文章编号:1671-4598(2019)06-0009-04 DOI:10.16526/j.cnki.11—4762/tp.2019.06.003 中图分类号:TP311 文献标识码:A

基于 Tent 混沌的测试用例优先级排序

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

摘要:针对标准粒子群算法(Particle Swarm Optimization,PSO)后期出现的早熟收敛,提出了一种基于 Tent 混沌的粒子群优化算法(Tent—ChaosParticle Swarm Optimization,TCPSO)用于测试用例优先级排序;首先,利用改进的 Tent 映射的三大特性初始化种群,使得粒子均匀分布,提高初始解的质量;并通过非线性递减的惯性权重函数对学习因子进行改进,以更新粒子速度与位置信息;其次,对陷入局部最优的粒子 P_{id} 进行混沌搜索,跳出局部最优,同时对当前种群中部分最差粒子 P_{iw} 进行混沌搜索,改善种群多样性;最后,采用测试用例缺陷检测率作为评价标准,评判测试用例优劣程度;实验表明,提出的改进方法在寻优能力和缺陷检测率指标上均有优势。

关键词:Tent 映射;粒子群算法;学习因子;混沌搜索;测试用例排序

Test Case Prioritization Based on Tent Chaos

Zhang Na[1], Teng Saina[1], Wu Biao[2], Bao Xiaoan[1]
(1. School of Information Science and Technology, Zhejiang Sci—tech University, Hangzhou 310018, China;
2. Graduate School of East Asian Studies, Yamaguchi University, Yamaguchi—shi 753—8514, Japan)

Abstract: Aiming at the premature convergence of the standard particle swarm optimization (PSO) algorithm, a new PSO algorithm based on Tent chaos (TCPSO) is proposed to prioritize test cases. Firstly, the population is initialized by using the randomness, ergodicity and regularity of the improved Tent map, so that the particles are evenly distributed and the quality of the initial solution is improved. At the same time, chaotic search is carried out for the optimal particle and some of the worst particle in the current population to improve the diversity of the population. Finally, the branch coverage of the test case and defect detection rate are used as the evaluation criterion to judge the quality of the test case. Experiments show that the improved method has advantages in branch coverage and defect detection rate index.

Keywords: Tent mapping; particle swarm optimization; learning factor; chaos search; test case prioritization

0 引言

回归测试是指对修改后的代码进行重复测试,确认未产生新的缺陷。在软件开发过程中,频繁使用回归测试可以确保软件的质量,所以降低回归测试的成本是重中之重,而生成后的测试用例集进行排序或优化[1]是一种非常有效的方法。近年来,智能搜索算法也开始被应用于解决测试用例排序问题,如粒子群算法[2]、蜂群算法等。

目前,在已有的研究中,Yu 等人[3]将软件转换为类级有向网络模型,通过杜铃模型的风险值作为排序依据,从而提高错误检出率。Zhang 等人[4]通过关注三个影响因子(需求覆盖率、测试用例失效率、测试用例重要度),动态调整测试同理优先级。Chang 等人[5]基于历史信息和动态调整策略,改进测试用例优先级技术以尽早地发现缺陷。Meng 等人[6]将混沌融入粒子群中,当最优粒子与普通粒子的距离小于某值时,进行混沌搜索。Zhang 等人[7]将 OTT 策略和粒子群相结合,在测试用例重要度和失效率上具有一定优势。Zhu[8]在测试用例优先级排序中引入了缺陷影响因素,通过实验证明其可以有效保证软件产品的质量。Xu 等人[9]提出粒子可以通过在混沌与稳定之间的交替运动,从而得到最优解,以跳出局部最优。Wang 等人[10]通过定义失效覆盖等价划分优化选择准则来提高错误定位的有效性,不同的测试同理赋予不同优先级。Zheng 等人[11]根据函数调用关系图进行关联性分析并对测试用例排序,大大减少了回归测试的成本。

结合上述研究,本文对 PSO 进行改进,结合了混沌算法的思想,提出了基于 Tent 混沌的粒子群优化算法(Tent—Chaos Particle Swarm Optimization,TCPSO)。对 Tent 映射引入参数,防止粒子落入小周期内,并引入带有权重函数的学习因子,两者相结合进行非线性递减变化,平衡 TCPSO 算法全局与局部能力,其次,对陷入局部最优的粒子及部分最差粒子进行混沌搜索优化,保证种群多样性,同时跳出局部最优,最后,以测试用例缺陷检测率作为排序的评判标准进行实验,并验证算法具有较好的寻优能力。

