基于粒子群优化算法的测试用例生成方法

张 娜[1] 滕赛娜[1] 吴 彪[2] 包晓安[1]
(浙江理工大学信息学院 杭州 310018)[1] (山口大学东亚研究科 山口 753-8514)[2]

摘 要 针对标准粒子群算法(Particle Swarm Optimization,PSO)中存在的早熟收敛、易于陷入局部极值的问题,提出了一种基于反向学习与再次搜索的粒子群优化算法(Reverse-Learning and Search-Again PSO,RSAPSO)用于测试用例生成。首先,通过非线性递减的惯性权重函数对学习因子进行改进,实现对种群的初步搜索,并采用梯度下降法完成对最优解与次优解的再次搜索;其次,以极值点为中心设定禁忌区域,对禁忌区域外的粒子进行反向学习,改善种群多样性;最后,采用分支距离法构造适应度函数,评判测试用例的优劣程度。实验结果表明,提出的改进方法在覆盖率、迭代次数和缺陷检测率指标上均有优势。

关键词 粒子群算法,学习因子,反向学习,再次搜索,测试用例生成
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.07.023

Test Case Generation Method Based on Particle Swarm Optimization Algorithm

ZHANG Na[1] TENG Sai-na[1] WU Biao[2] BAO Xiao-an[1]
(School of Information Science and Technology,Zhejiang Sci-tech University,Hangzhou 310018,China)[1]
(The Graduate School of East Asian Studies,Yamaguchi University,Yamaguchi-shi 753-8514,Japan)[2]

Abstract In order to solve the problem of premature convergence and being easy to fall into local extremum in standard particle swarm optimization,this paper put forward a particle swarm optimization based on reverse-learning and search-again for test case generation. Firstly,the learning factor is improved by the nonlinear decreasing inertia weight function,realizing the preliminary search for the population,and the gradient descent method is used to complete the search-again of the optimal solution and the suboptimal solution. Secondly,setting taboo areas with extreme points as the center,the population diversity is improved by the reverse learning of the particles outside the taboo region. Finally,the branch distance method is used to construct fitness function to evaluate the quality of test cases. Experiment results show that the proposed method has advantages in coverage,iteration times and defect detection rate.

Keywords Particle swarm optimization,Learning factors,Reverse learning,Search again,Test case generation

1 引言

在软件工程中,保证软件质量的关键环节是软件测试,理想的软件测试需要具备3个特点:高错误检测能力、广泛的适用性和低成本消耗。它的基本原理是对程序的副本提供一组有代表性的输入数据,在给定的环境下运行这个副本,并对程序的输出进行适当的检查和分析[1]。随着计算机技术的不断发展,软件的规模越来越大,但是软件的错误也越来越难以发现,这样造成的后果会越来越严重。可以说软件测试是保证软件质量、提高软件可靠性的关键[2]。

目前,在已有的粒子群算法的优化与测试用例生成研究中,Shi等[3]引入了惯性权重用于改进PSO算法的速度项,依据迭代进程和粒子飞行状况动态调整惯性权重,使其线性变化,以达到平衡搜索的收敛速度和全局性的目的。Xia等[4]在种群搜索过程中定期对变量空间进行划分,通过对历史信息的分析决定重点搜索区域,缩小搜索空间,进而达到提高搜索效率和求解精度的目的。Mendes等[5]利用所有粒子的历史最优值的加权平均值来引导粒子进行更新。Zhang等[6]提出了ABPSO,该算法根据粒子的聚集程度和种群的多样性原则,动态调节高斯采样的标准差,同时采用变异算子进一步增加种群的多样性。Yang等[7]将one-test-at-a-time策略与粒子群算法相结合,对惯性权重进行动态调整,构造优先级度量函数,更利于单条测试用例的生成。Shi等[8]提出了基于自适应的PSO算法,将粒子群依据粒子适应度和聚集度分为3部分,每部分使用不同惯性权重的调节方案,从而有效提高测试用例自动生成的效率。Li等[9]根据一定概率对粒子进行

