

基于学习理论的改进粒子群优化算法

徐 霜<sup>1</sup>, 万 强<sup>2</sup>, 余 琍<sup>2</sup>

(1. 武汉大学 高新技术产业发展部, 湖北 武汉 430072; 2. 武汉大学 计算机学院, 湖北 武汉 430072)

摘 要: 论文针对粒子群算法容易陷入局部最优的问题, 提出基于学习理论的粒子群算法(L-PSO). 该算法通过为粒子群全局最优粒子设定最大周期限制, 使达到最大周期的全局最优粒子可以被取代, 同时利用聚类的思想对粒子群进行分组, 通过随机选择两个组中心, 以一定概率进行交叉变异, 生成竞争粒子并替换达到最大周期的全局最优粒子, 能够较好地避免算法陷入局部最优, 提高算法的收敛速度. 在基准测试函数集上的测试结果表明该算法有效.

关键词: 粒子群算法; 最优化; 有效性; 测试函数

中图分类号: TP311 文献标志码: A doi:10.13705/j.issn.1671-6833.2019.02.007

0 引言

最优化问题分为传统优化方法和启发式优化方法. 传统优化方法大多利用目标函数的梯度(或导数)信息实现单可行解的惯序、确定性搜索; 启发式优化方法以仿生算法为主, 通过启发式搜索策略实现多可行解的并行、随机优化. 启发式搜索算法不要求目标函数连续、可微等信息, 具备较好的全局寻优能力, 因而成为最优化领域的一个研究热点<sup>[1-2]</sup>.

多启发式算法众多, 有基于蚂蚁觅食提出的蚁群算法(ant colony optimization, ACO)<sup>[3]</sup>, 基于蜜蜂采蜜提出的人工蜂群算法(artificial bee colony, ABC)<sup>[4]</sup>, 基于细菌捕食提出的细菌捕食算法(bacterial foraging optimization, BFO)<sup>[5]</sup>, 差分进化算法(different evolution, DE)<sup>[6]</sup>, 头脑风暴算法(brain storm optimization, BSO)<sup>[7-8]</sup>等. 作为一种经典的启发式算法——粒子群优化算法(particle swarm optimization algorithm, PSO)是对鸟群捕食行为的模拟<sup>[9-10]</sup>. PSO 概念简单, 易于实现, 目前已广泛运用于函数优化、神经网络训练、模糊系统控制以及其他的科学和工程应用领域<sup>[11-15]</sup>.

原始的粒子群算法易陷入局部最优. 为解决原始粒子群算法早熟的缺点, 笔者根据人类社会中领袖任期机制提出一种周期性最优机制, 同时

引入周期性的学习策略, 提出聚类分组变异方法来改进粒子群从而提高种群的多样性, 避免陷入局部最优, 同时保障种群的收敛速度.

1 标准粒子群算法

在 PSO 中, 每个解被称为“粒子”(particle), 所有的粒子都有一个适应度值(fitness value), 每个粒子还有一个速度向量决定他们的运动方向, 粒子通过不断更新运动方向搜索解空间. 算法的数学描述为: 在  $Q$  维空间中有  $n$  个粒子在进行搜索, 粒子的位置表示为  $X_i = (x_{i1}, x_{i2}, \dots, x_{iQ})$ , 粒子对应的速度可以表示为  $V_i = (v_{i1}, v_{i2}, \dots, v_{iQ})$ , 粒子在搜索时要考虑两个因素: ①单个粒子搜索到的个体历史最优值  $P_{bi}$ ,  $P_{bi} = (p_{bi1}, p_{bi2}, \dots, p_{biQ})$ ,  $i = 1, 2, 3, \dots, n$ ; ②全部粒子搜索到的全局历史最优值  $P_{gb}$ ,  $P_{gb} = (p_{gb1}, p_{gb2}, \dots, p_{gbQ})$ ,  $i = 1, 2, 3, \dots, n$ .