收稿日期:2018-10-24; **修回日期**:2018-11-22。
基金项目: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。
作者简介:张 娜(1977-),女,浙江宁波人,硕士,副教授,主要从事软件工程、软件测试方向的研究。
包晓安(1973-),男,硕士,教授,主要从事自适应软件、软件测试与智能信息处理方向的研究

1 粒子群算法优化

1.1 初始化优化

在标准粒子群的初始化中，解的质量对最终结果有着重要影响，种群的速度和位置信息一般随机产生，它可以使得初始种群均匀分布，由于部分粒子可能会远离最优解，所以粒子的质量不能完全保证，从而影响算法收敛速度和最终结果。

利用混沌序列本身具有的规律性，随机性及遍历性三大特性对粒子进行初始化优化，既能保持种群多样性，同时利于跳出局部最优，改善 PSO 算法的全局搜索能力。映射算法一般有四种，被引用最多的为 Logistic 和 Tent[12]两种，Dan 等人[13]指出在［0，1］区间内，Tent 映射产生的混沌序列与 Logistic 映射产生的混沌序列相比分布更均匀，所以本文在种群初始化中引入 Tent 混沌映射算法，并对 Tent 方程进行改进，以提高初始解质量。

改进后 Tent 映射的迭代公式如下：

$$x_{k+1}=\begin{cases}2x_k+\alpha & 0\leqslant x_k\leqslant\dfrac{1}{2}\\ 2-x_k+\beta & \dfrac{1}{2}<x_k\leqslant 1\end{cases}\tag{1}$$

其中：x_k 为粒子的位置信息，K 为粒子的迭代次数，α，β 为调度参数，取值范围［−0.1，0.1］，其作用是避免粒子落入小周期内。

Tent 映射经贝努利移位变换后的公式如下所示：

$$x_{k+1}=2(x_k)\bmod\tag{2}$$

1.2 位置和速度更新

标准粒子群算法[12]利用位置与速度信息来描述第 i（$i<N$）个粒子在第 t 代时的表现：位置 $X_i^t=(X_{i1}^t,X_{i2}^t,\cdots,X_{ij}^t,\cdots,X_{iL}^t)$；飞行速度 $V_i^t=(v_{i1}^t,v_{i2}^t,\cdots,v_{ij}^t,\cdots,v_{iD}^t)$，维度为 D。第 i 个粒子的第 j 维速度和位置的迭代更新公式如下：

$$v_{ij}^{t+1}=\omega*v_{ij}^t+c_1*r_1*(p_{ij}^t-x_{ij}^t)+c_2*r_2*(p_{gj}^t-x_{gj}^t)\tag{3}$$

$$x_{ij}^{t+1}=x_{ij}^t+v_{ij}^t\tag{4}$$

其中：ω 为惯性权重，c_1、c_2 为学习因子，r_1、$r_2\in$［0，1］，p_{ij}^t 为个体历史最优解，p_{gj}^t 为种群全局最优解，t 为当前迭代次数。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 $v_{ij}^{t+1}=-/v_{max}/$。

将学习因子与惯性权重相结合，两者进行相关联的非线性递减变化，公式如下所示：

$$\begin{cases}c_1=A\omega^2+B\omega+C\\ c_2+c_1=2\end{cases}\tag{5}$$

其中：A，B，C 为常系数。

同时惯性权重 ω 采用常用的指数函数递减法，用以匹配算法过程中的非线性变化特点：

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

在本文提出的 TCPSO 中，随着每一维位置与速度信息的更新，均计算个体历史最优 p_{id} 和种群全局最优 p_{gd}，而非所有维度的位置和速度信息更新完毕后，再计算 p_{id} 和 p_{gd}。

1.3 混沌优化算法

混沌运动有三大特点：1）随机性，混沌类似于随机，因而具有随机性；2）遍历性，在一定范围之内，混沌能够使粒子不重复经历任何一种状态；3）规律性，虽然混沌类似于随机，但是混沌本身也有一定的规律。因此，通过混沌运动，种群在跳出局部最优的同时也能寻找全局最优。

当粒子经过几次迭代后，少数优秀粒子被保存下来，此时，粒子容易陷入局部最优，所以为了跳出局部最优，保证种群多样性，本文引入了混沌搜索进行优化。首先分别以当前粒子的最优解 p_{id} 和最差的百分之 20 的粒子 p_{iw} 为基础，进行混沌搜索，产生与之对应的混沌序列，然后，以 p_{id} 为基础产生的混沌序列中的最优解代替原粒子的最优解 p_{id}，以 p_{iw} 为基础产生的混沌序列中的粒子代替原粒子中最差的百分之 20。