到稿日期:2018-06-07 返修日期:2018-09-25 本文受国家自然科学基金(61502430,61562015),广西自然科学重点基金(2015GXNSFDA139038),浙江理工大学521人才培养计划项目资助。

张 娜(1977-),女,硕士,副教授,主要研究方向为软件工程、软件测试;滕赛娜(1994-),女,硕士生,主要研究方向为软件工程、软件测试;吴 彪(1989-),男,博士生,主要研究方向为软件工程、软件测试;包晓安(1973-),男,硕士,教授,主要研究方向为自适应软件、软件测试与智能信息处理,E-mail:baoxiaoan@zstu.edu.cn(通信作者)。

变异操作,采用改进后的分支距离构造法设计适应度函数,并将其应用于字符串型测试数据的自动生成中,实验表明,该方法在数据生成效率上有所提高。Mao 等[10-12]的研究表明,粒子群算法在覆盖表生成规模和执行时间上具有竞争力。

结合上述研究,本文对标准粒子群算法进行改进,引入带有权重函数的学习因子,学习因子随惯性权重的线形或非线性变化相应地递增或者递减,利用两者间的相互作用来平衡算法的全局探索和局部开采能力。其次,采用梯度下降法对最优解与次优解进行再次搜索,设定禁忌区域,对禁忌区域外的粒子进行反向学习,以改善粒子种群的多样性,保证算法的全局探测能力,同时使算法的收敛精度得到提高。最后,采用分支距离法构造适应度函数,通过评判测试用例的优劣程度寻找满足适应度值要求的测试用例。

2　粒子群算法优化

2.1　学习因子

PSO 算法是基于群智能的启发式搜索算法,它模拟了鸟类觅食的过程。假设在 D 维空间中,有一个由 N 个粒子组成的群体,可用两个指标来描述第 $i(i<N)$ 个粒子在第 t 代的信息:位置 $X_i^t=(X_{i1}^t,X_{i2}^t,\cdots,X_{ij}^t,\cdots,X_{iD}^t)$;飞行速度 $V_i^t=(v_{i1}^t,v_{i2}^t,\cdots,v_{ij}^t,\cdots,v_{iD}^t)$。当第 i 个粒子搜索到第 t 代时,个体历史最优位置为 $p_i=(p_{i1},p_{i2},\cdots,p_{ij},\cdots,p_{iD})$,搜索到第 t 代时,整个粒子群的历史最优位置为 $p_g=(p_{g1},p_{g2},\cdots,p_{gj},\cdots,p_{gD})$,则在第 $t+1$ 代时,第 i 个粒子的第 j 维速度和位置的迭代更新公式如下:

$$v_{ij}^{t+1}=\omega * v_{ij}^t+c_1 * r_1 * (p_{ij}^t-x_{ij}^t)+c_2 * r_2 * (p_{gj}^t-x_{gj}^t) \tag{1}$$

$$x_{ij}^{t+1}=x_{ij}^t+v_{ij}^t \tag{2}$$

其中,ω 为惯性权重,作用是衡量下一时刻的速度对下次移动的影响;c_1 和 c_2 为学习因子;r_1 和 r_2 为[0,1]内的随机数。在迭代过程中,若 $v_{ij}^{t+1}>v_{\max}$,则 $v_{ij}^{t+1}=v_{\max}$;若 $v_{ij}^{t+1}<v_{\max}$,则 $v_{ij}^{t+1}=-v_{\max}$。

引入惯性权重从根本上改善了 PSO 算法的性能,对于平衡 PSO 算法的全局搜索能力有着重要的作用。学习因子的作用是加强粒子的学习能力,避免粒子陷入极值,有效改善收敛性。若惯性权重与学习因子两者相互脱离,则会削弱算法在进化过程中的统一性,不利于得到较好的搜索结果,因此本文通过惯性权重来控制学习因子,以解决算法进化过程的统一性问题。

