

一种自适应柯西变异的反向学习粒子群优化算法

康岚兰^{1,2} 董文永¹ 田降森¹

(武汉大学计算机学院 武汉 430072)¹ (江西理工大学应用科学学院 赣州 341000)²

摘要 针对传统粒子群优化算法易出现早熟的问题,提出了一种自适应变异的反向学习粒子群优化算法。该算法在一般性反向学习方法的基础上,提出了自适应柯西变异策略(ACM)。采用一般性反向学习策略生成反向解,可扩大搜索空间,增强算法的全局勘探能力。为避免粒子陷入局部最优解而导致搜索停滞现象的发生,采用 ACM 策略对当前最优粒子进行扰动,自适应地获取变异点,在有效提高算法局部开采能力的同时,使算法能更加平稳快速地收敛到全局最优解。为进一步平衡算法的全局搜索与局部探测能力,采用非线性的自适应惯性权值。将算法在 14 个测试函数上与多种基于反向学习策略的 PSO 算法进行对比,实验结果表明提出的算法在解的精度以及收敛速度上得到了大幅度的提高。

关键词 粒子群优化,一般性反向学习,自适应柯西变异,自适应惯性权值

中图分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.10.046

Opposition-based Particle Swarm Optimization with Adaptive Cauchy Mutation

KANG Lan-lan^{1,2} DONG Wen-yong¹ TIAN Jiang-sen¹

(Computer School, Wuhan University, Wuhan 430072, China)¹

(College of Applied Science, Jiangxi University of Science and Technology, Ganzhou 341000, China)²

Abstract To solve the problem of premature convergence in traditional particle swarm optimization (PSO), this paper proposed a opposition-based particle swarm optimization with adaptive Cauchy mutation. The new algorithm applies adaptive Cauchy mutation strategy (ACM) on the basis of generalized opposition-based learning method (GOBL). GOBL strategy to generate solutions can expand the search space and enhance the global explorative ability of PSO. Meanwhile, adaptive Cauchy mutation strategy was presented to disturb the current optimal particle and adaptively gain variation points in order to avoid the best particle being trapped into local optima, since this may cause search stagnation. This strategy is helpful to improve the exploitation ability of PSO and make the algorithm more smoothly fast converge to the global optimal solution. In order to further balance the global search and local explorative ability of the algorithm, this paper applied a nonlinear adaptive inertia weight. The new algorithm was compared with several opposition-based PSO on 14 benchmark functions. The experimental results show that the new algorithm greatly improves accuracy and convergence speed of solution.

Keywords Particle swarm optimization, Generalized opposition-based learning, Adaptive Cauchy mutation, Adaptive inertia weight

1 引言

粒子群优化算法(Particle Swarm Optimization, PSO)是 Kennedy 和 Eberhart 等人于 1995 年提出的一种群体智能优化算法^[1]。PSO 源于对鸟类捕食行为的模拟,通过鸟之间的集体协作使群体达到最优。它概念简单、易于理解、参数少且易实现,在求解最优化问题特别是一些非线性、多峰等较复杂的优化问题时,表现出较强的全局搜索能力,吸引了大量科研人员对其进一步研究,并成功应用于多个科学与工程实践领域,表现出良好的性能。

PSO 是一种基于种群搜索的优化算法。与其它群体智能算法一样,传统 PSO 存在着局部搜索能力较差、搜索精度不高、算法易陷入局部最优以及搜索性能对参数过分依赖等缺点。为了改善这一问题,很多研究者从参数设置、收敛性、与其它算法融合等角度对传统 PSO 进行改进,提出了各种不同的粒子群改进算法。在基本 PSO 中,每个粒子代表问题的一个可能解,其下一代中粒子飞行方向受自身速度、自身极值($pbest$)与当前全局最优位($gbest$)的影响,因此,如何给定惯性大小,如何充分利用两类极值指导粒子在快速“飞入”全局最优位的同时避免陷入局部最优,是改进基本 PSO 的关键。

到稿日期:2014-05-17 返修日期:2014-07-20 本文受国家自然科学基金项目:智能仿真优化理论与方法研究(61170305),伊藤算法及其在动态仿真优化中的理论研究(60873114)资助。

康岚兰(1979-),女,博士生,讲师,主要研究方向为演化计算、机器学习, E-mail: victorykil@163.com;董文永(1973-),男,教授,博士生导师,主要研究方向为计算机仿真与控制、演化计算、机器学习、数据挖掘;田降森(1988-),男,硕士生,主要研究方向为演化计算、机器学习。