最优解取代的具体步骤如下所示：

步骤 1：利用以下公式将最优解 p_{id} 的变量取值范围［a，b］映射到混沌算法的区间［0，1］

$$Z_i=\frac{p_{id}-a}{b-a}\tag{7}$$

步骤 2：利用 Tent 方程，经过 M 次迭代，得到 m 个混沌变量 Z_i^m（$m=1,2,3,\cdots,M$），产生新的混沌序列。

步骤 3：将 m 个混沌变量通过逆转换，从区间［0，1］映射到粒子群算法的取值区间［a，b］

公式如下所示：

$$P_{id}^m=a+(b-a)Z_i^m\tag{8}$$

步骤 4：将混沌序列中的最优解取代原粒子群算法得到最优解 p_{id}。

最差的百分之 20 的粒子 p_{iw} 的具体步骤如下所示：

步骤 1：利用以下公式将最差的百分之 20 的粒子 p_{iw} 的变量取值范围［a，b］映射到混沌算法的区间［0，1］。

$$Z_i=\frac{p_{iw}-a}{b-a}\tag{9}$$

步骤 2：利用 Tent 方程，经过 K 次迭代，得到 k 个混沌变量 Z_i^m（$m=1,2,3,\cdots,K$），产生新的混沌序列。

步骤 3：将 k 个混沌变量通过逆转换，从区间［0，1］映射到粒子群算法的取值区间［a，b］，公式如下所示：

$$P_{iw}^k=a+(b-a)\tag{10}$$

步骤 4：将混沌序列中的粒子取代原粒子群算法得到的最差的百分之 20 的粒子 p_{iw}。

2 基于混沌的测试用例优先级排序

2.1 实数编码

测试用例排序是指对测试用例集 TS 中的测试用例进行排序，通过判断最终找到一个最优的排列，降低测试成本，能更早发现程序中的错误。本文通过实数编码表示测试用例集中每个测试用例的序号。假设测试用例集 TS 中有 M 个测试用例，那么 TS 的任意一个序列可用粒子 $X=(x_{t1},x_{t2},\cdots,x_{tm})$ 表示，其中 t_m 表示测试用例集 TS 中第 m 个

测试用例，x_m表示测试用例t_m在测试用例集 TS 中的序号，且$1 \leqslant x_i \leqslant M$。

2.2　优先级评价标准

测试用例优先级技术（test case prioritization，TCP）是一个广泛的研究热点。该问题可以描述为满足下列公式：

$$(\forall T'')(T'' \in PT)(T'' \neq T')[f(PT') \geqslant f(PT'')] \quad (12)$$

其中：T 为测试用例集，PT 为测试用例集中所有的可能的排列组合，f 为目标函数。

适应度函数用以引导搜索算法的搜索方向，它决定了搜索算法能否快速有效地找到全局最优解，同时粒子的适应度值反应了该粒子是否为优质解，本文目的在于对测试用例进行优先级排序，减少回归测试成本，尽早发现缺陷，所以采用标准化的测试用例程序度量标准（normalized average of the percentage of faults detected，NAPFD）计算缺陷检测率，该标准用于衡量测试用例的优先级，公式如下所示：

$$NAPFD = p - \frac{TF_1 + TF_2 + \cdots + TF_m}{nm} + \frac{p}{2n} \quad (11)$$

其中：n 表示测试用例集中参与测试的测试用例个数，m 表示缺陷个数，p 是 n 中缺陷个数与候选测试用例集 T 中缺陷个数的比率，TF_i（$1 \leqslant i \leqslant m$）表示检测出第 i 个缺陷需要运行的测试用例个数。通过公式可以得出，当 $NAPFD$ 的值越大，则发现错误的速度越快。

3　实验仿真及结果分析

3.1　实验对象

实验使用 Matlab2012a 实现，基本参数设置如下：最大迭代次数 $K=1\ 000$，种群规模为 $K=30$，c_2，c_1 参考前文所写公式（5），ω 参考前文所写公式（6），$\Omega \in [0.4, 0.9]$，为了避免随机性带来的影响，每组实验运行 100 次。本文采用了四种典型测试函数验证 TCPSO 算法的性能，并将结果与表粒子群算法进行对比，4 种测试函数如表 1 所示。