调整策略可分为 3 类:线性、非线性和三角函数。本文通过测试 Griewank,Sphere 和 Rosenbrock 等典型测试函数以及已有的学习因子调整方案,挑选了一种最为合适的方案,即学习因子与惯性权重呈非线性关系。

$$\begin{cases} c_1=A\omega^2+B\omega+C \\ c_2+c_1=2 \end{cases} \tag{3}$$

其中,A,B,C 为常系数。

同时,为匹配算法进程中的非线性变化特点,对于惯性权重 ω,本文决定采用常用的指数函数递减法:

$$\omega=\omega_{\min}+(\omega_{\max}-\omega_{\min})\times\exp[-20(t/T)^6] \tag{4}$$

2.2　再次搜索

粒子群在每次迭代更新位置后,更新种群的历史最优位置 P_{g1} 和历史次优位置 P_{g2},若 P_{g1} 和 P_{g2} 一直保持不变,则判断 P_{g1} 和 P_{g2} 陷入局部极值,此时通过梯度下降法[13]对 P_{g1} 和 P_{g2} 进行再次搜索,并记录最终寻优结果,然后将其淘汰,;同时根据群体多样性准则生成新的粒子,以避免 P_{g1} 和 P_{g2} 对其他粒子产生错误引导从而造成局部最优。

梯度下降法是迭代法的一种,在求解无约束优化问题时最为常用,其具有计算过程简单、初始收敛较快等特点。梯度下降法的基本思想为某一质点沿着函数 $f(p)$ 的梯度下降方向可以快速滑落至函数的极值点处。其迭代公式为:

$$P_g^{k+1}=P_g^k+\lambda_k s^{(k)} \tag{5}$$

其中,$s^{(k)}$ 代表梯度负方向,梯度方向可通过对函数进行求导得到;λ_k 代表梯度方向上的搜索步长。梯度下降法整体可分为两部分:

1)计算梯度负方向:具体公式如下:

$$s^{(k)}=-\nabla f(p^k)=-\left(\frac{\partial f(p^k)}{\partial p_1^k},\frac{\partial f(p^k)}{\partial p_2^k},\cdots,\frac{\partial f(p^k)}{\partial p_D^k}\right) \tag{6}$$

2)计算搜索步长:步长的取值很关键,若过大则会导致发散,若过小则收敛速度太慢,因此 λ_k 取最优步长,必须满足以下函数:

$$f(p^k+\lambda_k s^{(k)})=\min f(p^k+\lambda_{k^{s^{(k)}}}) \tag{7}$$

对陷入局部极值的 P_{g1} 和 P_{g2} 采用梯度下降法进行再次搜索。每个粒子都有固定的适应度函数 $f(x)$,并且适应度函数 $f(x)$ 在第 $i(1,2,3,\cdots,D)$ 维上的偏导数存在。首先设置误差 ε,令 $\varepsilon>0$;其次,根据式(5)计算梯度负方向,若 $\|s^{(k)}\|\leqslant\varepsilon$,则此时的 P_{g1} 和 P_{g2} 位置被保留,并根据种群多样性准则生成新粒子,反之,根据式(6)计算最优步长 λ_k;最后根据式(4)计算出新的 P_{g1} 和 P_{g2}。再次搜索减少了粒子陷于局部极值的情况,同时采用梯度下降法提高了搜索精度,有效提高了寻优效率。

2.3　反向学习

以再次搜索后得到的最优粒子 P_{g1} 和次优粒子 P_{g2} 为中心,以某一具体长度 R 为半径,构成得到一个圆域空间 $\Omega_R(P_g)$,当粒子 X_i 搜索到 $\Omega_R(P_g)$ 边界时,令 V_j 反射,从而防止粒子重复搜索 $\Omega_R(P_g)$ 区域,避免陷入局部最优,进而提高寻优效率。粒子反射示意图如图 1 所示。

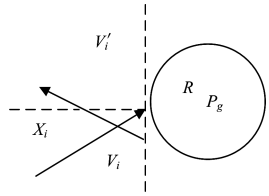

图 1　粒子 X_i 反射示意图

Fig. 1　Schematic diagram of reflection of particle X_i

对于反射的粒子进行反向学习。第 i 个粒子反向学习的对象为该粒子的历史最差位置 $W_i=(w_{i1},w_{i2},\cdots,w_{ij},\cdots,w_{iD})$,以及初始化种群时选择的规模为 m 的初始最差粒子个