本文通过分析自身极值与当前全局最优位的距离关系,提出一种新的自适应变异反向学习 PSO 算法(Adaption Cauchy Mutation Opposition-based PSO, ACMOPS)。ACMOPS 采用全局最优位自适应柯西变异策略(ACM),将其融入到反向学习中,使粒子在各 $pbest$ 逐渐收敛于 $gbest$ 时,根据两者的距离,产生变异位。该策略能有效避免陷入局部最优,增强算法的全局搜索能力。同时,本文采用非线性动态惯性权重策略(AW),使惯性权重随着粒子的目标函数值而自动改变。在 14 个典型的测试函数上进行的数值实验中,本文将算法与多种基于反向学习的 PSO 算法进行了对比。实验结果表明,本文算法在大部分测试函数上取得最优解的同时,大大提高了 PSO 的收敛速度。

2 相关工作

2.1 基本 PSO

PSO 算法是一种基于群体智能的全局随机搜索算法,群体中的每个粒子表征一个候选解,具有速度与位置两个属性。群体中的个体看作是在 D 维搜索空间中无质量和体积的粒子,每个粒子以一定的速度在解空间中运动,并向自身最佳位置($pbest$)及领域历史最佳位置($gbest$)移动,实现对候选解的进化。在 D 维空间中,设群体规模为 N ,第 i 个粒子 $x_i(t) = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ 在第 j 维的速度分量 $v_{i,j}$ 和位置分量 $x_{i,j}$ 在下次迭代中的更新公式如下^[2]:

$$v_{i,j}(t+1) = \omega v_{i,j}(t) + c_1 rand_1(pbest_{i,j} - x_{i,j}(t)) + c_2 rand_2(gbest - x_{i,j}(t)) \quad (1)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \quad (2)$$

其中, $i = 1, 2, \dots, N; j = 1, 2, \dots, D$; ω 是惯性权重(inertia weight), $\omega \in [0, 1]$; c_1 和 c_2 为加速系数(acceleration coefficients), $c_1, c_2 \in [0, 2]$; $rand_1$ 和 $rand_2$ 是分布在 $[0, 1]$ 区间上的均匀随机数。

2.2 反向学习策略

反向学习策略(Opposition-Based Learning, OBL)由 H. R. Tizhoosh 在 2005 年首次提出^[3]。在反向学习中,每个粒子 $x_i(t) = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ 的反向解 $\tilde{x}_j = (\tilde{x}_{j,1}, \tilde{x}_{j,2}, \dots, \tilde{x}_{j,D})$ (D 为粒子维度)由式(3)给定。

$$\tilde{x}_j = a_j + b_j - x_j \quad (3)$$

其中, $[a_j, b_j]$ 为 x_j 的边界。由于当前粒子的反向解比原粒子本身较优的可能性在概率上占 50%,因此,如将原粒子所构造的解空间与其反向解空间进行合并,那么在合并空间中进行搜索,获取最优解的概率将大大提高。

Wang 等人将 OBL 策略应用到粒子群算法中,提出一种基于反向学习的 PSO 算法(OPSO)。在 OPSO 中,应用柯西变异(Cauchy Mutation, CM)对 $gbest$ 进行扰动,帮助粒子减小陷入局部最优的可能性^[4]。在文献^[5]中,定义了一个一般化的反向学习策略(Generalized Opposition-Based Learning, GOBL),在 GOBL 下,得到一个增强化的反向学习粒子群算法 GOPSO^[5]。周新宇等人在文献^[6]中提出精英反向学习策略,并将其应用到 PSO 中,得到一种新的精英反向粒子群算法(EOPSO),该算法将粒子在每次迭代中的 $pbest$ 取为精英粒子,求其反向解,在其与原粒子群合并后的 $2N$ 个粒子中求

得适应值最优的前 N 个粒子进入下一代迭代中。同时,周等人提出了差分变异策略(Differential Evolutionary Mutation, DEM)对 $gbest$ 进行扰动^[6]。下面给出一般化反向学习策略的定义。

定义 1(一般化的反向学习策略, Generalized Opposition-based learning, GOBL)^[5,9] 设 $\vec{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ 是 D 维空间中的一个普通粒子,对应的反向解 $\tilde{x}_i = (\tilde{x}_{i,1}, \tilde{x}_{i,2}, \dots, \tilde{x}_{i,D})$, 可定义为:

$$\tilde{x}_{i,j} = k \cdot (da_j + db_j) - x_{i,j} \quad (4)$$

其中, $k \sim U(0, 1)$ 为介于 0 和 1 之间的均匀分布随机数,它使粒子获得更好的反向解^[7], $x_{i,j} \in [a_j, b_j]$, $[da_j, db_j]$ 为第 j 维搜索空间的动态边界,按式(5)计算得到。

$$da_j = \min(x_{i,j}), db_j = \max(x_{i,j}) \quad (5)$$

搜索空间动态边界的设定,使每步迭代生成的反向解始终位于逐渐缩小的搜索空间。为防止反向解跳出可行解边界 $[da_j, db_j]$, 对其取值做进一步判断:

$$\tilde{x}_{i,j} = rand(da_j, db_j), \text{ if } \tilde{x}_{i,j} < da_j \text{ or } db_j > \tilde{x}_{i,j} \quad (6)$$

