

# PSO 的向量整体修订策略和局部跳出策略

窦全胜<sup>1,2</sup>, 潘冠宇<sup>3</sup>, 刘 岩<sup>1</sup>, 周春光<sup>4</sup>, 史忠植<sup>2</sup>

(1. 山东工商学院 计算机科学与技术学院, 山东 烟台 264005; 2. 中国科学院 计算技术研究所, 北京 100080; 3. 吉林大学 数学学院, 长春 130012; 4. 吉林大学 计算机科学与技术学院, 长春 130012)

**摘 要:** 针对传统 PSO 方法对 CEC2005(The 2005 IEEE Congress on evolutionary computation) 中的 25 个 benchmark 函数搜索效果较差的问题, 提出了“向量整体修订”和“局部跳出”两种改进策略。改变 PSO 方法中粒子在每一维上的修订相互独立的传统机制, 按某一概率将粒子作为整体进行修正, 当群体最优长时间不变或变化值小于一定阈值时, 为跳出局部最优, 按某一概率重新定义群体最优或初始化群体。通过实验证明了改进后的 PSO 方法对 CEC2005 中的测试问题的有效性。

**关键词:** 计算机应用; 粒子群优化; 收敛; 向量修订; 局部跳出

**中图分类号:** TP301.6      **文献标志码:** A      **文章编号:** 1671-5497(2012)02-0429-05

## Vector correction and jump out of local optimum strategy for PSO

DOU Quan-sheng<sup>1,2</sup>, PAN Guan-yu<sup>3</sup>, LIU Yan<sup>1</sup>, ZHOU Chun-guang<sup>4</sup>, SHI Zhong-zhi<sup>2</sup>

(1. School of Computer Science and Technology, Shandong Institute of Business and Technology, Yantai 264005, China; 2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China; 3. College of Mathematics, Jilin University, Changchun 130012, China; 4. College of Computer Science and Technology, Jilin University, Changchun 130012, China)

**Abstract:** Using traditional Particle Swarm Optimization (PSO) the searching results for some new benchmark functions, e. g. the 25 benchmark functions in CEC2005, are not satisfactory. An improved version of PSO was designed to suit for new benchmark functions in CEC2005. Two improvement strategies, named Vector correction strategy and Jump out of local optimum strategy, were employed in this improved PSO. When the swarm optimum remains invariable for a long time, The improve PSO can revise the whole particle vector and re-initialize the swarm or generate a new swarm optimum according to certain probability. The improved PSO was tested by the 25 benchmark functions in CEC2005, and the experimental results show that the search efficiency and the ability to jump out of the local optimum of the improved PSO are significantly improved.

**Key words:** computer application; partical swarm optimization; convergence; vector correction; jump out of local optimum

**收稿日期:** 2011-03-19.

**基金项目:** 国家自然科学基金项目(61173173, 60970105, 60970088, 61035003, 60933004, 60172085); “973”国家发展规划项目(2007CB311004); 山东省中青年科学家奖励基金(2009BSD01383).

**作者简介:** 窦全胜(1971-), 男, 副教授, 博士. 研究方向: 智能科学, 数据挖掘. E-mail: douqs@ics.ict.ac.cn

粒子群优化(PSO)方法是由 Kennedy 和 Eberhart 于 1995 年提出的一种演化计算技术<sup>[1-2]</sup>,广泛应用于各类优化问题<sup>[3-7]</sup>。但对于一些新问题,如 CEC2005<sup>[8-9]</sup>(The 2005 IEEE Congress on evolutionary computation)中提出的 25 个 benchmark 问题,PSO 方法搜索效果不佳。本文对造成这种情况的原因进行了分析,提出了向量修订策略和局部跳出策略,实验证明了改进后的 PSO 较好地适应形如 CEC2005 中的 benchmark 问题,为解决此类优化问题提供了切实可行的方法。

## 1 问题概述

传统 PSO 方法将问题的潜在解看成是搜索空间中的一个“粒子”。粒子每次迭代的位置由式(1)确定:

$$\begin{cases} v_{t+1} = \omega v_t + c_1 r_1 (p_i - x_t) + c_2 r_2 (p_g - x_t) \\ x_{t+1} = x_t + v_{t+1} \end{cases} \quad (1)$$

