

# 一种精英反向学习的粒子群优化算法

周新宇<sup>1,2</sup>, 吴志健<sup>1,2</sup>, 王 晖<sup>1,3</sup>, 李康顺<sup>4</sup>, 张浩宇<sup>1</sup>

(1. 武汉大学软件工程国家重点实验室, 湖北武汉 430072; 2. 武汉大学计算机学院, 湖北武汉 430072;  
3. 南昌工程学院信息工程学院, 江西南昌 330099; 4. 华南农业大学信息学院, 广东广州 510642)

**摘 要:** 为解决传统粒子群优化算法易出现早熟的不足, 提出了精英反向学习策略, 引入精英粒子, 采用反向学习生成其反向解, 扩大搜索区域的范围, 可增强算法的全局勘探能力. 同时, 为避免最优粒子陷入局部最优而导致整个群体出现搜索停滞, 提出了差分演化变异策略, 采用差分演化算法搜索最优粒子的邻域空间, 可增强算法的局部开采能力. 在 14 个测试函数上将本文算法与多种知名的 PSO 算法进行对比, 实验结果表明本文算法在解的精度与收敛速度上更优.

**关键词:** 全局优化; 粒子群优化; 精英反向学习; 差分演化变异; 群体选择

**中图分类号:** TP18      **文献标识码:** A      **文章编号:** 0372-2112 (2013) 08-1647-06

**电子学报 URL:** <http://www.ejournal.org.cn>      **DOI:** 10.3969/j.issn.0372-2112.2013.08.031

## Elite Opposition-Based Particle Swarm Optimization

ZHOU Xin-yu<sup>1,2</sup>, WU Zhi-jian<sup>1,2</sup>, WANG Hui<sup>1,3</sup>, LI Kang-shun<sup>4</sup>, ZHANG Hao-yu<sup>1</sup>

(1. State Key Lab of Software Engineering, Wuhan University, Wuhan, Hubei 430072, China;

2. Computer School, Wuhan University, Wuhan, Hubei 430072, China;

3. School of Information Engineering, Nanchang Institute of Technology, Nanchang, Jiangxi 330099, China;

4. School of Information, South China Agricultural University, Guangzhou, Guangdong 510642, China)

**Abstract:** Traditional particle swarm optimization (PSO) algorithm tends to suffer from premature convergence; we proposed an elite opposition-based learning strategy in which elite particles are introduced to generate their opposite solutions by opposition-based learning. This mechanism can expand the search area and is helpful to enhance the global explorative ability of PSO. Meanwhile, a differential evolutionary mutation strategy is presented to avoid the best particle being trapped into local optima, since this may cause search stagnation of the whole swarm. This strategy adopts differential evolution algorithm to search for the neighborhoods of the global best particle and is helpful to enhance the exploitation ability of PSO. We compared our algorithm with some state-of-the-art PSOs on 14 benchmarks, the results show that our algorithm obtains better solution accuracy and quicker convergence speed.

**Key words:** global optimization; particle swarm optimization; elite opposition-based learning; differential evolutionary mutation; population-based selection

## 1 引言

粒子群优化算法 (Particle Swarm Optimization, PSO) 是由 Kennedy 和 Eberhart 等人提出的一种群体智能优化技术<sup>[1]</sup>. PSO 概念简单、易于实现, 已吸引了众多研究人员的兴趣, 并在多个科学与工程领域中得到了成功应用. 虽然 PSO 已表现出了良好的性能, 但在解决一些较复杂的优化问题时, 容易陷入局部最优且收敛速度较慢.

在 PSO 中, 粒子的飞行轨迹主要受自身极值 (pbest) 与全局极值 (gbest) 的影响, 这在很大程度上决定了算法的收敛速度与解的质量. 因而, 如何从这两类极值中挖

掘出更多的有效信息, 来指导粒子“飞入”更好的搜索区域是增强 PSO 性能的关键. 本文提出了一种精英反向学习的 PSO 算法 (Elite Opposition-based PSO, EOPSO), 采用精英反向学习策略 (Elite Opposition-based Learning, EOBL) 来生成精英粒子的反向解, 构造与当前群体对应的反向群体, 该策略有助于搜索更多的有效区域来提高群体的多样性, 增强算法的全局勘探能力. 同时, 提出了一种新的变异策略, 采用差分演化算法搜索最优粒子的邻域空间, 避免最优粒子陷入局部最优而导致整个群体出现搜索停滞, 增强算法的局部开采能力. 在 14 个典型的测试函数上进行数值实验, 将本文算法与多种知名

PSO 算法对比,实验结果表明本文算法在大部分测试函数上更优,有效地提高了 PSO 的性能.