其中, $rand(da_j, db_j)$ 为区间 $[da_j, db_j]$ 上随机数生成的一个随机数。

3 自适应变异反向学习 PSO

3.1 自适应柯西变异策略(ACM)

对于 PSO,算法在收敛之前,最优粒子 $gbest$ 的选取总是在之前的几个候选粒子之间振荡。因此,增加 $gbest$ 的搜索空间,将有效降低陷入局部最优以及过早收敛的可能性。为此,本文提出一种自适应柯西变异策略(Adaptive Cauchy Mutation, ACM),对每次迭代中的 $gbest$ 进行变异操作,拓展最优解的搜索空间。在 ACM 变异操作中,综合考虑个体极值点与当前全局最优点的距离以及进化代数的关系,分别计算个体自身极值 $pbest$ 到当前最优位 $gbest$ 各维上的距离,距离越小,表示群体间相似性越大,将获得距原 $gbest$ 位越远的变异位。若变异后的 $f(gbest^*)$ 优于 $f(gbest)$ ($f(\cdot)$ 为问题适应值函数),则取代之。同时,ACM 充分考虑到在算法初期, $gbest$ 应获得足够的扰动以增强算法的搜索能力,而在迭代后期,变异率减小可以避免最优解的动荡,从而加快收敛速度。

柯西分布是一个期望和方差均不存在的连续型分布函数,定义如式(7):

$$F(x; x_0, \gamma) = \frac{1}{\pi} \arctan\left(\frac{x - x_0}{\gamma}\right) + \frac{1}{2} \quad (7)$$

其中, x_0 是定义分布峰值位置的位置参数, γ 是最大值一半处的一半宽度的尺度参数。取 $x_0 = 0, \gamma = 1$, 构成如式(8)所示的标准柯西分布函数,分布图如图 1 所示。

$$F(x) = \frac{1}{\pi} \arctan(x) + \frac{1}{2} \quad (8)$$

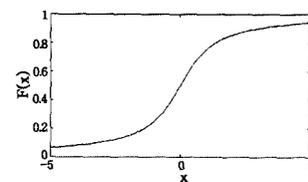


图 1 标准柯西分布函数

从图 1 可见,柯西分布函数是个递增的函数。在 ACM 变异中,分别计算 N 个个体的 $pbest$ 在各维上的平均值 $avg_pbest(i)$:

$$avg_pbest(i) = \left(\sum_{j=1}^N pbest[j][i] \right) / N \quad (9)$$

其中, $i=1, 2, \dots, D$, $pbest[j][i]$ 是第 j 个粒子在第 i 维上的位置。应用式(10)计算 $avg_pbest(i)$ 到 $gbest$ 上每一维的距离:

$$r(i) = gbest(i) - avg_pbest(i) \quad (10)$$

将 $r(i)$ 代入到式(11)中:

$$xm(i) = \exp(-\lambda \cdot t / t_{max}) \cdot (1 - r(i) / r_{max}) \quad (11)$$

其中, λ 是常数,本文中取 $\lambda=10$; t 为进化代数, t_{max} 是最大进化代数; r_{max} 是各维间最大距离。将 xm 代入到式(8)中,对应的关系如图 2 所示。

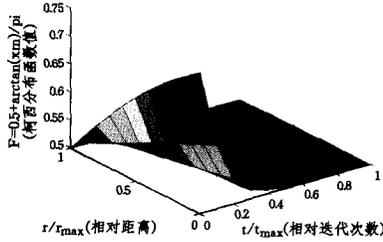


图 2 自适应柯西变异(ACM)三维空间图

从图 2 可知,在迭代初期,个体性能较差,变异值较大可造成足够的扰动,扩大解空间;随着迭代次数的增加,变异值逐渐减小,从而保证问题平滑收敛到最优值。同时,自适应变异在群体极值趋于一致时将获得较大的变异值,增强算法的搜索能力;而在群体搜索空间足够大时,减少变异值以避免最优值的动荡,加快算法收敛速度。

自适应柯西变异(ACM)操作如式(12)所示:

$$gbest^*(i) = gbest(i) + u(i) \cdot F(xm) \quad (12)$$

$$u(i) = \left(\sum_{j=1}^N v[j][i] \right) / N \quad (13)$$

其中, $v[j][i]$ 是第 j 个粒子在第 i 维上的速度分量, $u(i)$ 是各维变异权重平均值。

3.2 自适应更新惯性权重策略(AW)

为进一步平衡 PSO 算法的全局搜索能力和局部探测能力,本文采用非线性的自适应惯性权重系数公式,具体表达式如下^[8]:

$$w = \begin{cases} w_{min} + \frac{(w_{max} - w_{min}) * (f_i - f_{min})}{(f_{avg} - f_{min})}, & f \leq f_{avg} \\ w_{max}, & f > f_{avg} \end{cases} \quad (14)$$