体的位置集合 $W=\{W_1^0,W_2^0,\cdots,W_m^0\}$ 中的任一个体 $W_k^0=(W_{k1}^0,W_{k2}^0,\cdots,W_{kj}^0,\cdots,W_{kD}^0)(1\leqslant K\leqslant M)$。第 i 个粒子反向学习过程时的速度更新公式为：

$$v_{ij}^{t+1}=\omega * v_{ij}^t+c_3 * r_3 * (x_{ij}^t-w_{ij}^t)+c_4 * r_4 * (x_{ij}^t-w_{ij}^0) \tag{8}$$

其中，$w_{ij}^t(1\leqslant i\leqslant n)$为第 i 个粒子进化到第 t 代时，其历史最差位置 w_i 的第 j 维的值，$w_{kj}^0(1\leqslant k\leqslant m)$为随机选择的初始最差粒子的个体位置 W_k^0 的第 j 维的值。为保证 m 个初始最差粒子能把进行反向学习的粒子牵引出当前的局部最优陷阱，并较为广泛地分布到搜索区域中，这 m 个初始最差粒子间应具备较大的欧氏距离，因此在选择初始最差粒子时需保证它们两两间的距离大于 R。

反向学习能迅速逃离局部最优的原因是其利用了种群初始最差位置和个体历史最差位置的牵引作用。在反向学习期间，可动态调整粒子最大飞行速度 $v_{\max}$，因为较大的 $v_{\max}$ 可以提高粒子的逃逸速度，也能使粒子在更大的空间内进行搜索，从而提高求解成功率。

3 测试用例生成

3.1 测试用例生成方法的模型

本文提出一种基于粒子群优化算法的测试用例生成方法。首先，引入带有权重函数的学习因子，学习因子随惯性权重的线形或非线性变化相应地递增或者递减，因此通过惯性权重与学习因子的相互作用来达到平衡算法全局探索和局部开采能力的目的。其次，引入再次搜索和反向学习，以提高求解精度、改善种群多样性等。最后，在测试用例生成模块，考虑不同分支节点的优劣程度，设计更加合理的适应度函数评价。本文的基于粒子群优化算法的测试用例生成方法模型如图 2 所示。

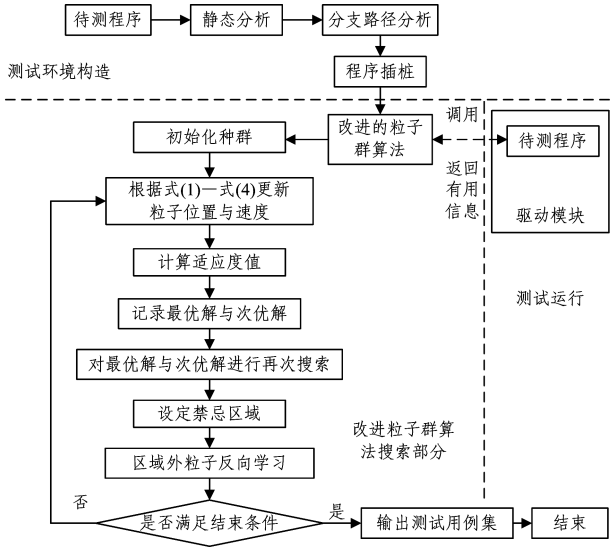

图 2 基于粒子群优化算法的测试用例生成方法模型

Fig. 2 Test case generation method model based on particle swarm optimization algorithm

3.2 适应度函数设计

构建适应度函数是测试用例生成的一个重要环节。将测试用例生成问题通过适应度函数转化为函数优化问题，PSO 算法才能发挥其优化能力，以解决测试用例生成问题，因此，作为连接测试用例生成和 PSO 算法的桥梁，适应度函数的设计对最终的结果有直接影响。