表 1　4 种测试函数表

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

其中函数 F1 函数为 Sphere，F2 函数为 Rosenbrock，F3 函数为 Rastrigrin，F4 函数为 Griewank。为验证本文提出的 TCPSO 算法在测试用例排序上的有效性，本文对 6 个不同类型的程序进行实验，并与标准粒子群算法进行结果对比，从缺陷检测率角度进行评价，程序相关信息如表 2 所示。

表 2　6 个被测程序

编号	名称	程序行数测试用例个数程序	来源
1	Print_tokens	7264130	文献[15]
2	Schedule	4122560	文献[15]
3	count	4241	文献[16]
4	series	288175	文献[16]
5	tokens	19237	文献[16]
6	ntree	30764	文献[16]

3.2　实验结果分析

针对 4 个测试函数，本文采用标准粒子群优化算法和 TCPSO 算法分别进行了实验，评判标准为平均适应度值的大小，如表 3 所示。

表 3　各测试函数的平均适应度值

名称	函数	标准 PSOTCPSO
f_1	Sphere	2.819E+02 1.982E−09
f_2	Rosenbrock	1.862E+01 3.783E−05
f_3	Rastrigrin	1.056E+03 7.643E−05
f_4	Griewank	3.028E+07 3.269E−12

从表 3 中可以看出，本文提出的 TCPSO 算法得到的函数值明显优于标准粒子群算法，搜索精度提高了 5 倍以上，普遍在 10 倍左右，其中改进后的粒子群算法在 f_4 函数上取得了最好的优化结果，将精度提高了近 20 倍。实验结果表明，TCPSO 算法的适用范围广泛，全局和局部搜索能力得到提高。

本文对标准粒子群优化算法和 TCPSO 算法分别通过表 1 中的 6 个被测程序进行了测试用例排序实验。对于程序 Print _ tokens 和 Schedule，由于两者的测试用例较多，所以均匀地从中随机选取了 56 和 75 个测试用例用于排序实验。通过计算 6 个被测程序排序后的最优排列的 $NAPFD$，以上实验进行 100 次，由于实验次数较多，所以实验结果用箱形图形式展现，便于整体观察分析，箱形图如图 1～2 所示。

通过图 1 和 2 的箱形图对比可知，TCPSO 算法在缺陷检测率上明显优于标准粒子群算法，其中对于 Schedule 程序而言，优化程度是最高的，$NAPFD$ 将近提高百分之二十五，但对于 tokens 程序而已，缺陷检测率提高并不明显，两种算法的 $NAPFD$ 相近，其余算法的缺陷检测率提高程度近似；count 程序在 PSO 算法的应用中，$NAPFD$ 数值波动较大，而在 TCPSO 算法的应用中，较为均衡。

通过两个实验结果表明，基于 Tent 混沌的粒子群算法在测试用例排序问题上效果显著，在粒子寻优能力与测试用例缺陷检测率两个指标上均有优势。

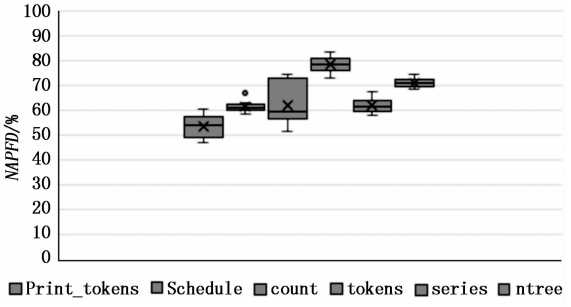

图 1 PSO 算法对应的 *NAPFD*

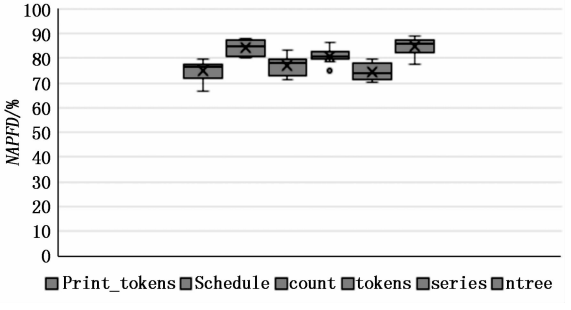

图 2 TCPSO 算法对应的 *NAPFD*

4 结束语

本文将粒子群算法与混沌算法相结合，应用于测试用例排序研究中。对 Tent 映射添加参数，使粒子避免落入小周期内，提高了初始种群质量；学习因子与惯性权重相结合，进行非线性递减变化，用以平衡算法；对陷入局部最优和部分最差粒子进行混沌搜索优化，以跳出局部最优同时保证种群多样性。在实验部分，通过 4 种典型测试函数和 6 个被测程序对优先级排序进行验证，结果表明在粒子寻优能力和测试用例缺陷检测率上有优势。本实验用到的程序并非大型程序，如何对大型的程序进行测试用例优先级排序是进一步的研究问题。

参考文献：

[1] Christen P. Data matching：Concepts and techniques for record linkage，entity resolution，and duplicate detection [M]. New York：Springer Science & Business Media，2012.