其中, w_{max} 、 w_{min} 分别为 w 的最大值与最小值; f_i 表示粒子 i 的适应值; f_{avg} 、 f_{min} 分别为当前所有粒子的平均适应值和最小适应值。当各粒子的适应值趋于一致或趋于局部最优时,惯性权重增加;比较分散时,惯性权重减小。同时,对于适应值优于平均适应值的粒子,对应惯性权重较小,以保护该占优粒子;反之,取较大惯性权重。使该粒子向较好的搜索领域靠拢。

3.3 ACMOPS

将 ACM 策略应用到反向学习 PSO 中,并采用 AW 策略自适应更新惯性权重。算法 1 给出了 ACMOPS 算法的基本步骤,其中, P 为当前种群, OP 为反向群体。

算法 1 ACMOPS 算法基本步骤

输入:种群大小 N ,惯性权重最大值与最小值 w_{max} 、 w_{min} ,加速系数 c_1 、 c_2 ,迭代次数 M ,粒子维数 D

输出:最优值

1. 随机初始化 N 个粒子种群 P ;
2. 根据式(5)更新 $[da_j, db_j]$;
3. 根据式(4)、式(6)得到 N 个粒子的反向点种群 OP ,选择 $P \cup OP$ 中适应值较优的 N 个粒子作为初始种群;
4. 对于初始种群,if 随机数 $rand < jr$ (若反向学习策略使用概率);
5. 根据式(5)更新 $[da_j, db_j]$,根据式(4)、式(6)求每个粒子的反向点,更新 OP ,在 $P \cup OP$ 所构成的解空间中选取适应值优于其它粒子的前 N 个粒子(本文取适应值最小的前 N 个粒子);
6. else
7. 根据式(14)求得自适应惯性系数,并代入到式(1)、式(2)中,对每个粒子执行基本 PSO 操作;
8. 根据步骤 4—步骤 7 执行结果更新 $pbest$ 与 $gbest$;
9. 对当前获得的 $gbest$ 用 ACM 策略即式(12)进行变异操作;
10. 当算法未达到结束条件时返回步骤 4。

3.4 算法时间复杂度分析

由算法 1 可知,ACMOPS 算法主要包含以下部分:初始种群、GOBL 策略、速度与位置更新操作(基本 PSO)、AW 策略、ACM 策略。易知,速度与位置更新操作和 AW 策略的时间复杂度分别为 $O(N)$ 和 $O(1)$ 。初始种群部分包括随机个体、反向点生成及群体选择机制 3 个部分,GOBL 策略中除包括反向点生成及群体选择机制部分,还包括动态边界更新部分,其中随机个体生成、动态边界更新和反向点生成时间复杂度均为 $O(N \cdot D)$,而群体选择机制中,基本操作为排序,复杂度为 $O(N^2)$ 。对于 GOBL 策略,若维度 D 较小,群体规模 N 可近似于 D ;反之, D 较大时,例如: $D \geq 100$, N 通常小于 D 。因此,GOBL 策略计算复杂度可记为 $O(N \cdot D)$ 。综合上述分析,ACMOPS 的计算复杂度为 $O(N \cdot D)$ 。

4 数值实验及分析

4.1 实验设置

所有算法统一在 MATLAB2012 环境下通过 14 个测试函数进行实验,分为单峰函数 $f_1 \sim f_5$ 和多峰函数 $f_6 \sim f_{14}$ 两类,其中多峰函数按其属性又被分成一般性多峰函数 $f_6 \sim f_8$ 、旋转多峰函数 $f_9 \sim f_{11}$ 和移位多峰函数 $f_{12} \sim f_{14}$ 3 部分。对于移位多峰函数,本文参考 CEC 2010 特别报告^[11]中的函数定义,将 f_{bias} 统一设定为 0,使所有测试函数除 f_5 外,均在零点取得全局最优值 0。由于实验均在维数 $D=30$ 下进行, f_5 的最优值为 -12569.5 , $X \in R^D$ 为搜索空间。具体测试函数见表 1,其中:在旋转多峰函数中, M 为 $D \times D$ 的一个随机正交矩阵;在移位多峰函数中, $o = (o_1, o_2, \dots, o_D)$ 为随机位移动向量。

为了更好地检验算法性能,将 ACMOPS 与与其相近的其它 3 种 OBL-based PSO 算法进行比较,它们分别是 EOPSO^[6]、GOPSO^[5] 和 OPSO^[4]。同时,基本 PSO 也加入到对比中。以上算法参数设置与原文献保持一致;粒子最大速度 v_{max} 为搜索空间的一半;种群规模均取 $N=40$ 。ACMOPS 参数设置如表 2 所列。