在测试用例生成中，构造适应度函数的方法分为层接近度法和分支距离法。本文采用的方法为分支距离法。该方法由 Korel 提出[14]，他用表达式表示程序中的每一个分支（或称为分支谓词），用以描述测试用例覆盖该分支所需的条件，并且根据分支谓词，将分支函数插入在各个分支节点以获得当前分支谓词的状态。

在适应度函数设计模块中，假设程序含有 m 个分支节点，被测程序有 n 个输入参数，即测试用例有 n 维度（x_1，x_2，…，x_n）则需要在目标路径上的每个分支前插入分支函数 $f_1(x_1,x_2,\cdots,x_n)$，$f_2(x_1,x_2,\cdots,x_n)$，…，$f_m(x_1,x_2,\cdots,x_n)$然后在被测程序结束时以分支函数叠加的形式插入适应度函数 $F=f_1+f_2+\cdots+f_m$，当 $F=0$ 时，表示测试用例达到了覆盖标准，即覆盖了目标路径，但无法确定适应度函数最大值，因此对适应度函数进行标准化，让其分布在 0～100 的区间之内，适应度函数公式为：

$$F=\frac{\sum_{i=0}^{m}\frac{1}{f_i+1}}{m}\times 100 \tag{9}$$

3.3 RSAPSO 算法流程

RSAPSO 算法流程如算法 1 所示。

算法 1 RSAPSO 算法

输入：算法搜索维数 D，种群规模 K，最大惯性权重 ω_{max}、最小惯性权重 ω_{min}，学习因子 c_1 和 c_2，反向学习因子 c_3 和 c_4，最大飞行速度 v_{max}

输出：种群历史最优解 P_g

Begin

1. 根据 D 和 K，随机初始化粒子种群位置 $X_i^0=(X_{i1}^0,X_{i2}^0,\cdots,X_{ij}^0,\cdots,X_{iK}^0)$，飞行速度 $V_i^0=(v_{i1}^0,v_{i2}^0,\cdots,v_{ij}^0,\cdots,v_{iK}^0)$，$t=0$，$P_i=X_i$；
2. 计算粒子适应度值，令最优粒子个体为 P_{g1}，次优粒子个体为 P_{g1}；
3. WHILE
4. 根据式(3)、式(4)计算学习因子 c_1，c_2 及惯性权重 ω；
5. 根据式(1)、式(2)更新位置 X_i 与速度 V_i；
6. 计算粒子的适应度值，并更新 P_i，P_{g1}，P_{g2}；
7. 根据式(5)－式(7)进行再次搜索；
8. 根据某一具体半径设定禁忌区域；
9. IF 满足反向学习条件
10. 根据式(8)、式(2)分别更新 $X_1^t\sim X_k^t$，$V_1^t\sim V_k^t$；
11. 根据式(1)、式(2)分别更新 $X_{k+1}^t\sim X_K^t$，$V_{k+1}^t\sim V_K^t$；
12. END IF
13. $t=t+1$；
14. END WHILE

END

4 实验仿真及结果分析

4.1 实验对象

实验使用 Matlab2012a 实现，为保证算法性能比较的科学性，基本参数设置如下：最大迭代次数为 $K=1000$，种群规

模为 30,c_2 和 c_1 参考式(3),ω 的取值范围为[0.4,0.9];为了避免随机性带来的影响,每组实验运行 100 次。采用了 4 种典型测试函数,其中 Sphere 和 Rosenbrock 为单峰函数,Rastrigrin 和 Griewank 为带有三角函数的多峰函数,具体如表 1 所列。

表 1　4 种典型测试函数

Table 1　Four typical test functions

函数	名称	表达式	取值范围
F1	Sphere	$f_1(x)=\sum_{i=1}^{D}x_i^2$	[−100,100]
F2	Rosenbrock	$f_2(x)=\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)^2+(1-x_i)^2$	[−30,30]
F3	Rastrigrin	$f_3(x)=\sum_{i=1}^{D}[x_i^2-10\cos(2\pi x_i)+10]$	[−5,5]
F4	Griewank	$f_4(x)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1]$	[−600,600]

为了验证本文提出的改进粒子群算法在测试用例生成中的有效性,选取了 7 个基准被测程序用于实验,并与标准粒子群算法进行比较,从迭代次数、平均覆盖率方面进行评价。被测程序如表 2 所列。