[2] 张卫祥，齐玉华，李德治. 基于离散粒子群算法的测试用例优先排序 [J]. 计算机应用，2017，37 (1)：108－113，169.

[3] 于 海，杨 月，王 莹，等. 基于风险分析的回归测试用例优先级排序 [J/OL]. 计算机学报，2017：1－19.

[4] 张 娜，姚 澜，包晓安，等. 多目标优化的测试用例优先级在线调整策略 [J]. 软件学报，2015，26 (10)：2451－2464.

[5] 常龙辉，缪淮扣，肖 蕾. 基于历史信息的自适应测试用例优先级技术 [J]. 计算机科学，2015，42 (9)：154－158.

[6] Meng H J，Zheng P，Wu R Y，et al. A hybrid particle swarm algorithm with embedded chaotic search [A]. Proceedings of the 2004IEEE Conference on Cybernetics and Intelligent Systems [C]. Singapore，2004，367－371.

[7] 张 娜，林青霞，吴 彪，等. 基于 OTT 策略的可变力度组合测试用例优先级排序方法 [J]. 计算机测量与控制，2018，26 (7)：26－31.

[8] 朱凌燕. 基于缺陷的测试用例优先级排序方法 [J]. 电子技术与软件工程，2017 (23)：56－58.

[9] 胥小波，郑康锋，李 丹，等. 新的混沌粒子群优化算法 [J]. 通信学报，2012，33 (1)：24－30，37.

[10] 王克朝，王甜甜，苏小红，等. 面向有效错误定位的测试用例优选方法 [J]. 计算机研究与发展，2014，51 (4)：865－873.

[11] 郑锦勤，牟永敏. 基于函数调用路径的回归测试用例选择排序方法研究 [J]. 计算机应用研究，2016，33 (7)：2063－2067.

[12] Shamir A. Identity Based Crypto systems and signature Schemes [A]. Proc. of cryptology－CRYPTO' 84 [C]. Berlin，Germany：Springer－Verlag，1984.

[13] 单 梁，强 浩，李 军，等. 基于 Tent 映射的混沌优化算法 [J]. 控制与决策，2005，20 (2)：179－182.

[14] Kennedy J，Eberhart R C. Particle swarm optimization [A]. Proceedings of the IEEE IntConf on Neural Net－works [C]. Perth，Australia：IEEE Press，1995：1942－1948.

[15] Hutchins M，Foster H，Goradia T，et al. Experiments on the effectiveness of data－flow and control－flow based test adequacy criteria [A]. Proc. 16th International Conf. Software Eng [C]. 1994：191－200.

[16] 沈嘉懿，李 芳，徐飞龙，等 . 中文组织机构名称与简称的识别 [J]. 中文信息学报，2007，21 (6)：17－21.

(上接第 4 页)

参考文献：

[1] 夏定纯，徐 涛. 人工智能技术与方法 [M]. 华中科技大学出版社，2004.

[2] 张 妮，徐文尚，王文文. 人工智能技术发展及应用研究综述 [J]. 煤矿机械，2009，30 (2)：4－7.

[3] 沈林城，关世义. 开放式飞行任务规划方法 [J]. 宇航学报，1998，19 (2)：13－18.

[4] 席 政. 人工智能在航天飞行任务规划中的应用研究 [J]. 航空学报，2007，28 (4) ：791－795.

[5] 张 克，邵长胜，强文义. 基于面向 Agent 技术的任务规划系统研究 [J]. 高技术通讯，2002，12 (5)：82－86.

[6] 鲁 宇. 中国运载火箭技术发展 [J]. 宇航总体技术，2017 (3)：5－12.

[7] 郭凤英，何洪庆. 人工智能技术在航天领域的应用 [J]. 中国航天，1996 (6)：19－21.

[8] 谭 勇，王 伟. 智能故障诊断技术及发展 [J]. 飞航导弹，2009 (7)：35－38.