各粒子的速度和位置按如下公式进行更新:

$$V_{id}(t+1) = \omega V_{id}(t) + c_1 r_1 (P_{bid}(t) - X_{id}(t)) + c_2 r_2 (P_{gbd}(t) - X_{id}(t)); \tag{1}$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1), \tag{2}$$

式中:  $d$  为粒子的维度;  $\omega$  是保持原来速度的系数, 即惯性权重;  $c_1$ 、 $c_2$  是粒子跟踪自己和群体历史最优值的权重系数;  $r_1$ 、 $r_2$  是  $[0, 1]$  均匀分布的随机数.

收稿日期:2018-04-11; 修订日期:2018-06-21  
基金项目:国家自然科学基金资助项目(161773296)  
作者简介:徐 霜(1975—), 男, 武汉大学高级工程师, 博士, 主要从事智能计算、数据挖掘等研究, E-mail: xushuang@whu.edu.cn.  
通信作者:余 琍(1977—), 女, 博士, 主要研究方向为智能计算、数据管理及安全, E-mail: yul@whu.edu.cn.

## 2 基于学习理论的改进粒子群优化算法

为避免早熟,提高种群的多样性,笔者引入了周期性学习策略,分别提出周期性最优机制、聚类分组变异机制以及分类预测评估机制.周期性最优机制通过为全局最优粒子设定数据,全局最优粒子陷入局部更新,避免整个种群陷入局部最优.聚类分组变异机制通过聚类划分种群,再选取不同类的粒子产生竞争粒子.分类预测评估机制通过依照以往若干代竞争粒子与全局最优粒子的进化结果构建分类器,预测竞争粒子是否能够替换全局最优粒子,确保全局最优粒子的优越性.

### 2.1 周期性最优机制

在自然界中,每个群居生物种群都有对应的领袖带领种群前行,无论是动物还是人类都是如此.Kubota 等<sup>[16-17]</sup>将这个概念运用到遗传算法设计.在包含老年化机制的遗传算法中,每个个体都被分配一个年龄.当它的年龄达到一个预定义的寿命时,该个体会被移除.Chen 等<sup>[18]</sup>同样引入衰老机制到粒子群算法中,以此提高粒子群的多样性.

笔者借鉴这种思想,针对粒子群算法中全局最优粒子提出一种限制条件——周期性,即周期性地改变粒子群中的全局最优粒子,从而避免种群在全局最优粒子陷入局部最优而导致整个群体也陷入局部最优.

### 2.2 聚类分组变异机制

为了周期性地生成新的全局最优粒子,笔者首先生成一个竞争粒子来尝试替换这个全局最优粒子.

聚类可以用于寻找数据内在的分布结构,因此研究者提出将聚类分析应用到多目标演化计算中<sup>[19-21]</sup>.笔者提出一种聚类分组变异机制,借助聚类的思想对粒子群依照其个体分布情况进行分组<sup>[22]</sup>.当全局最优粒子存在的时间超过周期限制时,对粒子群采用 K-means.依据粒子间的欧氏距离进行分组,最后随机选择 2 个组依照公式(3)进行变异产生竞争粒子,

$$competitor_i = \begin{cases} P_{gbt} + R * (X_{k1,i} - X_{k2,i}) & \text{if rand} < r \\ P_{gbt}, & \text{otherwise} \end{cases} \quad (3)$$

笔者提出的改进粒子群算法流程如图 1 所示.