表 2　7 个被测程序

Table 2　Seven tested programs

程序编号	程序名称	变量个数	搜索范围
1	Equal	20	$[-10^2,10^2]$
2	NextDate	3	$[-10^4,10^4]$
3	等边三角形	3	$[-10^4,10^4]$
4	Density	4	$[-10^4,10^4]$
5	IuhnCheck	16	$[-10^4,10^4]$
6	Transform	10	[0,255]
7	QuaDratic	3	$[-10^4,10^4]$

该组程序分别有 Equal、NextDate、等边三角形、Density、IuhnCheck、Transform 和 QuaDratic7 种。Equal 的目标路径为判断两个数组是否相等,长度均设为 10,变量个数为 20,程序来源于 Java Collections;NextDate 的目标路径为日期满足 1912−2050 年的闰年的 2 月 29 日,变量个数为 3,程序来源于文献[15];三角形分类程序具有清晰但比较复杂的逻辑结构,所以常用来作为 TC 自动生成的基准程序,其目标路径为判断三角形是否是等边三角形,变量个数为 3,程序来源于文献[16];Density 为三角形分布的概率密度函数,目标路径是某一随机变量 x 等于中位数 c,变量个数为 4,程序来源于 Apache Commons Math;IuhnCheck 的目标路径为检测银行卡是否有效,变量个数为 16,程序来源于 Apache Commons Valid;Transform 的目标路径为将输入的 16 进制字符转化为 10 进制字符的总和,且总和在 50～100 之间,程序来源于文献[16];QuaDratic 的目标路径为满足二次方程,即满足 $b^2-4ac=0$,且 $a\neq0$,$c\neq0$,程序来源于文献[15]。

为了验证优化过的粒子群算法在测试用例生成中的有效性,需要使用缺陷检测率对其进行评价,选择标准化的测试用例程序度量标准(Normalized APFD,NAPFD)计算缺陷检测率,数值越大表明能更早地发现更多的错误。本文选用 count,tokens,series 和 ntree4 个程序进行对比实验,每组实验运行 50 次,通过多次实验得到的对比结果可减小误差,这 4 个被测程序所存在的缺陷个数如表 3 所列。

表 3　4 个被测程序的缺陷个数

Table 3　Number of defects in four tested programs

程序名称	代码行数	缺陷个数
count	42	15
tokens	288	23
series	192	21
ntree	307	31

4.2　实验结果分析

考虑到惯性权重和学习因子之间的关系有线性、非线性和三角函数 3 种,本文采用了 3 种方案对学习因子进行改进。方案 1 为学习因子与惯性权重呈线性关系,$c_1=X\omega+Y$,$c_2+c_1=2$;方案 2 为学习因子与惯性权重呈非线性关系,$c=A\omega^2+B\omega+C$;$c_2+c_1=2$;方案 3 为学习因子与惯性权重呈三角函数关系,$c=M\pm N\cos(\pi\omega)$,同时惯性权重采用常用的指数函数递减法,$\omega=\omega_{\min}+(\omega_{\max}-\omega_{\min})\times\exp[-20(\frac{t}{T})^6]$,取 $\omega_{\max}=0.9$,$\omega_{\min}=0.4$,$A=0.45$,$B=0,9$,$C=0.45$,$M=1.6$,$N=1.2$,$X=0.45$,$Y=1.45$。各学习因子策略下的最佳适应度值如表 4 所列。

表 4　各学习因子策略下的最佳适应度值

Table 4　Optimal fitness values under each learning factor strategy

学习因子策略	f_1	f_2	f_3	f_4
方案 1	−5.213	0.931	0.902	−5.332
方案 2	−6.461	0.513	0.209	−6.028
方案 3	−5.923	0.690	0.823	−5.397

