

## 带反向预测及斥力因子的改进粒子群优化算法

范成礼, 邢清华, 李响, 王振江

(空军工程大学 防空反导学院, 西安 710051)

**摘要:** 针对传统粒子群算法在求解高维空间中复杂多峰函数时容易陷入局部最优的问题, 提出带反向预测和斥力因子的改进粒子群优化算法. 算法通过引入反向预测因子改进速度更新方式, 以降低粒子在运动过程中产生惰性而出现早熟收敛的概率, 并给出带斥力因子的位置修正策略, 使粒子均匀分散于搜索空间, 从而避免陷入局部最优. 实验分析表明, 在对高维空间中复杂多峰函数进行优化求解时, 改进的粒子群优化算法较传统粒子群算法更加优越.

**关键词:** 粒子群优化; 局部最优; 反向预测因子; 斥力因子

**中图分类号:** TP182; TP391

**文献标志码:** A

## Improved particle swarm optimization algorithm with reverse forecast and repulsion

FAN Cheng-li, XING Qing-hua, LI-Xiang, WANG Zhen-jiang

(School of Air and Missile Defense, Air Force Engineering University, Xi'an 710051, China. Correspondent: FAN Cheng-li, E-mail: aw7983@163.com)

**Abstract:** For the complex multi-peaks function with high dimension, the improved particle swarm optimization algorithm with reverse forecast and repulsion(RFRPSO) is proposed on the basis of analyzing the problem of premature. Firstly, this method improves the speed renewal way by introducing the reverse forecast factor in order to decrease the probability of premature convergence. Furthermore, the repulsion factor is introduced to make the swarm even distribution in search space, which can avoid local optimum. Finally, experimental results show that the performance of RFRPSO algorithm is superior to the traditional PSO algorithm.

**Keywords:** particle swarm optimization; local optimum; reverse forecast factor; repulsion factor

### 0 引言

粒子群优化(PSO)算法是一种基于社会群体行为的智能优化算法, 由 Kenney 等<sup>[1-2]</sup>于 1995 年首次提出. 由于 PSO 算法收敛速度快, 结构简单且易于实现, 在对优化问题的求解中取得了巨大的成功, 成为国内外研究的热点<sup>[3-10]</sup>. PSO 算法在解决高维空间中复杂多峰函数时, 与其他进化算法类似, 存在容易发生早熟收敛而陷入局部最优的问题. 针对这一问题, 人们近几年提出了许多改进算法<sup>[3-17]</sup>, 但算法大多着眼于 PSO 的参数选择或某个参数的动态修改策略<sup>[5-7]</sup>, 或其他混合智能算法<sup>[8-12]</sup>, 难以克服 PSO 算法易陷入局部极小的固有弱点. 文献[13]提出一种基于动态边界的粒子群优化算法, 在低维函数优化中效果良好, 但在高维函数优化中效果不佳. 文献[14]结合粒子群

搜索的随机行为分析, 提出了均匀搜索粒子群算法, 但并未从根本上改变粒子的位置和速度更新方式.

针对地形复杂且含有较多局部最优的函数优化问题, 本文根据粒子运动的特点, 通过增加反向测试因子和斥力因子, 改进粒子的速度和位置更新策略, 提出带反向预测及斥力因子的改进粒子群优化算法(RFRPSO), 有效地避免了传统 PSO 算法在求解过程中容易发生早熟收敛而陷入局部最优的问题. 最后通过典型测试函数的仿真实验对比分析表明, RFRPSO 算法的性能优于传统粒子群算法.

### 1 基本 PSO 算法

基本 PSO 算法的粒子的速度和位置更新公式为

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{id} - x_{id}^k) + c_2 r_2 (p_{gd} - x_{id}^k),$$

收稿日期: 2013-12-01; 修回日期: 2014-03-11.

基金项目: 国家自然科学基金项目(61272011, 61309022, 61309008); 陕西省基金项目(2013JM1003).

作者简介: 范成礼(1988-), 女, 博士生, 从事智能信息处理、信息融合的研究; 邢清华(1966-), 女, 教授, 博士生导师, 从事防空作战仿真建模、防空作战智能决策等研究.