式中:  $x_t$  表示粒子当前的位置;  $v_t$  表示粒子当前速度;  $p_i$  为粒子  $i$  经过的最好位置;  $p_g$  为群体中所有粒子经过的最好位置;  $c_1, c_2$  为正常数;  $r_1, r_2$  为  $[0, 1]$  区间中满足正态分布的随机数;  $\omega$  为惯性参数。

在 CEC2005 会议上,提出了如下形式的优化问题:

$$\min\{f[(x - o) * M] + f\_bias\} \quad (2)$$

这些测试问题是在传统 benchmark 函数的基础上经过偏移、旋转等操作得到的。式(2)中,  $f()$  为传统的 benchmark 函数,  $o$  为平移向量,  $M$  为旋转矩阵,  $f\_bias$  为偏移参数。实践证明<sup>[9]</sup>,传统的 PSO 方法不能很好地适应这一问题,搜索效果较差。从式(1)可以看出,传统 PSO 对每个维度的修订都是独立的,其中速度  $v$  充当着修订向量的角色,当各维度之间互不关联时,这种修订是可行的,个体在群体最优  $p_g$  和个体最优  $p_i$  的启发下,每一步都是向着最有希望发现最优解的区域移动。然而当搜索空间被旋转之后,这种策略显然是无效的。以二维空间为例,不妨设个体最优  $p_i$ 、群体最优  $p_g$  的坐标分别为  $(x_i, y_i)$  和  $(x_g, y_g)$ , 旋转矩阵

$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \quad (3)$$

向量旋转后,问题的搜索空间各维度之间相

互关联,旋转后的个体最优  $M(p_i)$  和群体最优  $M(p_g)$  的坐标分别为  $(m_{11}x_i + m_{21}y_i, m_{12}x_i + m_{22}y_i)$  和  $(m_{11}x_g + m_{21}y_g, m_{12}x_g + m_{22}y_g)$ 。由式(1)不难看出,修订向量与经过旋转后得到的实际的个体最优  $M(p_i)$  和群体最优  $M(p_g)$  没有关系,这也是传统的 PSO 对于这样的问题无效的原因。

## 2 整体修正策略和局部跳出策略

### 2.1 整体修正策略

传统 PSO 方法的每一步骤  $t$ , 都按如下方式对粒子的位置进行修订:

```
For i=1 to Dims
    vt+1(i) = ωvt(i) + c1r1(pi(i) - xt(i)) + c2r2(pg(i) - xt(i))
    xt+1(i) = xt(i) + vt+1(i)
End
```

其中, Dims 为搜索空间的维度,粒子在进行修订时,各维度分量之间是相互独立的,而当各维度分量之间相互关联时,这种修订策略是无效的。为解决这一问题,提高搜索效率,本文按某一概率将粒子整体修正,对于算法的每一步,修订方式如下:

```
Vector revision strategy:
If Rand > α
For i=1 to Dims
    vt+1(i) = ωvt(i) + c1r1(pi(i) - xt(i)) + c2r2(pg(i) - xt(i))
    xt+1(i) = xt(i) + vt+1(i)
End
Else
    vt+1 = ωvt + c1r1(pi - vt) + c2r2(pg - vt)
    xt+1 = xt + vt+1
End If
```

其中,  $Rand$  为小于 0 和 1 之间的随机数,  $\alpha$  为给定阈值,这样的策略可以加快个体收敛到群体最优的效率,特别是当各维度分量之间相互关联时,搜索效率的提高是显著的。在这里阈值  $\alpha$  的设定非常重要,若  $\alpha$  值过小,群体中粒子按整体修订的数量过大,效率提高的同时也增加了“早熟”的可能;反之,搜索效率就会受到影响。在本文实验中,设定  $\alpha = 0.945$  效果较为理想。

### 2.2 局部最优跳出策略

在 PSO 方法的搜索过程中,算法陷入局部最优而产生“早熟”现象的原因主要包括:①由于群体中粒子的速度衰减过快,粒子被修订的步长迅速减小,使搜索效率过低或搜索停滞;②对于一些