注:适应度值用以 10 为底的对数形式表示

从表 4 可以看出,本通过非线性惯性权重来控制学习因子的方法得到的函数取值优于其他学习因子方案,方案 2 的效果最佳,方案 3 的效果次之,方案 1 的效果最差。可以发现引入非线性惯性权重对学习因子进行改进,可以增强粒子的相互作用,同时方案 2 在 4 种函数上的优化性能良好,可证明其适用范围较广,可以有效改善算法的全局和局部搜索能力。

从平均覆盖率和迭代次数上对 RSAPSO 与 PSO 进行实验对比,结果如表 5 所列。本文中的平均覆盖率为 100 次运行中被覆盖的路径数目的平均值占总路径数的百分比,迭代次数表示覆盖到目标路径所需要的迭代次数的平均值,若 T 代之内无法覆盖目标路径,则迭代次数为 T。该 7 组程序的目标路径在 4.1 节中均有相应描述。

表 5　RSAPSO 与 PSO 的实验对比结果

Table 5　Experimental comparison results of RSAPSO and PSO

测试程序	RSAPSO		PSO	
	覆盖率/%	迭代次数	覆盖率/%	迭代次数
Equal	100	221.63	80	352.73
NextDate	100	233.75	100	265.79
等边三角形	100	310.28	100	391.63
Density	100	72.43	100	75.26
IuhnCheck	100	471.23	40	753.79
Transform	100	376.82	56	629.71
QuaDratic	52	733.15	49	782.15

从表 5 中可以得出,经过改进后的粒子群算法(RSAPSO)在平均覆盖率和迭代次数上均优于标准的粒子群算法。对于 NextDate、等边三角形和 Density 这 3 个程序而言,在平均覆盖率上两种算法都达到了 100%;在迭代次数方面,由于

这 3 个程序较为简单，所以迭代次数上的差距并不明显。对于 QuaDratic 程序而言，RSAPSO 与 PSO 在平均覆盖率与迭代次数上的表现并未相差太多，因为其取值范围大，且逻辑复杂。在 Equal，IuhnCheck 和 Transform 3 个程序上，RSAPSO 的平均覆盖率均优于 PSO 算法，并且都达到了 100%；迭代次数方面，因为 RSAPSO 的变量多且较为复杂，所以其迭代次数明显少于 PSO 的迭代次数。

因为每组实验均进行 50 次，每次模拟实验的数据无法通过表格的形式完整地展现出来，所以对于缺陷检测的情况通过箱形图的形式进行展现，如图 3 所示。

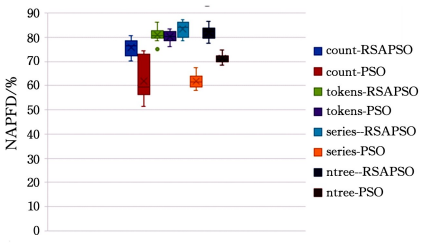

图 3 4 个程序的 NAPFD

Fig. 3 NAPFD of four programs

从图 3 中可以看出，改进的 RSAPSO 在缺陷检测能力上多优于标准的粒子群算法，对于 tokens 程序，两者之间差别并不明显，平均缺陷检测率均在 80%左右，对于其他 3 个程序，RSAPSO 算法相比 PSO 算法有较大提高，尤其是在 series 程序上，差异最明显。

结束语 本文对标准粒子群算法进行改进，结合了反向学习与再次搜索策略，并将其应用于测试用例生成中。对学习因子的改进能更好地平衡算法的全局探索和局部开发能力，反向学习机制可改善种群多样性，保证了算法的全局探索能力，而再次搜索策略能提高收敛速度。在实验部分，通过 7 个测试程序对方法进行了验证，结果证明所提方法在平均覆盖率与迭代次数上均有优势。我们通过另外 4 个程序验证了 RSAPSO 方法在测试用例里缺陷检测率上具有一定优势。本实验用到的数组预先确定了长度，但在实际测试中长度是不确定的，如何测试不确定长度的数组是进一步研究的方向。

参 考 文 献

[1] CHEN H Y, TSE T H, CHEN T Y. TACCLE: a methodology for object-oriented software testing at the class and cluster levels[J]. ACM Transactions on Software Engineering & Methodology, 2001, 10(1): 56-109.