## 3 参数设置

笔者提出的 L-PSO 算法引入了 3 个主要参数.包括聚类的分组个数 *SET*、产生竞争粒子时的

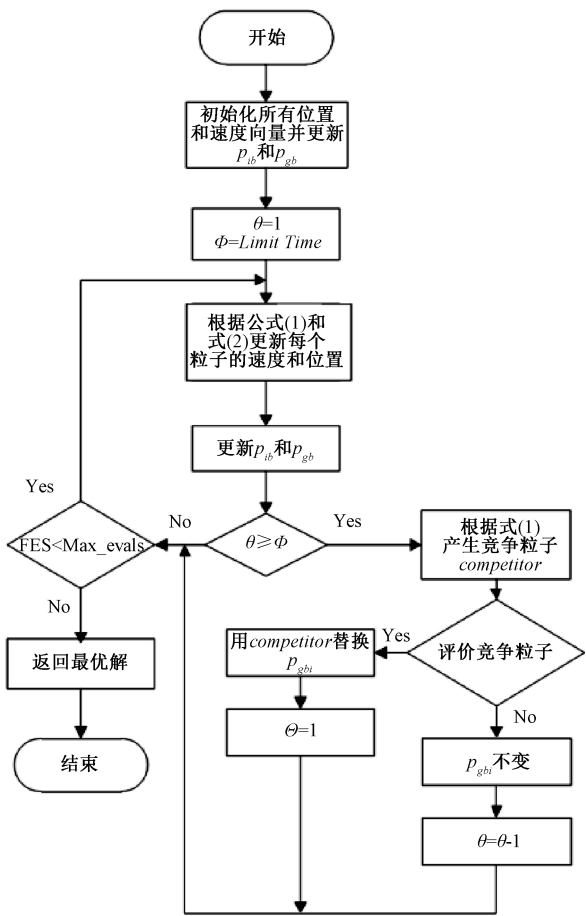


图 1 L-PSO 算法流程框图

Fig.1 Flowchart of L-PSO

变异概率 *r* 以及 *P<sub>gb</sub>* 的周期限制 *tenure*.

聚类分组中较大的 *SET* 将加重算法的时间复杂度;反之,聚类分组后无法从个体在决策空间的分布上看出个体间的差异,因此,为平衡时间复杂度和方法的有效性,我们在文中将 *SET* 设置为 3.

产生竞争粒子时的变异概率为 *r*. 因为当前的 *P<sub>gb</sub>* 粒子仍然是已知的最优粒子,因此 *P<sub>gb</sub>* 的解在不同维度上的值是具有参考价值的. 因此, *r* 值设置为  $1/n$ , *n* 是解的维度大小. 这样,生成的竞争粒子在极大程度上保持 *P<sub>gb</sub>* 解结构的同时,在每一个维度上都有相同的产生变异的机率. 这样,即保证了生成的竞争粒子质量,也有很大机会生成有较好进化潜能的新粒子.

此外,为了衡量 *tenure* 值的变化对算法性能的影响,我们从所选择的 22 个测试函数如表 1 所示. 选择了单峰函数 f1,从 CEC2013 中选择一个多峰函数(在本文编号为 f22)做参数测试,统计结果如图 2. 可以看出,当 *tenure* 设置为 10 时, L-PSO 在解决单峰和多峰函数时能显示出较好的性能,因此,在实验中我们将 *tenure* 的值取为 10.

表 1 标准测试函数

Tab.1 Basic test functions

编号	测试函数	$N$	搜索范围	最优值
f1	Sphere	30	$[-100, 100]$	0
f2	Schwefel's P2.22	30	$[-10, 10]$	0
f3	Quadric	30	$[-100, 100]$	0
f4	Rosenbrock's	30	$[-10, 10]$	0
f5	Step	30	$[-100, 100]$	0
f6	Schwefels	30	$[-500, 500]$	-12 569.5
f7	Rastrigin	30	$[-5.12, 5.12]$	0
f8	Noncontinuous Rastrigin	30	$[-5.12, 5.12]$	0
f9	Ackley	30	$[-32, 32]$	0
f10	Griewank	30	$[-600, 600]$	0
f11	Penalized 1	30	$[-50, 50]$	0
f12	Penalized 2	30	$[-50, 150]$	0