## 2 PSO 算法

PSO 是一种群体随机搜索算法,对一个  $D$  维的优化问题而言,群体中的每个粒子表征一个可行解,具有速度与位置两个属性.在算法的演化过程中,粒子的自身极值(pbest)与群体的全局极值(gbest)均被保存,用于更新粒子的速度与位置.设当前群体  $P(t)$  中包含了  $NP$  个粒子,第  $i$  个粒子  $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$  的第  $j$  维速度分量  $v_{i,j}$  和位置分量  $x_{i,j}$ ,可分别用如下公式(1)和(2)来更新.

$$v_{i,j}(t+1) = \omega v_{i,j}(t) + c_1 r_1 (\text{pbest}_{i,j} - x_{i,j}(t)) + c_2 r_2 (\text{gbest}_j - x_{i,j}(t)) \quad (1)$$

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

其中,  $i = 1, 2, \dots, NP; j = 1, 2, \dots, D; \omega$  是惯性权值,  $\omega \in [0, 1]$ .  $c_1$  和  $c_2$  是学习因子,  $c_1, c_2 \in [0, 2]$ .  $r_1$  和  $r_2$  是  $[0, 1]$  区间上的均匀分布随机数.

## 3 精英反向学习的 PSO 算法

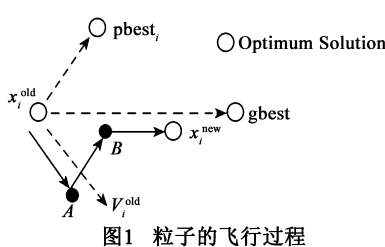
### 3.1 精英反向学习策略

反向学习策略(Opposition-based Learning, OBL)是近年来计算智能领域内出现的一种新技术<sup>[4]</sup>,已应用到了多种优化算法中<sup>[5~8]</sup>,它的主要思想:对一个可行解,同时计算并评估其反向解,从中选择较优的解作为下一代个体.下面给出反向点(可视为反向解)的定义.

**定义 1** 反向点(Opposite Point)<sup>[7]</sup>. 设  $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$  是  $D$  维空间中的一个点(可视为可行解),  $x_j \in [a_j, b_j]$ , 其对应的反向点  $\check{x}_i = (\check{x}_1, \check{x}_2, \dots, \check{x}_D)$  可定义为

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

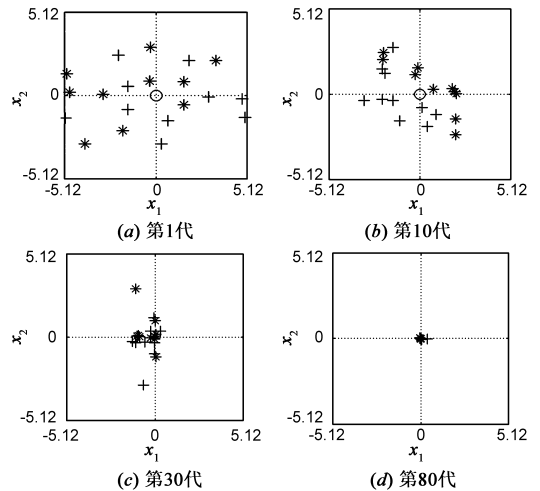
在速度更新公式(1)中,粒子的飞行轨迹由 3 项决定:  $V_i$ ,  $\text{pbest}_i$  和  $\text{gbest}$ . 但对一个给定的粒子  $i$ ,  $V_i$  为已知,则其飞行轨迹是由  $\text{pbest}_i$  和  $\text{gbest}$  共同决定.图 1 描述了粒子的飞行过程,从中可看出  $\text{pbest}_i$  和  $\text{gbest}$  作为搜索空间中的关键点,引导着粒子飞向新的位置.



因而,若用 OBL 策略生成  $\text{pbest}_i$  和  $\text{gbest}$  的反向解,它们是否比粒子自身的反向解更好呢? 为此,进行初步实验,结果证实了  $\text{pbest}_i$  和  $\text{gbest}$  的反向解通常比粒子自身的反向解更优.为突出  $\text{pbest}_i$  和  $\text{gbest}$  的重要作

用,与群体中的粒子(即普通粒子)区分开,将其定义为精英粒子.下面以优化二维 Rastrigrin 函数为例来进一步说明这一现象.