[2] GALLAGHER M N, ARASIMHAN V L. ADT EST: A Test Data Generation Suite for A da Software Systems[J]. IEEE Transactions on Software Engineering, 1997, 23(8): 473-484.

[3] SHI Y, EBERHART R C. Fuzzy adaptive particle swarm optimization[C]// Proceedings of the IEEE Congress on Evolutionary Computation. Seoul, Korea, 2001: 101-106.

[4] XIA X W, LIU J N, GAO K F, et al. An improved particle swarm optimizer based on tabu detecting and local learning strategy in a shrunk search space[J]. Applied Soft Computing, 2014, 23(1): 76-90. (in Chinese)
夏学文，刘经南，高柯夫，等. 具备反向学习和局部学习能力的粒子群算法[J]. 计算机学报，2015，38(7)：1397-1407.

[5] MENDES R, KENNEDY J, NEVES J. The fully informed particle swarm: Simpler, maybe better[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 204-210.

[6] ZHANG Y, GONG D W, SUN X Y, et al. Adaptive barebones particle swarm optimization algorithm and its convergence analysis[J]. Soft Computing, 2014, 18(7): 1337-1352.

[7] BAO X A, YANG Y J, ZHANG N, et al. Composite test case generation method based on Adaptive Particle Swarm Optimization[J]. Computer Science, 2017, 44(6): 177-181. (in Chinese)
包晓安，杨亚娟，张娜，等. 基于自适应粒子群优化的组合测试用例生成方法[J]. 计算机科学，2017，44(6)：177-181.

[8] SHI J J, JIANG S J, HAN H, et al. Adaptive particle swarm optimization algorithm and its application in test data generation [J]. Chinese Journal of Electronics, 2013, 41(8): 1555-1559. (in Chinese)
史娇娇，姜淑娟，韩寒，等. 自适应粒子群优化算法及其在测试数据生成中的应用研究[J]. 电子学报，2013，41(8)：1555-1559.

[9] LI G, SUN L, SUN H H, et al. String type test data generation based on mutation particle swarm optimization algorithm[J]. Computer Science, 2016, 43(11): 252-256, 279. (in Chinese)
李刚，于磊，孙回回，等. 基于变异粒子群算法的字符串型测试数据生成[J]. 计算机科学，2016，43(11)：252-256，279.

[10] BUENO P M S, JINO M, WONG W E. Diversity oriented test data generation using metaheuristic search techniques [J]. Information Science, 2014, 25(9): 490-509.

[11] MAO C Y, YU X X, XUE Y Z. Algorithm Design and Empirical Analysis for Particle Swarm Optimization Based Test Data Generation[J]. Journal of Computer Research and Development, 2014, 51(4): 824-837. (in Chinese)
毛澄映，喻新欣，薛云志. 基于粒子群优化的测试数据生成及其实证分析[J]. 计算机研究与发展，2014，51(4)：824-837.

[12] CHEN X, GU Q, WANG Z Y, et al. Framework of Particle Swarm Optimization Based Pairwise Testing[J]. Journal of Software, 2011, 22(12): 2879-2893. (in Chinese)
陈翔，顾庆，王子元，等. 一种基于粒子群优化的成对组合测试算法框架[J]. 软件学报，2011 22(12)：2879-2893.

[13] LU C, SHENG W, HAN Y, et al. Phase-only pattern synthesis based on gradient-descent optimization[J]. Journal of Northeast Petroleum University, 2016, 38(4): 297-307.

[14] KOREL B. Automated software test data generation[C]// IEEE Trans on Software Engineering, 1990, 16(8): 870-879.

[15] KIFETEW M F, PANICHELLA A, DE LUCIA A, et al. Orthogonal exploration of the search space in evolutionary test case generation [C] // Proceedings of the 2013 International Symposium on Software Testing and Analysis. Lugano: ACM, 2013: 257-267.

[16] XIAO M, EL-ATTAR M, REFORMAT M, et al. Empirical evaluation of optimization algorithms when used in goal-oriented automated test data generation techniques[J]. Empirical Software Engineering, 2007, 12(2): 183-223.