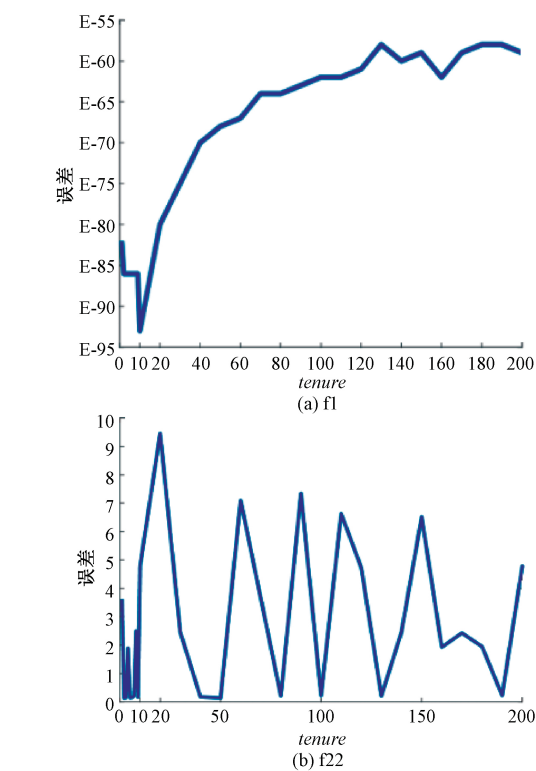


图 2 f1 和 f22 上不同 tenure 值 L-PSO 的算法性能

Fig.2 Performance of the L-PSO under different tenure values on f1 and f22

## 4 对比实验

### 4.1 测试函数和对比算法

为了验证本文所提策略的有效性,笔者选择了 22 个测试函数来评估 L-PSO 的性能.前 12 个为表 1 所示的基本函数,其余函数为 CEC2013 中的前十个测试函数,在本文中编号为 f13 ~ f22.

为了比较算法性能,笔者选择了 4 个有代表性的改进的 PSO 算法作为对比算法,包括 ALC-PSO、DMS-PSO、GPSO 和线性权重递减版本的 GPSO.所有算法的参数设置与原文献中相同,具体设置如表 2 所示.此外,L-PSO 中公式(3)的  $R$  值为生成竞争粒子的控制参数,设置为 0.5.为保

证实验结果准确,每一个算法都在各个测试函数上独立地运行 20 次并求取平均值.每个测试函数的维度  $N$  设置为 30,算法种群大小设置为 20,算法迭代的停止条件为最大函数评估次数  $\text{Max\_evals} = 200\ 000$ .

表 2 对比算法及参数设置	
Tab.2 Parameters settings for algorithms	
算法名称	参数设置
ALC-PSO	$c_1 = 2, c_2 = 2, \omega = 0.4, T = 2, \omega = 60$
DMS-PSO	$\chi = 0.729\ 8, c_1 = c_2 = 1.494\ 45, M = 4, R = 10$
GPSO	$c_1 = 2, c_2 = 2, \omega = 0.4$
线性权重递减的 GPSO	$c_1 = 2, c_2 = 2, \omega \in [0.4, 0.9]$
L-PSO	$c_1 = 2, c_2 = 2, \omega = 0.4, SET = 3, tenure = 10, r = 1/n$

### 4.2 数值结果对比

各个算法 20 次独立运行后的平均误差值  $MEAM$  以及标准差  $SD$  的统计结果如表 3 所示.最好的结果用粗体表示,次优的结果用下划线表示.

从表 3 中可以看出 L-PSO 算法在 22 个测试函数中的 f5、f7、f9、f10、f11、f12、f13、f15、f19、f20 和 f22 上取得了最好的结果.在 f6、f8、f14、f16、f18 和 f21 上取得了次优的结果.此外,L-PSO 在解决多峰函数时,展现出了优越的性能,在 f6 ~ f12、f18 ~ f22 这 12 个多峰函数的试验中,L-PSO 在其中的 8 个函数上取得了最优解,综合所有函数最后的结果可以看出,基于学习理论的改进粒子群算法在性能上具有一定的优越性.