图 2(a)~(d)是用基本 PSO 优化二维 Rastrigrin 函数,在算法的第 1,10,30 和 80 代时,精英粒子与普通粒子的反向解在搜索区域的位置分布,群体规模为 10.图中“+”,“O”,“\*”分别表示普通粒子的反向解,全局最优解,精英粒子的反向解.可看出,在算法的演化过程中粒子群的活动区域不断变化,逐步向最优解靠拢.值得注意的是,大多数精英粒子的反向解比普通粒子的反向解更靠近最优解.因而,若在搜索过程中引入这些精英粒子的反向解,则可拓宽群体的活动区域,提高多样性,有利于避免算法陷入局部最优.下面给出精英反向解的定义.



**定义 2** 精英反向解(Elite Opposite Solution). 设  $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$  是一普通粒子,对应的自身极值即为精英粒子  $X_i^e = (x_{i,1}^e, x_{i,2}^e, \dots, x_{i,D}^e)$ , 则精英反向解  $\check{x}_i^e = (\check{x}_{i,1}^e, \check{x}_{i,2}^e, \dots, \check{x}_{i,D}^e)$  可定义为

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

其中  $x_{i,j}^e \in [a_j, b_j]$ ,  $k \in U(0, 1)$  为一般化系数,  $[da_j, db_j]$  为第  $j$  维搜索空间的动态边界,可按如下公式(5)计算得到.

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

用搜索空间的动态边界代替固定边界,有利于保存搜索经验,使得生成的反向解能够位于逐步缩小的搜索空间中.此外,反向解可能会跳出边界  $[a_j, b_j]$  成为非可行解,对其采用随机生成的方法来重置,如公式(6)所示.

$$\check{x}_{i,j}^e = \text{rand}(a_j, b_j) \quad \text{if } \check{x}_{i,j}^e < a_j \text{ or } b_j > \check{x}_{i,j}^e \quad (6)$$

其中,  $\text{rand}(\cdot)$  是区间  $[a_j, b_j]$  上的一个随机数.

### 3.2 差分演化变异策略

在 PSO 中,群体的最优粒子  $g_{best}$  有重要作用,引导着群体向最优解前进.但若  $g_{best}$  陷入局部最优,则易导致群体出现搜索停滞,可对其进行变异操作,搜索邻域空间,增加跳出局部最优的概率. Wang 等人在文献[7, 8]中,提出了一种柯西变异策略(Cauchy Mutation, CM),采用如下公式(7)对  $g_{best}$  进行变异.

$$g_{best_j}^* = g_{best_j} + \text{cauchy}() \quad (7)$$

其中,  $g_{best_j}$  是最优粒子在第  $j$  维分量,  $\text{cauchy}()$  是服从标准柯西分布的随机数,  $g_{best_j}^*$  为变异后的值. 该策略在一些测试函数上效果良好,如: Schwefel 函数,但对另外一些函数却作用不大.

本文提出了一种差分演化变异策略(Differential Evolutionary Mutation, DEM). 该策略采用差分演化算法对  $g_{best}$  进行扰动. 在差分演化算法中,有多种试验向量生成策略及参数设置<sup>[9]</sup>. 而在 DEM 策略中,需要试验向量生成策略能够使得搜索集中在  $g_{best}$  的邻域空间. 为此,采用“best/2/bin”作为变异策略,如公式(8)所示.

$$m_j = g_{best_j} + F \cdot (X_{r1,j} - X_{r2,j}) + F \cdot (X_{r3,j} - X_{r4,j}) \quad (8)$$

其中,  $F$  是尺度系数,  $r1, r2, r3$  和  $r4$  是区间  $[1, NP]$  上互不相同的随机整数,代表不同粒子的下标,  $j$  为维度.  $m_j$  是扰动后的值. 然后,采用如下二项式交叉操作生成试验向量  $g_{best}^*$ .

$$g_{best_j}^* = \begin{cases} m_j, & \text{if } \text{rand} \leq CR \text{ or } j_{\text{rand}} = j \\ g_{best_j}, & \text{otherwise} \end{cases} \quad (9)$$

其中,  $\text{rand}$  是  $[0, 1]$  间的均匀分布随机数,对每一维都需重新生成.  $CR$  为交叉概率,  $j_{\text{rand}}$  是区间  $[1, D]$  上随机生成的一整数,保证  $g_{best}^*$  至少有一维不同于  $g_{best}$ . 实验中,参数  $F, CR$  分别设为 1 和 0.1.

### 3.3 EOPSO

在 EOBL 策略中,还需解决如何从当前群体与反向群体中选择出  $NP$  个解作为下一代个体. 本文采用群体选择机制,将当前群体与反向群体的个体按适应值排序,选择最好的  $NP$  个进入下一代. 算法 1 用伪代码描述了 EOPSO 的基本步骤,  $P$  为当前群体,  $\text{rand}$  是服从  $U(0, 1)$  的随机数,  $JR$  是使用 EOBL 策略的概率,  $OP$  为反向群体.