表 1 数值实验中使用的 14 个测试函数

	Test Function	D	Search Space	f_{min}	Name of function
Unimodal	$f_1(x) = \sum_{i=1}^D x_i^2$	30	$[-100, 100]^D$	0	Sphere
	$f_2(x) = \sum_{i=1}^D (\sum_{j=1}^i x_j)^2$	30	$[-100, 100]^D$	0	Quadric
	$f_3(x) = \sum_{i=1}^D (\lfloor x_i + 0.5 \rfloor)^2$	30	$[-100, 100]^D$	0	Step
	$f_4(x) = \sum_{i=1}^{D-1} [100 \cdot (x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	30	$[-30, 30]^D$	0	Rosenbrock
	$f_5(x) = \sum_{i=1}^D -x_i \cdot \sin(\sqrt{ x_i })$	30	$[-500, 500]$	-12569.5	Schwefel
Multimodal	$f_6(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]^D$	0	Rastrigin
	$f_7(x) = -20 \cdot \exp(-0.2 \cdot \sqrt{\frac{(\sum_{i=1}^D x_i^2)/D}{e}}) + 20 + \exp(\frac{\sum_{i=1}^D \cos(2\pi x_i)}{D})$	30	$[-32, 32]^D$	0	Ackley
	$f_8(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600, 600]^D$	0	Griewank
	$f_9(x) = f_6(z), z = x * M$	30	$[-5.12, 5.12]^D$	0	Rotated Rastrigin
	$f_{10}(x) = f_7(z), z = x * M$	30	$[-32, 32]^D$	0	Rotated Ackley
	$f_{11}(x) = f_8(z), z = x * M$	30	$[-600, 600]^D$	0	Rotated Griewank
	$f_{12}(x) = f_6(z), z = x - o$	30	$[-5.12, 5.12]^D$	0	Shifted Rastrigin
	$f_{13}(x) = f_7(z), z = x - o$	30	$[-32, 32]^D$	0	Shifted Ackley
	$f_{14}(x) = f_8(z), z = x - o$	30	$[-600, 600]^D$	0	Shifted Griewank

表 2 ACMOPS 参数设置

c_1	c_2	w_{max}	w_{min}	jr	λ
1.49618	1.49618	0.4	0.2	0.3	10

4.2 实验结果与分析

实验总体分成 3 个部分,第一部分将上述 5 种算法进行对比分析;第二部分将详细分析 ACMOPS 算法中的两种自适应策略 ACM 与 AW 各自对算法性能的影响;第三部分对

ACMOPS 算法进行参数敏感性分析。

4.2.1 算法对比分析

在对 5 种 PSO 算法的对比分析中,所有算法在每个测试函数上运行 30 次,每次均迭代 10000 次,记录全局最优值的平均结果,如表 3 所列。符号“+”,“-”,“~”分别表示 ACMOPS 的性能优于、劣于和相当于对比算法。每个测试函数中最好结果用粗体显示。

表 3 PSO, OPSP, GOPSO, EOPSO 和 ACMOPS 算法在 14 个测试函数上的结果均值

Funs.	PSO		OPSO		GOPSO		EOPSO		ACMOPS
f_1	1.56E-03	+	4.59E-36	+	0.00E+00	~	0.00E+00	~	0.00E+00
f_2	7.18E-02	+	2.09E-35	+	3.02e-321	+	1.97e-323	+	0.00E+00
f_3	6.82E-02	+	4.59E+04	+	1.40E+04	+	2.00E-01	+	0.00E+00
f_4	1.26E+00	-	7.18E+00	+	2.82E+01	~	1.57E-24	-	2.37E+01
f_5	-1.72E+02	+	-2.21E+306	+	-1.33E+04	~	-1.99E+05	+	-1.16E+04
f_6	2.06E+00	+	1.51E+01	+	0.00E+00	~	2.09E+00	+	0.00E+00
f_7	4.45E-02	+	1.85E+00	+	0.00E+00	~	1.86E-01	+	0.00E+00
f_8	1.14E-03	+	3.83E-01	+	0.00E+00	~	1.26E-02	+	0.00E+00
f_9	2.99E+00	+	1.51E+01	+	0.00E+00	~	4.11E+00	+	0.00E+00
f_{10}	2.81E-02	+	2.98E+00	+	0.00E+00	-	3.41E-01	+	5.57E-15
f_{11}	7.95E-04	+	2.33E-02	+	0.00E+00	~	1.22E-02	+	0.00E+00
f_{12}	4.68E+00	+	1.35E+01	+	0.00E+00	~	1.76E+00	+	0.00E+00
f_{13}	4.25E-01	+	2.98E+00	+	0.00E+00	-	1.34E-01	+	2.74E-14
f_{14}	7.69E+00	+	1.95E-02	+	0.00E+00	~	1.44E-02	+	0.00E+00
+	13		14		2		12		--
~	0		0		10		1		--
-	1		0		2		1		--