### 4.3 算法收敛比较

图 3 为 L-PSO 算法与其他算法的比较结果.其中横坐标为函数评估次数,纵坐标为误差值.由于篇幅限制,只展示部分有代表性的结果.从收敛曲线可以看出,L-PSO 保持了较好的收敛性可

表 3 对比实验结果

Tab.3 Results comparison on all test functions

		L-PSO	GPSO	GPSO 0.9-0.4	ALC-PSO	DMS-PSO
f1	MEAN	6.49E-94	<b>3.42E-165</b>	4.65E-53	<u>4.32E-165</u>	4.95E-24
	SD	2.02E-93	<b>0.00E+00</b>	7.97E-53	<u>0.00E+00</u>	1.37E-23
f2	MEAN	3.89E-69	<b>3.70E-93</b>	3.06E-34	<u>1.53E-87</u>	1.53E-12
	SD	7.88E-69	<b>7.11E-93</b>	5.91E-34	<u>4.83E-87</u>	3.36E-12
f3	MEAN	4.61E-05	<b>8.40E-11</b>	8.33E-02	<u>1.18E-10</u>	1.68E+01
	SD	7.49E-05	<b>1.11E-10</b>	6.34E-02	<u>1.31E-10</u>	1.40E+01
f4	MEAN	2.10E+01	<u>1.22E+01</u>	2.51E+01	<b>4.96E+00</b>	2.34E+01
	SD	<u>4.81E-01</u>	4.48E+00	2.88E+01	6.70E+00	<b>2.25E-01</b>
f5	MEAN	<b>0.00E+00</b>	3.63E+01	0.00E+00	<u>0.00E+00</u>	0.00E+00
	SD	<b>0.00E+00</b>	1.14E+02	0.00E+00	<u>0.00E+00</u>	0.00E+00
f6	MEAN	<u>2.37E+03</u>	4.42E+03	2.67E+03	<b>1.99E+03</b>	4.24E+03
	SD	7.69E+02	1.07E+03	7.42E+02	<b>2.78E+02</b>	<u>5.63E+02</u>
f7	MEAN	<b>0.00E+00</b>	5.37E+01	3.54E+01	<u>6.00E-14</u>	2.14E+01
	SD	<b>0.00E+00</b>	2.06E+01	1.37E+01	<u>3.55E-14</u>	8.34E+00
f8	MEAN	<u>9.20E-11</u>	3.00E+00	3.00E-01	<b>0.00E+00</b>	1.81E+00
	SD	<u>2.91E-10</u>	1.70E+00	4.83E-01	<b>0.00E+00</b>	2.80E+00
f9	MEAN	<b>8.88E-17</b>	1.20E+00	<u>9.50E-15</u>	1.09E-14	3.79E-14
	SD	<b>1.12E-15</b>	9.18E-01	<u>3.67E-15</u>	4.97E-15	3.85E-14
f10	MEAN	<b>0.00E+00</b>	1.63E-02	2.29E-02	3.10E-02	<u>2.22E-03</u>
	SD	<b>0.00E+00</b>	2.04E-02	1.98E-02	5.12E-02	<u>4.99E-03</u>