#### 算法 1 EOPSO 算法的基本步骤.

Randomly initialize  $NP$  particles in the swarm  $P$ ;

while stopping criterion is not meet do

if  $\text{rand} \leq JR$  then

Update  $[da_j, db_j]$  according to Eq. (5);

Generate the opposite solution of each  $X_i^c$  in the current swarm  $P$  to construct  $OP$  by Eq. (4);

Select the  $NP$  best fittest solutions from  $P \cup OP$  as the next

swarm by population-based selection mechanism;

end

else

Execute the canonical PSO according to Eqs. (1)and(2);

end

Execute the DEM strategy on  $g_{best}$  by Eqs. (8)and(9);

end

## 4 数值实验及分析

### 4.1 测试函数

实验中采用表 1 中所列的 14 个测试函数,按其属性分为三类:单峰及简单的多峰函数( $f_1 \sim f_2$ ),非旋转的多峰函数( $f_3 \sim f_8$ ),带旋转的多峰函数( $f_9 \sim f_{14}$ ),这些函数的全局最优值均为 0,维度  $D = 30$ . 表中,  $X \in R^n$  为定义域,“Threshold”为解的精度阈值,用于判断算法是否运行成功.

表 1 数值实验使用的 14 个测试函数

Funs.	Name	$X$	Threshold
$f_1$	Sphere	$[-100, 100]$	1.00E-06
$f_2$	Rosenbrock	$[-2.04, 2.04]$	1.00E+01
$f_3$	Ackley	$[-32.7, 32.7]$	1.00E-06
$f_4$	Griewanks	$[-600, 600]$	1.00E-06
$f_5$	Weierstrass	$[-0.5, 0.5]$	1.00E-06
$f_6$	Rastrigin	$[-5.12, 5.12]$	1.00E-06
$f_7$	Nocontinuous Rastrigin	$[-5.12, 5.12]$	1.00E-06
$f_8$	Schwefel	$[-500, 500]$	2.00E-06
$f_9$	Rotated Ackley	$[-32.7, 32.7]$	1.00E-06
$f_{10}$	Rotated Griewanks	$[-600, 600]$	1.00E-06
$f_{11}$	Rotated Weierstrass	$[-0.5, 0.5]$	1.00E-06
$f_{12}$	Rotated Rastrigin	$[-5.12, 5.12]$	1.00E+01
$f_{13}$	Rotated Nocontinuous Rastrigin	$[-5.12, 5.12]$	1.00E-06
$f_{14}$	Rotated Schwefel	$[-500, 500]$	3.00E+03

### 4.2 与 OBL-based PSO 算法的比较

将 EOPSO 与其他两种 OBL-based PSO 算法进行比较,分别是 OPPO<sup>[8]</sup>和 GOPSO<sup>[9]</sup>. OPPO 采用 OBL 策略和 CM 策略,而 GOPSO 采用了一般反向学习策略和 CM 策略. 另外,基本 PSO 也加入对比. 在 EOPSO 中,参数  $c_1 = c_2 = 1.193$ ,  $\omega = 0.721$ ,  $JR = 0.3$ ,粒子最大速度  $V_{\max}$  为搜索空间的一半,  $NP$  为 40. 基本 PSO, OPPO 和 GOPSO 的参数设置与原文献相同. 对所有算法,适应值最大评估次数设为 200000,在每个测试函数上运行 30 次,记录结果的均值. 为判断 EOPSO 与 3 种对比算法的性能是否存在显著性差异,采用双尾  $t$  检验对结果进行统计分析,显著性水平设为 0.05,自由度为 58,符号“+”,“-”和“ $\approx$ ”分别表示 EOPSO 的性能要优于,劣于和相当于对比算法. 表 2 给出 4 种算法的实验结果,最好值用粗体显示.

从表 2 可得出 EOPSO 是 4 种算法中性能最好的,在所有函数上均优于 OPPO. 与 PSO 相比,除  $f_3$  外, EOPSO 在其

他函数上更优.虽然在  $f_3$  上, GOPSO 为最优, 但 EOPSO 在 8 个函数上更好. 图 4 描述了上述 4 种 PSO 算法的收敛曲线, 受篇幅限制, 仅给出了两个代表性的函数.