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}. \quad (1)$$

其中:  $c_1$ 、 $c_2$  为加速因子;  $r_1$ 、 $r_2$  为  $[0,1]$  区间的随机数;  $X_i^k = (x_{i1}^k, x_{i2}^k, \dots, x_{iD}^k)$  ( $i = 1, 2, \dots, N, d = 1, 2, \dots, D$ ) 为第  $k$  代中第  $i$  个粒子的  $D$  维位置矢量, 粒子每一维上的位置取值范围为  $x_{id} \in [x_{\min}, x_{\max}]$ ;  $V_i^k = (v_{i1}^k, v_{i2}^k, \dots, v_{iD}^k)$  为第  $k$  代中第  $i$  个粒子的飞行速度, 粒子每一维上的速度取值范围为  $v_{id} \in [v_{\min}, v_{\max}]$ ;  $P_i^k = (p_{i1}^k, p_{i2}^k, \dots, p_{iD}^k)$  为截止  $k$  代, 第  $i$  个粒子经历过的最优位置;  $P_g^k = (p_{g1}^k, p_{g2}^k, \dots, p_{gD}^k)$  为截止  $k$  代, 整个粒子群搜索到的最优位置。

对于惯性权重系数  $\omega$ , 采用非线性递减权重策略, 具体形式为

$$\omega = \omega_{\text{ini}} - (\omega_{\text{ini}} - \omega_{\text{end}})(k/k_{\text{max}})^2. \quad (2)$$

其中:  $\omega_{\text{ini}}$  为初始权重;  $\omega_{\text{end}}$  为最终权重;  $k_{\text{max}}$  为最大迭代次数,  $k$  为当前迭代次数。

由基本 PSO 算法的寻优机理可知, 当粒子过于相信自身历史最优值和群体最优值时, 容易过快陷入局部极值点, 从而发生“早熟收敛”现象。而许多高维空间中的复杂多峰函数在全局最优值附近往往分布着一些局部最优点, 这些局部最优将很容易吸引住粒子群, 因此算法容易发生早熟收敛而陷入局部最优, 算法优化精度难以提高。

## 2 RFRPSO 算法

为了避免算法过快陷入局部最优, 根据优化解的分布特性, RFRPSO 算法引入反向预测因子和斥力因子以改进基本 PSO 算法的速度和位置更新策略。

### 2.1 带反向预测因子的速度更新方式

为降低粒子在运动过程中产生惰性而出现早熟收敛的概率, RFRPSO 算法在迭代的前期阶段, 在速度更新公式中增加一个包含个体最优位置和群体最优位置的反向预测因子。此时, 粒子的速度更新公式为

$$v_{id}^{k+1} = \omega_1 v_{id}^k + c_1 r_1 (p_{id} - x_{id}^k) + c_2 r_2 (p_{gd} - x_{id}^k) - \omega_2 r_3 (p_{id} + p_{gd}). \quad (3)$$

其中: 第 1 项为粒子对自身的记忆能力, 体现 PSO 算法的多样性, 是迭代寻优的基础; 第 2、3 项分别为粒子对自身学习和群体间的协作, 反应集中化特点; 第 4 项为反向预测因子, 是对粒子在运动过程中个体最优和群体最优的反向预测, 以降低粒子在运动过程中陷入局部最优而发生早熟收敛的概率, 是粒子多样化与集中化之间的平衡; 为了简化算法的复杂性, 令  $\omega_1 = \omega_2$ , 均采用非线性递减权重策略;  $r_1$ 、 $r_2$ 、 $r_3$  为  $[0,1]$  区间的随机数。

图 1 为加入反向预测因子后粒子的运动过程。图 1 中, 粒子运动路线为  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ , 其中