f11	MEAN	<b>1.82E-32</b>	8.29E-02	3.11E-02	<u>2.04E-32</u>	5.46E-28
	SD	<b>2.68E-33</b>	1.94E-01	5.01E-02	<u>1.47E-32</u>	1.58E-27
f12	MEAN	<b>6.30E-32</b>	3.75E-01	3.30E-03	<u>1.03E-31</u>	2.20E-03
	SD	<b>7.63E-32</b>	7.05E-01	5.31E-03	<u>2.82E-31</u>	4.63E-03
f13	MEAN	<b>2.27E-13</b>	1.59E+03	3.33E+03	1.32E-12	<u>3.41E-13</u>
	SD	<b>0.00E+00</b>	1.86E+03	1.20E+03	1.86E-12	<u>1.20E-13</u>
f14	MEAN	<u>4.22E+06</u>	6.01E+06	2.25E+07	<b>3.92E+06</b>	6.06E+06
	SD	<b>2.08E+06</b>	8.95E+06	2.13E+07	4.18E+06	3.89E+06
f15	MEAN	<b>2.67E+07</b>	1.71E+10	7.30E+10	1.06E+10	<u>4.69E+08</u>
	SD	<b>2.39E+07</b>	1.46E+10	8.61E+10	1.31E+10	1.26E+09
f16	MEAN	<u>1.43E+03</u>	2.92E+03	9.95E+03	3.09E+03	<b>8.58E+02</b>
	SD	<u>3.74E+02</u>	1.88E+03	9.56E+03	5.73E+02	<b>1.56E+02</b>
f17	MEAN	8.21E+00	1.04E+03	2.34E+03	<b>1.93E-13</b>	<u>5.46E-13</u>
	SD	1.80E+01	9.35E+02	1.09E+03	<b>5.49E-14</b>	<u>3.16E-13</u>
f18	MEAN	<u>5.11E+01</u>	1.35E+02	2.63E+02	<b>4.57E+01</b>	5.33E+01
	SD	<u>2.18E+01</u>	1.03E+02	1.62E+02	2.30E+01	<b>1.43E+01</b>
f19	MEAN	<b>2.83E+01</b>	1.39E+02	1.76E+02	1.29E+02	<u>4.10E+01</u>
	SD	<b>1.47E+01</b>	4.97E+01	6.81E+01	8.60E+01	<u>2.26E+01</u>
f20	MEAN	<b>2.09E+01</b>	2.09E+01	2.10E+01	<u>2.09E+01</u>	2.11E+01
	SD	6.50E-02	5.66E-02	<b>4.46E-02</b>	6.11E-02	<u>5.30E-02</u>
f21	MEAN	<u>1.84E+01</u>	2.74E+01	2.13E+01	2.49E+01	<b>1.74E+01</b>
	SD	3.69E+00	5.01E+00	3.74E+00	<u>3.52E+00</u>	<b>3.18E+00</b>
f22	MEAN	<b>1.89E+00</b>	2.31E+02	8.42E+02	1.19E+02	<u>2.05E+01</u>
	SD	<b>5.56E+00</b>	2.10E+02	3.85E+02	1.06E+02	<u>4.30E+01</u>