表 2 PSO, OPSP, GOPSO 和 EOPSO 算法在 14 个测试函数上的结果均值

Funs.	PSO	OPSO	GOPSO	EOPSO
$f_1$	9.78E-30+	8.05E-81+	1.24E-272+	<b>0.00E+00</b>
$f_2$	2.93E+01+	9.78E+00+	1.48E+01+	<b>3.74E-02</b>
$f_3$	3.94E-14≈	1.39E+00+	<b>3.43E-15</b> ≈	3.99E-15
$f_4$	8.13E-03+	1.27E-02+	<b>0.00E+00</b> ≈	<b>0.00E+00</b>
$f_5$	1.30E-04+	2.21E+00+	1.04E-08+	<b>7.36E-10</b>
$f_6$	2.90E+01+	5.23E+01+	<b>0.00E+00</b> ≈	<b>0.00E+00</b>
$f_7$	2.97E+01+	8.64E+01+	<b>0.00E+00</b> ≈	<b>0.00E+00</b>
$f_8$	1.10E+03+	1.45E+03+	1.14E+03+	<b>4.01E+02</b>
$f_9$	2.80E-01+	1.89E+00+	9.59E-13+	<b>1.63E-15</b>
$f_{10}$	1.64E-01+	4.37E-02+	<b>0.00E+00</b> ≈	<b>0.00E+00</b>
$f_{11}$	6.66E-01+	8.65E+00+	<b>2.64E-13</b> -	1.17E-09
$f_{12}$	9.90E+00+	9.02E+01+	<b>0.00E+00</b> ≈	<b>0.00E+00</b>
$f_{13}$	6.32E+01+	1.86E+02+	1.89E+02+	<b>9.78E+00</b>
$f_{14}$	2.67E+03+	2.63E+03+	2.69E+03+	<b>2.17E+03</b>
+	13	14	8	--
-	0	0	1	--
≈	1	0	6	--

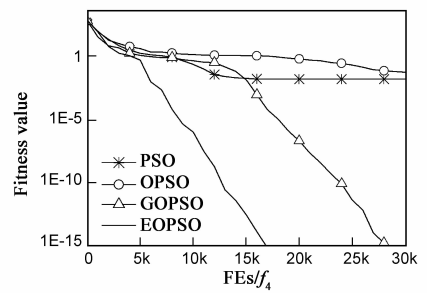
图 3(a) 是在  $f_4$  上的收敛曲线, 虽然 GOPSO 和 EOPSO 都找到了最优解, 但 EOPSO 收敛更快. 在  $f_8$  上, EOPSO 整体上比其他算法收敛更快. EOPSO 在解的精度和收敛速度上具有优良性能的原因是: (1) 精英反向解有助于勘探更多的优质搜索区域, 提高群体的多样性; (2) DEM 策略有利于增强算法的开采能力, 而群体选择策略可加快收敛速度. 总之, EOPSO 期望能平衡算法的勘探与开采能力.

### 4.3 与知名 PSO 算法的比较

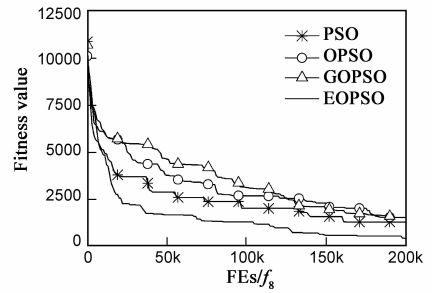
本节中, 将 EOPSO 与 5 种知名 PSO 算法对比, 分别是: UPSO<sup>[11]</sup>, FDR-PSO<sup>[12]</sup>, FIPS<sup>[13]</sup>, CPSO-H<sup>[14]</sup> 和 CLP-