从表 3 中可得出,总体而言,相比于其他 4 种算法,ACMOPS 取得了更好的结果。其中,与 PSO 的比较中,除 f_4 处,ACMOPS 在其它函数上均取得较优解;对于 OPSP 算法,实验结果显示 ACMOPS 在 14 个函数上均优于 OPSP;EOPSO 虽然在 f_4 上取得最优解,但在其它 12 个函数上 ACMOPS 更好。值得注意的是,ACMOPS 与 GOPSO 虽然在大多数函数中取得了相当的最优解,但在 f_2 与 f_3 上,ACMOPS 优于 GOPSO。

表 4 进一步对 ACMOPS 与 GOPSO 两种算法在 14 个测试函数上运行 30 次后的平均值 (Mean)、标准方差 (Std Dev)、

最好值 (Best) 及最差值 (Worst) 进行全面分析,实验结果显示:ACMOPS 多次运行后的 Worst 值在 12 个测试函数中均较优,算法运行结果相对 GOPSO 更稳定。同时,本文将这两种算法在结束条件重新设置为 $fitness(gbest) > 1.0E-16$ & $t < 10000$ (t 为迭代次数)后,再次对 14 个测试函数重新实验,取得运行结束时算法的迭代次数,如表 5 所列,ACMOPS 在 3 个测试函数中劣于 GOPSO,在 9 个测试函数中迭代次数均显著少于算法 GOPSO,进一步说明了本文算法能较快地平稳收敛到全局最优解。

表4 GOPSO 和 ACMOPS算法在 14 个测试函数上的运行结果

Funs.	GOPSO				ACMOPS			
	Mean	Std Dev	Best	Worst	Mean	Std Dev	Best	Worst
f_1	0.00E+00	0.00E+00	0.00E+00	7.41E-323	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_2	3.02e-321	0.00E+00	0.00E+00	9.07e-320	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_3	1.40E+04	1.89E+04	4.94E-110	6.09E+04	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_4	2.82E+01	1.36E+00	2.46E+01	2.90E+01	2.37E+01	8.87E-01	2.21E+01	2.56E+01
f_5	-1.33E+04	2.18E+03	-2.23E+04	-9.68E+03	-1.16E+04	1.81E+03	-1.57E+04	-8.32E+03
f_6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{10}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.57E-15	7.03E-15	0.00E+00	3.20E-14
f_{11}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{12}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{13}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.74E-14	1.22E-13	0.00E+00	6.82E-13
f_{14}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

表5 GOPSO 和 ACMOPS算法在 14 个测试函数上的迭代次数比较

Iterations	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	Less iter
GOPSO	507	559	116	10000	10000	1054	1503	627	713	1392	730	911	929	565	3
AMCOPS	108	119	33	10000	10000	153	3720	136	131	10000	142	121	10000	126	9

4.2.2 策略分析

自适应变异反向学习算法 (ACMOPS) 在反向学习 (GOBL) 的基础上引入了两种自适应策略——自适应柯西变异策略 (ACM) 和自适应惯性策略 (AW) 对 PSO 算法进行了改进。为进一步了解两种自适应策略各自对算法性能优化产生的影响, 以 f_3 与 f_6 为例, 分别将两种策略与 GOBL 进行组合, 即 GOBL+ACM 与 GOBL+AW, 与算法 ACMOPS (= GOBL+ACM+AW) 进行对比实验, 实验结果如图 3 所示。其中: GOBL+ACM 实验中, ω 取 0.72984^[10], AW 策略的引入有效地平衡了算法的全局搜索和局部探测能力, 使算法性能得到显著改进; ACM 策略则有效地降低了 g_{best} 陷入局部最优的可能性, 保证算法平滑快速地收敛到全局最优值。

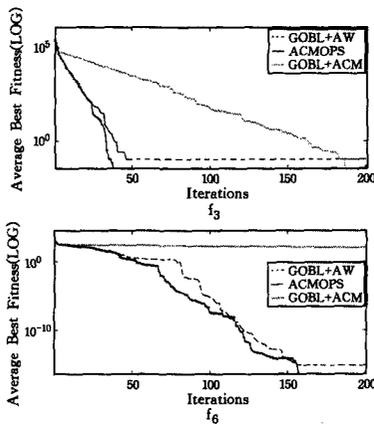


图3 ACM 与 AW 两种自适应策略对算法的影响

4.2.3 参数敏感性分析

ACMOPS 算法通过 GOBL 与 ACM 策略对基本 PSO 操作进行了改进。其中各个参数的设置对算法的性能有很大的影响, 文献[4]中对 OBL 策略概率 jr 给出了参数分析, 指出 $jr=0.3$ 时算法取得最佳性能。在 ACM 策略中, λ 的取值对算法是否能更平滑地快速收敛到最优值具有较大的影响。鉴于篇幅的限制, 图 4 显示了 ACMOPS 在 f_3 与 f_7 中 λ 取不同值时算法收敛到全局最优值的过程。在 f_3 中, 当 $\lambda=10$ 时, 算法能较快地收敛到全局最优解, 这一结论同样可从 f_7 中得到。同时, 从图 4 中的 f_7 图中可见, $\lambda \geq 10$ 时, 算法性能趋于稳定。