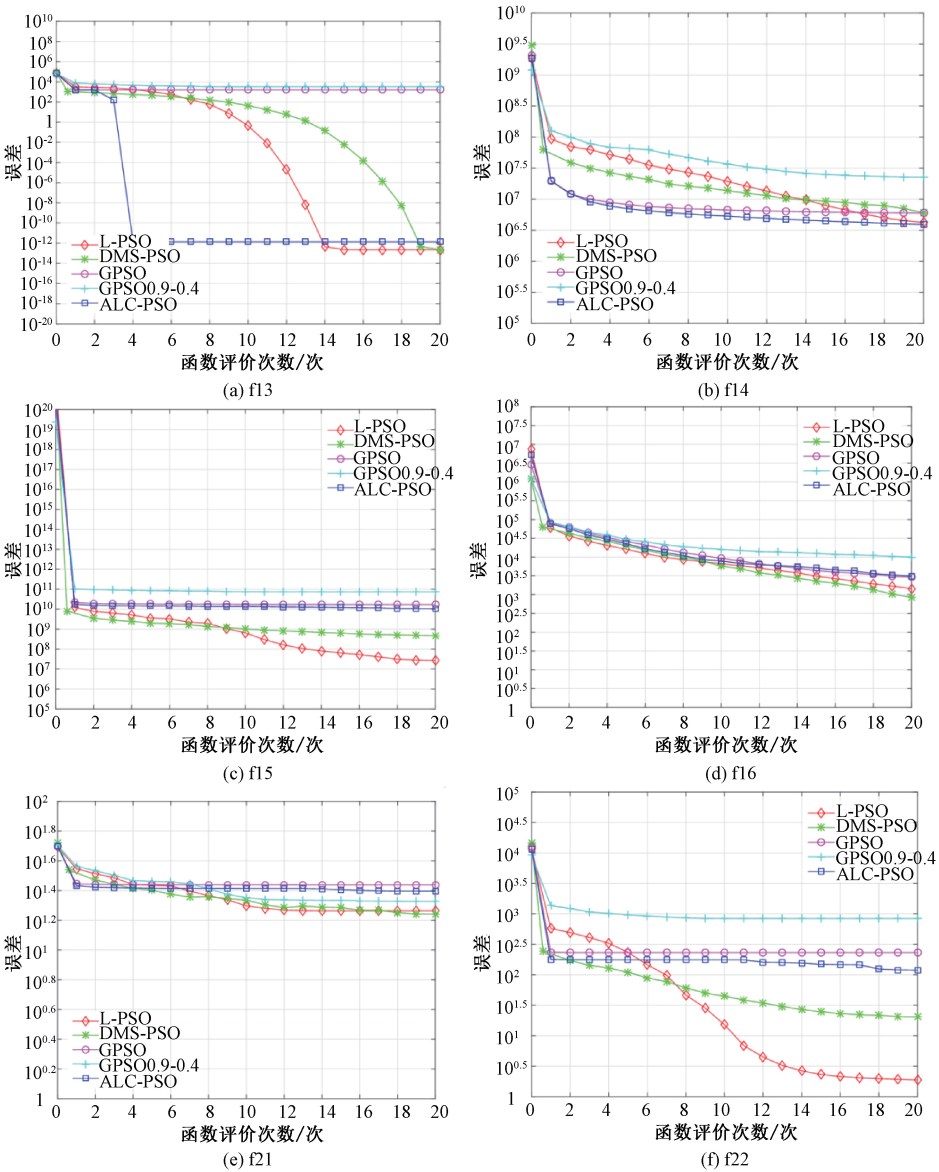


图3 收敛过程对比图

Fig.3 Comparrson of convergencemap

见 L-PSO 在解决本文中的测试函数时显现出了较好的性能。

5 结论

引入周期性学习策略,提出了周期性最优机制以及聚类分组变异机制对标准粒子群进行改进,提出基于学习理论的改进粒子群算法.该算法有效克服了标准粒子群优化及一些经典演化算法的缺陷,能够快速寻优,避免陷入局部最优,在单峰和多峰函数上都能较好地求解.下一步将尝试应用该算法到大规模的实际应用问题上进行优化求解.

参考文献:

[1] XUE B, ZHANG M, BROWNE W N, et al. A survey on evolutionary computation approaches to feature

selection[J]. IEEE transactions on evolutionary computation, 2016, 20(4):606-626.

[2] 梁静,刘睿,瞿博阳,等. 进化算法在大规模优化问题中的应用综述[J]. 郑州大学学报(工学版), 2018, 39(3): 15-21.

[3] DORIGO M, MANIEZZO V, COLORNI A. Ant system: optimization by a colony of cooperating agents [J]. IEEE transactions on systems, man, and cybernetics, part B: cybernetics, 1996, 26(1): 29-41.

[4] KARABOGA D. An idea based on honey bee swarm for numerical optimization[R]. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.

[5] PASSINO K M. Biomimicry of bacterial foraging for distributed optimization and control [J]. Control systems, IEEE, 2002, 22(3): 52-67.