表 3 UPSO, FDR-PSO, FIPS, CPSO-H, CLPSO 和 EOPSO 算法在 14 个测试函数上的结果均值

Funs.	USPO	FDR-PSO	FIPS	CPSO-H	CLPSO	EOPSO
$f_1$	4.17E-87+	4.88E-102≈	2.69E-12+	1.16E-13+	1.46E-14+	<b>0.00E+00</b>
$f_2$	1.51E+01+	5.39E+00+	2.45E+01+	7.08E+00+	2.01E+01+	<b>3.74E-02</b>
$f_3$	1.22E-15-	2.84E-14+	4.81E-07+	4.93E-14+	<b>0.00E+00</b> -	3.99E-15
$f_4$	1.66E-03+	1.01E-02+	1.16E-06+	3.63E-02+	3.14E-10+	<b>0.00E+00</b>
$f_5$	9.60E+00+	7.49E-03+	1.54E-01+	<b>7.82E-15</b> -	3.45E-07+	7.36E-10
$f_6$	6.59E+01+	2.84E+01≈	7.30E+01+	<b>0.00E+00</b> ≈	4.85E-10+	<b>0.00E+00</b>
$f_7$	6.34E+01+	1.44E+01≈	6.08E+01+	1.00E-01+	4.36E-10+	<b>0.00E+00</b>
$f_8$	4.84E+03+	3.61E+03+	2.05E+03+	1.08E+03+	<b>1.27E-12</b> -	4.01E+02
$f_9$	1.00E+00+	1.40E-01≈	2.25E-15≈	1.36E+00+	3.65E-05≈	<b>1.63E-15</b>
$f_{10}$	7.76E-02+	1.44E-01+	1.70E-01+	1.20E-01+	4.50E-02+	<b>0.00E+00</b>
$f_{11}$	2.61E+00+	3.34E-01+	5.93E-14-	4.35E+00+	3.72E-10-	1.17E-09
$f_{12}$	1.52E+01+	9.25E+00+	1.20E+01+	2.67E+01+	5.97E+00+	<b>0.00E+00</b>
$f_{13}$	7.74E+01+	7.58E+01+	4.36E+01+	8.80E+01+	3.77E+01+	<b>9.78E+00</b>
$f_{14}$	5.60E+03+	2.60E+03+	3.78E+03+	3.64E+03+	<b>1.70E+03</b> -	2.17E+03
+	13	10	12	12	9	--
-	1	0	1	1	4	--
≈	0	4	1	1	1	--



(a) Griewanks function ( $f_4$ )



(b) Schwefel function ( $f_8$ )

图 3 PSO, OPSP, GOPSO 和 EOPSO 算法在函数  $f_4$  和  $f_8$  上的收敛曲线

SO<sup>[10]</sup>. UPSO 是统一的 PSO 算法, 它集成了全局与局部版本的 PSO 算法. 在 FDR-PSO 中, 粒子的速度更新中除了 pbest 和 gbest 外, 还采用了具有较好适应值和距离较近的邻域粒子, 即适应值与距离比率最高的粒子. FIPS 是信息互通的 PSO 算法, 用邻域粒子的信息来更新速度. CPSO-H 是一种协作的 PSO 算法, 它的主要思想是将搜索空间划分成多个子空间再搜索. CLPSO 利用其他粒子的 pbest 代替 gbest 来指导粒子的速度更新, 可提高群体的多样性. EOPSO 的参数设置与 4.2 节相同, 而上述 5 种对比 PSO 算法直接采用文献[10]中的设置. 表 3 给出了 6 种 PSO 算法的实验结果.

从表 3 可看出, EOPSO 在 13 个函数上均优于 UPSO. 与 FDR-PSO 相比, EOPSO 在所有函数上的结果均更优或相当. FIPS 在  $f_{11}$  上表现最优, 但 EOPSO 在 12 个函数上更优. 在  $f_6$  上, CPSO-H 与 EOPSO 表现相同, 均找到了最优解, 但 EOPSO 在 12 个函数上更优. 与 CLPSO 相比, EOPSO 在 9 个函数上更好. 为在统计意义上比较多个算法的性能, 采用 Friedman 检验对结果进行分析, 表 4 给出了 6 种 PSO 算法的平均排名, EOPSO 为第一.

表 4 6 种 PSO 算法的平均排名

Algorithms	Rankings
EOPSO	1.61
CLPSO	2.39
FDR-PSO	3.61
CPSO-H	4.25
FIPS	4.43
UPSO	4.71

#### 4.4 DEM 策略与算法计算复杂度分析

为研究 DEM 策略的有效性, 将不含变异策略的 EOPSO (EOPSO-NULL) 和集成 CM 策略的 EOPSO (EOPSO-CM) 作为对比算法. 实验中记录算法 30 次运行的变

异策略成功次数的均值 (Mean Number of Successful Mutation, MNSM). 若变异后的 gbest 值更优, 则认为变异策略成功运行. 此外, 也记录算法在 30 次运行中的成功率 (Successful Rate, SR%), 若算法的最终解小于表 1 中规定的阈值, 则认为算法成功运行. 表 5 给出了实验结果, 其中“Mean”代表结果的均值.

从表 5 可看出, 除  $f_8$  和  $f_{14}$  外, EOPSO-DEM 在大部分函数上均比 EOPSO-CM 的 MNSM 值更高, 表示 DEM 策略效果更好. 然而, 在  $f_1, f_4$  和  $f_{10}$  上, EOPSO-DEM 和 EOPSO-CM 的结果与不带变异策略的 EOPSO-NULL 相同, 原因是 EOPSO 采用的 EOBL 策略可以直接找到最优解. 在  $f_8$  上, EOPSO-DEM 和 EOPSO-CM 的实验结果明显好于 EOPSO-NULL. 但在  $f_8$  的旋转版本, 即  $f_{14}$  上, EOPSO-CM 和 EOPSO-NULL 的结果类似, 表明 CM 策略在搜索空间复杂化后效果不佳, 与之不同的是 DEM 策略的鲁棒性和稳定性更好.