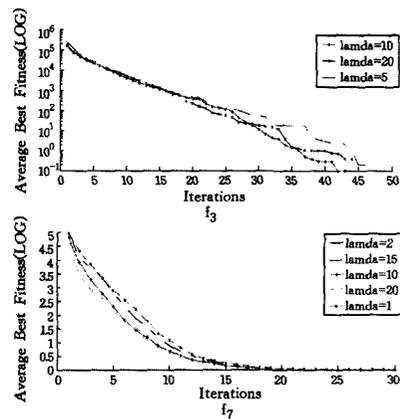


图4 λ (lamda) 在不同取值下 ACMOPS 的全局收敛过程

结束语 为进一步提高反向学习 PSO 算法性能, 提出了一种自适应柯西变异的反向学习 PSO 算法 (ACMOPS)。该算法由一般性反向学习策略 (GOBL) 和自适应柯西变异策略 (ACM) 组成, 并结合自适应惯性权重策略 (AW), 对惯性权重 ω 进行动态更新。其中, GOBL 策略用于生成粒子的反向解, 扩大搜索空间范围, 提高算法全局勘探能力。ACM 策略综合考虑个体极值点与当前全局最优点的距离以及进化代数的关系, 当群体极值趋于一致时获得较大的变异值, 对 g_{best} 造成足够的扰动, 从而有效降低 g_{best} 陷入局部最优的可能性; 随着迭代次数的增加, 变异值逐渐减小, 可避免最优值的动荡, 保证算法平滑快速地收敛到全局最优值。AW 策略的引入可进一步平衡算法的全局搜索和局部探测能力。

将算法在 14 个测试函数上进行实验, 结果表明 ACMOPS 算法在解的精度以及收敛速度上得到了大幅度的提高。但算法在 f_4 和 f_5 中仍无法避免陷入局部最优。如何进一步提高算法性能, 使其在所有测试函数中均取得最优解是下一步需要考虑的问题。

参考文献

[1] Kennedy J, Eberhart R C. Particle swarm optimization[C]// Proceedings of IEEE International Conference on Neural Networks, 1995. Perth, Australia, 1995: 1942-1948
 [2] Shi Y, Eberhart R C. A modified particle swarm optimizer[C]//

- Proceedings of the IEEE Congress on Evolutionary Computation (CEC 1998). Piscataway, NJ, 1998; 69-73
- [3] Tizhoosh H R. Opposition-based learning: A new scheme for machine intelligence [C] // Proceedings of the IEEE International Conference on Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce. 2005; 695-701
- [4] Wang Hui, Li H, Liu Y, et al. Opposition-based particle swarm algorithm with Cauchy mutation [C] // Proceedings of the IEEE Congress on Evolutionary Computation, 2007. Tokyo, 2007; 356-360
- [5] Wang Hui, Wu Zhi-jian, Rahnamayan S, et al. Enhancing particle swarm optimization using generalized opposition-based learning [J]. Information Sciences, 2011, 181(20): 4699-4714
- [6] 周新宇, 吴志健, 王晖, 等. 一种精英反向学习的粒子群优化算法 [J]. 电子学报, 2013, 41(8): 1647-1652
Zhou X Y, Wu Z J, Wang H, et al. Elite Opposition-Based Particle Swarm Optimization [J]. Acta Electronica Sinica, 2013, 41(8): 1647-1652
- [7] Wang Hui, Wu Z J, Liu Y, et al. Space transformation search: a new evolutionary technique [C] // Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, 2009. New York, USA, 2009; 537-544
- [8] 龚纯, 王正林. 精通 MATLAB 最优化计算 [M]. 电子工业出版社, 2012; 283-285
Gong Chun, Wang Zheng-lin. Proficient optimization calculation in MATLAB [M]. Electronic Industry Press, 2012; 283-285
- [9] 汪慎文, 丁立新, 谢承旺, 等. 应用精英反向学习策略的混合差分演化算法 [J]. 武汉大学学报(理学版), 2013, 59(2): 111-116
Wang Shen-wen, Ding Li-xin, Xie Cheng-wang, et al. A hybrid differential evolution with elite opposition-based learning [J]. J. Wuhan Univ. (Nat. Sci. Ed.), 2013, 59(2): 111-116
- [10] van den Bergh F. An Analysis of Particle Swarm Optimizers [D]. South Africa: Department of Computer Science, University of Pretoria, 2002
- [11] Tang Ke, Li Xiao-dong, Suganthan P N, et al. Benchmark Functions for the CEC' 2010 Special Session and Competition on Large-Scale Global Optimization [R]. Hefei: Nature Inspired Computation and Applications Laboratory, USTC, 2009