[6] STORN R, PRICE K. Differential evolution - a sim-

- ple and efficient heuristic for global optimization over continuous spaces[J]. *Journal of global optimization*, 1997, 11(4): 341–359.
- [7] SHI Y. Brain storm optimization algorithm [M] // *Advances in Swarm Intelligence*. Springer Berlin Heidelberg, 2011: 303–309.
- [8] 程适, 陈俊风, 孙奕菲, 等. 数据驱动的发展式头脑风暴优化算法综述[J]. *郑州大学学报(工学版)*, 2018, 39(3): 22–28.
- [9] KENNEDY J. Particle swarm optimization [M] // *Encyclopedia of Machine Learning*. Springer US, 2010: 760–766.
- [10] SHI Y, EBERHART R. A modified particle swarm optimizer [C] // *Evolutionary Computation Proceedings*, 1998. *IEEE World Congress on Computational Intelligence*, The 1998 *IEEE International Conference on*. IEEE, 1998: 69–73.
- [11] 严晶晶, 阎新芳, 冯岩. WSN 中基于梯度和粒子群优化算法的分集簇算法[J]. *郑州大学学报(工学版)*, 2016, 37(2): 33–36.
- [12] SHANG R, DAI K, JIAO L, et al. Improved memetic algorithm based on route distance grouping for multiobjective large scale capacitated arc routing problems [J]. *IEEE transactions on cybernetics*, 2017, 46(4): 1000–1013.
- [13] LIANG J J, QIN A K, SUGGANTHAN P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. *IEEE transactions on evolutionary computation*, 2006, 10(3): 281–295.
- [14] 赵新超, 刘国莅, 刘虎球, 等. 基于非均匀变异和多阶段扰动的粒子群优化算法[J]. *计算机学报*, 2014(9): 15.
- [15] YE W, FENG W, FAN S. A novel multi-swarm particle swarm optimization with dynamic learning strategy[J]. *Applied soft computing*, 2017(61): 832–843.
- [16] KUBOTA N, FUKUDA T. Genetic algorithms with age structure [J]. *Soft computing*, 1997, 1(4): 155–161.
- [17] GHOSH A, TSUTSUI S, TANAKA H. Individual aging in genetic algorithms [C] // *Intelligent Information Systems*, 1996. *Australian and New Zealand Conference on*. IEEE, 1996: 276–279.
- [18] CHEN W N, ZHANG J, LIN Y, et al. Particle swarm optimization with an aging leader and challengers[J]. *IEEE transactions on evolutionary computation*, 2013, 17(2): 241–258.
- [19] KOTINIS M. Improving a multi-objective differential evolution optimizer using fuzzy adaptation and K-medoids clustering [J]. *Soft computing*, 2014, 18(4): 757–771.
- [20] ZHANG H, ZHOU A, SONG S, et al. A self-organizing multiobjective evolutionary algorithm [J]. *IEEE transactions on evolutionary computation*, 2016, 20(5): 792–806.
- [21] AES F G G, WANNER E F, TAKAHASHI R H C. A quality metric for multi-objective optimization based on hierarchical clustering techniques [C] // *Evolutionary Computation*, 2009. *CEC'09. IEEE Congress on*. IEEE, 2009: 3292–3299.
- [22] LU X F, TANG K. Classification-and regression-assisted differential evolution for computationally expensive problems [J]. *Journal of computer science and technology*, 2012, 27(5): 1024–1034.

## An Improved Particle Swarm Optimization Algorithm Based on Learning Theory

XU Shuang<sup>1</sup>, WAN Qiang<sup>2</sup>, YU Li<sup>2</sup>

(1. The Department of Hi-tech Industry, Wuhan University, Wuhan 430072, China; 2. School of Computer Science, Wuhan University, Wuhan 430072, China)

**Abstract:** Since PSO algorithm was easy to get trapped into local optimum, in this paper, based on the learning theory a new PSO algorithm named as L-PSO was proposed. In L-PSO, an integer value was set as the maximum cycle limitation for the global best particles, and propose a clustering grouping mutation mechanism which could divide the particles into some sub-swarms and generates the competitive particle by crossover and mutation of the two centers selected randomly from sub-swarms. Then the competitive particle was used to replace the global optimum particle which could help jump out of the local optimum and improve the convergence speed. Experimental results on several benchmark test functions showed that L-PSO was very effective.

**Key words:** PSO; optimization; efficiency; test functions