表 5 不同变异策略的比较结果

Funcs.	EOPSO-NULL			EOPSO-CM			EOPSO-DEM		
	Mean	SR%	MNSM	Mean	SR%	MNSM	Mean	SR%	MNSM
$f_1$	0.00E+00	100	--	0.00E+00	100	4.26E+00	0.00E+00	100	3.35E+02
$f_2$	7.83E-02	100	--	1.18E-01	100	2.00E-01	3.74E-02	100	6.90E+02
$f_3$	3.99E-15	100	--	3.99E-15	100	1.73E+00	3.99E-15	100	3.04E+01
$f_4$	0.00E+00	100	--	0.00E+00	100	1.15E+01	0.00E+00	100	2.34E+01
$f_5$	9.09E-10	100	--	1.30E-09	100	5.30E-01	7.36E-10	100	1.77E+01
$f_6$	1.93E+01	53.33	--	9.45E+00	60	6.00E-02	0.00E+00	100	1.79E+02
$f_7$	4.42E+01	26.67	--	9.17E+00	20	2.00E-01	0.00E+00	100	1.24E+02
$f_8$	6.65E+03	0	--	1.69E+03	46.67	8.74E+01	4.01E+02	100	8.16E+01
$f_9$	3.99E-15	100	--	3.99E-15	100	2.00E+00	1.63E-15	100	2.98E+01
$f_{10}$	0.00E+00	100	--	0.00E+00	100	1.20E+01	0.00E+00	100	2.41E+01
$f_{11}$	5.62E-09	100	--	5.17E-09	100	2.00E-01	1.17E-09	100	1.90E+01
$f_{12}$	2.01E+01	40	--	4.87E+01	20	2.00E-01	0.00E+00	100	2.54E+02
$f_{13}$	4.80E+01	0	--	3.20E+01	0	0.00E+00	9.78E+00	70	7.58E+01
$f_{14}$	4.93E+03	20	--	4.69E+03	0	1.82E+02	2.17E+03	93.33	8.58E+01

从算法 1 可看出, EOPSO 主要包含了 4 个部分: 初始化, EOBL 策略, 速度与位置更新, DEM 策略. 不难得出, 初始化、速度与位置更新有相同的复杂度  $O(NP \cdot D)$ , DEM 策略为  $O(D)$ . EOBL 策略可分为 3 个部分: 动态边界更新, 反向解生成, 群体选择机制. 动态边界更新与反向解生成的复杂度相同, 均为  $O(NP \cdot D)$ . 但在群体选择机制中, 主要操作为排序, 复杂度为  $O(NP^2)$ . 在 EOBL 策略中, 若维度  $D$  较小时, 群体规模  $NP$  近似

于  $D$ ; 但若  $D$  较大时, 例如:  $D \geq 100$ ,  $NP$  通常小于  $D$ ; 所以其复杂度可计为  $O(NP \cdot D)$ . 综上所述, EOPSO 的计算复杂度为  $O(NP \cdot D)$ . 表 6 给出了本文实验中出现的所有对比算法的计算复杂度, 其中  $I$  表示邻居数,  $K$  是解被划分成的片段数<sup>[15]</sup>. 可得出, EOPSO 与 PSO, OPSO, GOPSO, FIPS, CLPSO 有相同的复杂度, 小于 UPSO, FDR-PSO 和 CPSO-H.

表 6 9 种 PSO 算法的计算复杂度

Algos.	PSO	OPSO	GOPSO	UPSO	FDR-PSO	FIPS	CPSO-H	CLPSO	EOPSO
$O(\cdot)$	$NP \cdot D$	$NP \cdot D$	$NP \cdot D$	$NP \cdot I \cdot D$	$NP^2 \cdot D$	$NP \cdot D$	$NP \cdot K \cdot D$	$NP \cdot D$	$NP \cdot D$

## 5 结论

本文提出了一种精英反向学习的 PSO 算法(EOPSO),包含了精英反向学习策略(EOBL)和差分演化变异策略(DEM).EOBL策略用于生成精英粒子的反向解,可扩大搜索的区域范围,提高种群多样性,增强 PSO 的全局勘探能力.DEM策略用于对 gbest 进行变异操作,搜索其邻域空间,防止 gbest 陷入局部最优,可增强 PSO 的局部开采能力.在 14 个测试函数上开展实验,将 EOPSO 与多种知名 PSO 算法进行对比,实验结果表明 EOPSO 在解的精度与收敛速度上具有更优的性能.