具有较强的欺骗性的复杂问题,搜索一旦陷入局部最优,即使粒子的速度并没有完全衰减,粒子撞到比现有最优点更好的点的概率依然较小,从而造成搜索的停滞。为了使粒子有效地跳出局部最优,避免“早熟”,本文制定了如下策略:

```

Jump out of local optimum strategy:
If Immovable >  $\gamma$  or Sigm <  $\delta$ 
  Save global best into Bestgroup;
  If Rand >  $\theta$  and BestGroup not Null
    global best = Randomly select different
    particle from BestGroup
  Else
    Initial population;
    Initial global best;
  End If;
End If;

```

上段程序中,Immovable 为群体最优保持不变的次数,  $\gamma, \delta, \theta$  为指定的阈值, BestGroup 为一数组,用来保存经过 Immovable 次迭代保持不变或修订效率较低的群体最优,这里用 *sigm* 来确定群体最优的修订效率:

$$sigm = \frac{\sum_{i=1}^M rate_i}{M}$$

初始时为其每个分量  $rate_i (i = 1, \dots, M)$  赋予相对较大的值,搜索过程中按如下方式对 *rate* 进行赋值:

$$rate(2, \dots, M) = rate(1, \dots, M - 1)$$

$$rate_1 = \frac{g\_best_{i+1} - g\_best_i}{S(i + 1) - S(i)}$$

式中:  $g\_best_{i+1}, g\_best_i$  分别为算法相继两次搜索到的群体最优,  $S(i + 1), S(i)$  分别为获得上述两个群体最优的时间。

由上述改进策略可以看出,当群体最优长时间不变或变化值小于一定阈值时,算法按某一概率或从 BestGroup 中选择一个粒子作为全局最优,或重新初始化群体,生成新的群体最优。由于 BestGroup 中的粒子都是曾经的群体最优,当其被再次选择时,群体原有的分布模式打破,群体中粒子的速度会间接地得到补偿,完成搜索的概率依然很高。对于一些复杂的多峰问题,粒子只有跳出局部最优才有希望完成搜索,因此,当群体最优长时间不变或变化值小于一定阈值时,按某一概率重新初始化群体,生成新的群体最优,以跳出局部最优的吸引。本文将引入上述改进策略的

PSO 方法称为改进的 PSO 方法 (Improved particle swarm optimization, IPSO), 其执行方式如下所示:

```

Initialize Population and Parameters
Do
  Evaluate Every Particle in Swarm;
  Select Individual Optimum  $p_i$  and Global
  Optimum  $p_g$ ;
  Run Vector revision strategy;
  Run Jump out of local optimum strategy;
Until termination criterion is met.

```

图 1 为 PSO 改进前、后的执行情况对比。图中两个测试函数分别为 CEC2005 中的  $f_3$  (Shifted rotated high conditioned elliptic function) 和  $f_{12}$  (Schwefel's problem 2.13)。其中,  $f_3$  是在 Elliptic function 基础上经过移动和旋转得到的,只有唯一极值点。  $f_{12}$  是由 Schwefel's problem 经过移动操作得到的,存在多个局部最优点和一个全局最优点。在图 1(a) 中,由于进行了旋转和移动,虽然测试函数只有一个最优点,但是搜索到最优点的难度依然很大。  $f_3$  的旋转操作的实现方式如下:

$$\vec{x} = \vec{x} \times \mathbf{M}$$

式中:  $\mathbf{M}$  为指定矩阵。

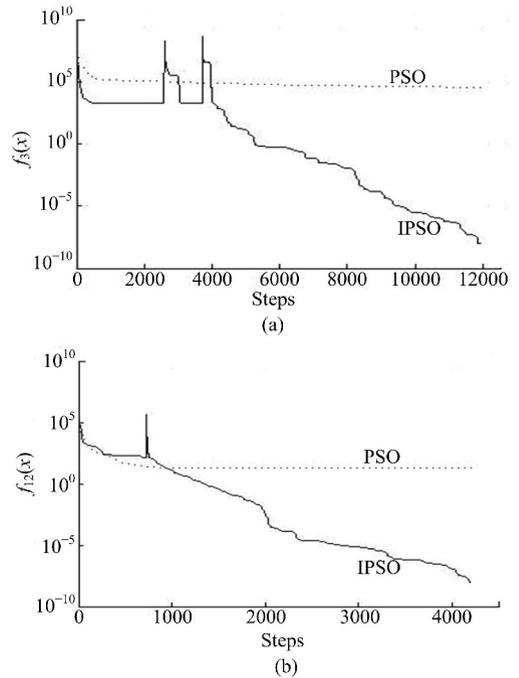


图 1 IPSO 方法跳出局部最优样例  
 Fig. 1 Example of IPSO jumping out of local optimum point

不难看出,经过旋转之后,位置向量各分量之间相互关联。从图 1(a)中可以看出,IPSO 的执行曲线有两次“突变”,这是“局部最优跳出策略”触发的结果(每次运行突变的位置、次数通常不相同),经过两次跳出之后,群体最终收敛到全局最优点上,而 PSO 陷入“早熟”状态,搜索失败。在“局部最优跳出”策略触发之前,在 IPSO 中只有“整体修正策略”发生作用,从图 1(a)中不难看出,经过“整体修正策略”改进后的 IPSO 收敛速度明显好于 PSO。图 1(b)经历了一次跳出,由于  $f_{12}$  并没有经过旋转操作,各分量之间独立,“整体修正策略的”作用似乎并不是很大。

### 3 实验过程和结果

采用 CEC2005 中的 25 个 benchmark 问题<sup>[8]</sup>进行实验,在这 25 个函数中  $f_1 \sim f_6$  为单峰函数,  $f_7 \sim f_{25}$  为多峰函数,这些函数都是在传统 benchmark 函数的基础上经过偏移、旋转等操作得到的,文献[8]对这些函数进行了详细的描述。文献[9]对 CEC2005 会议中的 11 个算法进行了详尽的比较。本文同样采用与文献[9]一致的比较方式:对于不同的测试问题,维度分别取 Dims=10 和 Dims=30,最大迭代次数为  $\text{Dims} \times 10^5$ ,每一个测试问题分别执行 25 次,实验中的 benchmark 测试函数来自于网站 <http://www3.ntu.edu.sg/home/EPNSugan/>提供的 matlab 源程序。文献[9]中用下式衡量算法的执行效果:

$$FEs = \frac{\text{mean}(\# \text{ fevals}) \times \# \text{ all runs}}{\# \text{ successful runs}}$$

式中: # fevals 为每次成功运行的迭代次数; # all runs 为算法执行的总次数(25 次); # successful runs 为 25 次中成功执行的次数; FEs 的值越小,表明算法的搜索性能越高。

表 1 和表 2 列出了搜索空间维度为 10 和 30 的条件下,文献[9]中获得最好结果的方法与 IPSO 运行结果的比对情况。

表 1 和表 2 中,圆括号中的数字是文献[9]中 11 个算法中 25 次运行至少成功一次的方法数,IPSO 栏中的第 3 子栏表示的是 IPSO 方法的 FEs 值与前一栏中 Best FEs 的比,对于 CEC2005 中的 25 个测试问题,文献[9]所提到的 11 个算法对其中的  $f_8, f_{13}, f_{14}, f_{16} \sim f_{25}$  在规定时间内均未完成搜索。IPSO 在搜索空间维度为 10 的情况下,在 25 次执行中两次完成了函数

$f_{13}$  搜索,由于无法比较,表 1 中列出了对应的 FEs 值。当搜索空间维度 Dims=10 时,在可比较的 13 个函数中用 IPSO 方法取得了 8 个函数最好的搜索结果。从表 2 可以看出,当 Dims=30 时,有 7 个函数取得了最好的搜索效果,而未经改进的 PSO 除  $f_1, f_2, f_4$  外均未完成搜索。这也充分证明了本文改进策略的有效性。

表 1 Dims=10 时 IPSO 与文献[9]中最好算法实验结果对比

Table 1 Results of IPSO method compared with literature [9] in the conditions that Dims=10

Fun	Method name	Best in [9]			IPSO	
		Succ.	Best	Succ.	FEs	$\frac{FEs}{B\_FEs}$
		rate/%	FEs/%	rate/%		
1	K-PCX	100(11)	1000	100	450	0.45
2	K-PCX	100(11)	2400	100	800	0.33
3	G-CMA-ES	100(7)	6500	100	16900	2.6
4	G-CMA-ES	100(10)	2900	100	1190	0.41
5	G-CMA-ES	100(7)	5900	100	2360	0.40
6	K-PCX	100(8)	7100	100	2400	0.34
7	G-CMA-ES	100(9)	4700	100	5730	1.22
9	L-SaDE	100(7)	17000	100	12900	0.76
10	K-PCX	92(2)	55000	80	77000	1.40
11	DE	48(3)	190000	28	720100	3.79
12	K-PCX	56(3)	8200	100	7790	0.95
13	—	—	—	8		$[1.07 \times 10^6]$
15	L-SaDE	92(3)	33000	16	269280	8.16