(上接第 207 页)

- [4] Institute for Energy Efficiency. Greenscale [EB/OL]. <http://www.iee.ucsb.edu/greenscale>
- [5] Rasmussen N. Determining Total Cost of Ownership for Data Center and Network Room Infrastructure [EB/OL]. http://apcpartnercentral.com/assets/2012/07/CMRP-5T9PQG_R4_EN.pdf
- [6] Global action plan. An inefficient truth [EB/OL]. [2011-02-12]. <http://globalactionplan.org.uk>
- [7] 廖彬, 于炯, 张陶, 等. 基于分布式文件系统 HDFS 的节能算法研究 [J]. 计算机学报, 2013, 36(5): 1047-1064
Liao Bin, Yu Jiong, Zhang Tao, et al. Energy-Efficient Algorithms for Distributed File System HDFS [J]. Chinese Journal of Computers, 2013, 36(5): 1047-1064
- [8] 廖彬, 于炯, 孙华, 等. 基于存储结构重配置的分布式存储系统节能算法 [J]. 计算机研究与发展, 2013, 50(1): 3-18
Liao Bin, Yu Jiong, Sun Hua, et al. Energy-Efficient Algorithms for Distributed Storage System Based on Data Storage Structure Reconfiguration [J]. Journal of Computer Research and Development, 2013, 50(1): 3-18
- [9] 廖彬, 于炯, 张陶, 等. 一种适应节能的云存储系统元数据动态建模与管理方法 [J]. 小型微型计算机系统, 2013, 34(10): 2407-2412
Liao Bin, Yu Jiong, Zhang Tao, et al. Novel Energy-efficient Metadata Dynamic Modeling and Management Approach for Cloud Storage System [J]. Journal of Chinese Computer Systems, 2013, 34(10): 2407-2412
- [10] Gray J. Tape is dead, disk is Tape, flash is disk, RAM locality is king [EB/OL]. http://www.signallake.com/innovation/Flash_is_Good.pdf
- [11] Schall D, Hudlet V, Härder T. Enhancing energy efficiency of database applications using SSDs [C] // Proceedings of the Third C* Conference on Computer Science and Software Engineering. ACM, 2010; 1-9
- [12] Lee S W, Moon B. Design of flash-based DBMS: an in-page logging approach [C] // Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data. ACM, 2007; 55-66
- [13] Lee S W, Moon B, Park C, et al. A case for flash memory ssd in enterprise database applications [C] // Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. ACM, 2008; 1075-1086
- [14] Tsirogiannis D, Harizopoulos S, Shah M A. Analyzing the energy efficiency of a database server [C] // Proc. of SIGMOD '10 Indianapolis, IN, USA, 2010; 231-242
- [15] 王江涛, 赖文豫, 孟小峰. 闪存数据库: 现状、技术与展望 [J]. 计算机学报, 2013, 36(8): 1549-1567
Wang Jiang-tao, Lai Wen-yu, Meng Xiao-feng. Flash-Based Database: Studies, Techniques and Forecasts [J]. Chinese Journal of Computers, 2013, 36(8): 1549-1567
- [16] Xu Z. Building a power-aware database management system [C] // Proceedings of the Fourth SIGMOD PhD Workshop on Innovative Database Research. ACM, 2010; 1-6
- [17] Xu Z, Tu Y C, Wang X. Exploring power-performance tradeoffs in database systems [C] // 2010 IEEE 26th International Conference on Data Engineering (ICDE). IEEE, 2010; 485-496
- [18] Xu Z, Tu Y C, Wang X. PET: reducing database energy cost via query optimization [J]. Proceedings of the VLDB Endowment, 2012, 5(12): 1954-1957
- [19] Rodriguez-Martinez M, valdivia H, Seguel J, et al. Estimating Power/Energy consumption in Database Servers [J]. Procedia Computer Science, 2011, 6: 112-117
- [20] Times N Y. Power, Pollution and the Internet [EB/OL]. [2013-5-20]. <http://www.nytimes.com/2012/09/23/technology/data-centers-waste-vast-amounts-of-energy-belying-industry-image.html>
- [21] Kansal A, Zhao F, Liu J, et al. Virtual machine power metering and provisioning [C] // Proceedings of the 1st ACM Symposium on Cloud Computing. Indianapolis, USA, 2010; 39-50
- [22] Bao Y, Chen M, Ruan Y, et al. HMTT: A platform independent full-system memory trace monitoring system [J]. ACM SIGMETRICS Performance Evaluation Review, 2008, 36(1): 229-240