### 参考文献

- [1] Z Ji, Y Wang, Y Chu, Q Wu. Bacterial particle swarm optimization[J]. Chinese Journal of Electronics, 2009, 18(2): 195 - 199.
- [2] 吴晓军, 杨战中, 赵明. 均匀搜索粒子群算法[J]. 电子学报, 2011, 39(6): 1261 - 1266.  
Wu Xiao-jun, Yang Zhan-zhong, Zhao Ming. A uniform searching particle swarm optimization algorithm[J]. Acta Electronica Sinica, 2011, 39(6): 1261 - 1266. (in Chinese)
- [3] 李文锋, 梁晓磊, 张煜. 具有异构分簇的粒子群优化算法研究[J]. 电子学报, 2012, 40(11): 2194 - 2099.  
Li Wen-feng, Liang Xiao-lei, Zhang Yu. Research on PSO with clusters and heterogeneity[J]. Acta Electronica Sinica. 2012, 40(11): 2194 - 2099. (in Chinese)
- [4] H R Tizhoosh. Opposition-based learning: a new scheme for machine intelligence[A]. Proc of the IEEE International Conference of Intelligent Agents, Web Technologies and Internet Commerce[C]. Vienna, IEEE Press. 2005. 695 - 701.
- [5] S Rahnamayan, H R Tizhoosh, M M A Salama. Opposition-based differential evolution[J]. IEEE Transaction on Evolutionary Computation, 2008, 12(1): 64 - 79.
- [6] Hui Wang, Zhijian Wu, S Rahnamayan. Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems[J]. Soft Computing-A Fusion of Foundations, Methodologies and Applications, 2011, 15(11): 2127 - 2140.
- [7] Hui Wang, Zhijian Wu, S Rahnamayan, Yong Liu, Mario Ventresca. Enhancing particle swarm optimization using generalized opposition-based learning[J]. Information Sciences, 2011, 181(20): 4699 - 4714.
- [8] H Wang, H Li, Y Liu, C Li, S Zeng. Opposition-based particle swarm algorithm with cauchy mutation[A]. Proc of IEEE Congress on Evolutionary Computation[C]. Tokyo, IEEE Press, 2007. 356 - 360.
- [9] S Das, P N Suganthan. Differential evolution; a survey of the State-of-the-Art[J]. IEEE Transactions on Evolutionary Com-

putation, 2011, 15(1): 4 - 31.

- [10] J J Liang, A K Qin, P N Suganthan, S Baskar. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281 - 295.
- [11] K E Parsopoulos, M N Vrahatis. UPSO-a unified particle swarm optimization scheme[A]. Proc of Computational Methods in Sciences and Engineering[C]. Zeist, VSP International Science Publishers; 2004. 868 - 873.
- [12] T Peram, K Veeramachaneni, C K Mohan. Fitness-distance-ratio based particle swarm optimization[A]. Proc of IEEE Swarm Intelligence Symposium[C]. Indianapolis, IEEE Press. 2003. 174 - 183.
- [13] R Mendes, J Kennedy, J Neves. The fully informed particle swarm: simpler, maybe better[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 204 - 210.
- [14] F van den Bergh, A P Engelbrecht. A cooperative approach to particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 225 - 239.
- [15] Y Shi, H Liu, L Gao, G Zhang. Cellular particle swarm optimization[J]. Information Sciences, 2011, 181(20): 4460 - 4493.

### 作者简介



**周新宇(通信作者)** 男, 1987年生, 武汉大学计算机学院博士研究生. 研究方向: 智能计算、并行计算.

E-mail: xyzhou@whu.edu.cn



**吴志健** 男, 1963年生, 教授, 博士生导师. 武汉大学软件工程国家重点实验室副主任. 研究方向: 智能计算、并行计算、智能信息处理.

E-mail: zhijianwu@whu.edu.cn

**王晖** 男, 1983年生, 博士. 南昌工程学院信息工程学院讲师. 研究方向: 智能计算、并行计算.

**李康顺** 男, 1962年生, 教授, 博士生导师. 华南农业大学信息学院、软件学院院长. 研究方向: 智能计算、多目标优化、视频流图像识别.

**张浩宇** 女, 1987年生, 武汉大学软件工程国家重点实验室硕士生. 研究方向: 智能计算、并行计算.