表 2 Dims=30 时 IPSO 与文献[9]中实验结果对比

Table 2 Results of IPSO method compared with literature [9] in the conditions that Dims=30

Fun	Method name	Best in [9]			IPSO	
		Succ.	Best	Succ.	FEs	$\frac{FEs}{B\_FEs}$
		rate/%	FEs	rate/%		
1	K-PCX	100(7)	2700	100	810	0.30
2	K-PCX	100(9)	12000	100	3600	0.30
3	G-CMA-ES	100(4)	43000	100	253700	5.9
4	G-CMA-ES	40(4)	59000	100	25960	0.44
5	G-CMA-ES	100(2)	66000	100	23760	0.36
6	G-CMA-ES	100(6)	60000	100	15000	0.25
7	G-CMA-ES	100(10)	6100	100	10675	1.75
9	L-SaDE	100(3)	99000	100	84150	0.85
10	K-PCX	56(2)	450000	44	990000	2.20
11	G-CMA-ES	4(1)	500000	—	—	—
12	K-PCX	20(3)	180000	100	126000	0.70

### 4 结束语

针对类似 CEC2005 中的 benchmark 问题,

提出了 PSO 方法的“整体修正”和“局部最优跳出”两种改进策略,并采用 CEC2005 提供的 25 个 benchmark 函数和相同的运行方式对改进后的 PSO 进行实验,实验结果与 CEC2005 会议上的 11 个方法进行了比较。结果表明,改进后的 PSO 方法的搜索效率和跳出局部最优的能力都有了显著提高,改进策略是有效的。

#### 参考文献:

- [ 1 ] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[C]//Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995.
- [ 2 ] Kennedy J, Eberhart R C. Particle swarm optimization[C]// Proc IEEE International Conference on Neural Networks, IEEE Service Center, Piscataway, 1995.
- [ 3 ] Shi Y, Eberhart R. Monitoring of particle swarm optimization[J]. Frontiers of Computer Science in China, 2009, 3(1):31-37.
- [ 4 ] 崔光照, 李小广, 张勋才, 等. 基于改进的粒子群遗传算法的 DNA 编码序列优化[J]. 计算机学报, 2010, 33(2):311-316.
- Cui Guang-zhao, Li Xiao-guang, Zhang Xun-cai, et al. The optimization of DNA encodings based on modified PSO/GA algorithm[J]. Chinese Journal of Computers, 2010, 33(2):311-316.
- [ 5 ] 纪震, 周家锐, 廖惠连, 等. 智能单粒子优化算法[J]. 计算机学报, 2010, 33(3):556-562.
- Ji Zhen, Zhou Jia-rui, Liao Hui-lian, et al. A novel intelligent single particle optimizer [J]. Chinese Journal of Computers, 2010, 33(3):556-562.
- [ 6 ] 潘冠宇, 刘大有, 窦全胜, 等. PSO 方法的收敛性及基于微分演化的参数确定策略[J]. 吉林大学学报: 工学版, 2007, 37(4): 842-845.
- Pan Guan-yu, Liu Da-you, Dou Quan-sheng, et al. Convergence property and parameters-selection strategy base on differential evolution of PSO [J]. Journal of Jilin University(Engineering and Technology Edition), 2007, 37(4): 842-845.
- [ 7 ] Poli R. Mean and variance of the sampling distribution of particle swarm optimizers during stagnation [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(4): 712-721.
- [ 8 ] Suganthan P N, Hansen N, Liang J J, et al. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization[DB/OL]. [2006-12-20]. <http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/Shared%20Documents/CEC2005/Tech-Report-May-30-05.pdf>
- [ 9 ] Hansen N. Compilation of results on the 2005 CEC benchmark function set [DB/OL]. [2006-12-20]. [http://www3.ntu.edu.sg/home/epnsugan/index\\_files/CEC-05/compareresults.pdf](http://www3.ntu.edu.sg/home/epnsugan/index_files/CEC-05/compareresults.pdf)