tao<sup>1</sup>, MEI Ruowei<sup>1</sup>, HU Shipeng<sup>1</sup>, LI Jing<sup>2</sup>, XU Zili<sup>2</sup>, WANG Guoqiang<sup>2</sup>

(1. College of Electronic Engineering, Chengdu University of Information Technology, Chengdu 610225, China; 2. The Second Research Institute of CAAC, Chengdu 610041, China)

**Abstract:** Aiming at the problem of automatic data generation in software testing field, an improved artificial fish swarm algorithm (AFSA) is proposed. The dynamic step is proposed into the praying, swarming and following behaviors of artificial fish swarm algorithm, which further optimizes the iteration times and efficiency. The improved artificial fish swarm algorithm is used to solve the problem of test data generation of right triangle discriminant program. The results show that the improved AFSA has advantage of fast convergence speed and high accuracy under different populations over the traditional GA and PSO methods.

**Keywords:** artificial fish swarm algorithm; software testing; test data; path-oriented