阴影部分是粒子下一步可能出现的位置, 与随机变量  $r_1$ 、 $r_2$ 、 $r_3$  的取值相关。

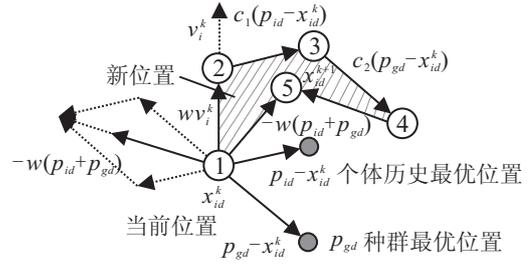


图 1 粒子运动过程

在算法运行后期, 为了使粒子尽量聚焦在小范围内寻优, 提高搜索精度, 应在算法迭代后半段取消反向预测因子的影响, 回到传统的速度更新模式。

### 2.2 带斥力因子的位置更新方式

为了避免种群在算法迭代初始阶段过分聚集, 并使粒子尽量均匀地分散于搜索空间, RFRPSO 算法在粒子位置更新阶段加入了一个带斥力因子的位置修正策略。位置修正过程从效果上看如同粒子间存在一个虚拟斥力, 当间距小于允许值时, 该斥力便产生作用, 将各粒子推至斥力影响范围之外。其主要思想为: 首先, 为粒子的各维设定最小允许间距 ( $\Delta s_1^k, \Delta s_2^k, \dots, \Delta s_D^k$ ), 间距的大小应随迭代次数的增加而逐渐减小, 以体现算法在初始阶段的开掘能力(大范围快速寻优)和搜索后期的开拓能力(小范围局部求精); 然后, 采用传统的位置更新公式对各个粒子的位置进行更新, 并计算每个粒子各维与其他粒子相应维的距离之和 ( $\sum_{i \neq j, j=1}^N \|x_{i1}^k - x_{j1}^k\|, \sum_{i \neq j, j=1}^N \|x_{i2}^k - x_{j2}^k\|, \dots, \sum_{i \neq j, j=1}^N \|x_{iD}^k - x_{jD}^k\|$ ); 最后, 将其与 ( $\Delta s_1^k, \Delta s_2^k, \dots, \Delta s_D^k$ ) 进行比较, 并根据式 (4) 对  $x_{id}^{k+1}$  进行修正。

对于  $\forall d \in [1, D]$ , 有

$$x_{id}^{k+1} = \begin{cases} x_{id}^k + v_{id}^{k+1} + 2\Delta s_d^k, & \sum_{i \neq j, j=1}^N \|x_{id}^k - x_{jd}^k\| < \Delta s_d^k; \\ x_{id}^k + v_{id}^{k+1}, & \sum_{i \neq j, j=1}^N \|x_{id}^k - x_{jd}^k\| \geq \Delta s_d^k. \end{cases} \quad (4)$$

由于粒子在每一维方向上有正负之分, 为消除影响, 在  $\Delta s_d^k$  上需乘以系数 2。图 2 为二维空间中, 粒子  $i$  各维与粒子  $i-1$ 、 $i+1$  相应维的距离之和的计算示意图。其中

$$\Delta d_y(i) = \|y_i - y_{i-1}\| + \|y_i - y_{i+1}\|, \\ \Delta d_x(i) = \|x_i - x_{i-1}\| + \|x_i - x_{i+1}\|.$$

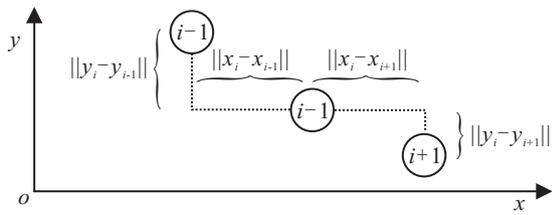


图2 粒子各维间距离总和计算方法

为了简化算法设计过程, 粒子各维最小允许间距取统一数值, 即  $\Delta s_1^k = \Delta s_2^k = \Delta s_D^k = \Delta s^k$ , 变化规律与惯性权重系数  $\omega$  相同, 具体形式为

$$\Delta s^k = s_{ini} - (s_{ini} - s_{end})(k/k_{max})^2, \quad (5)$$

其中  $s_{ini}$  和  $s_{end}$  分别为粒子各维最小允许间距  $\Delta s^k$  的初始值和终止值。

### 2.3 RFRPSO 算法描述

结合带反向预测因子的速度更新方式和带斥力因子的位置更新方式, 给出 RFRPSO 算法详细步骤。

输入: 初始化 RFRPSO 算法参数, 设置种群规模  $N$ , 最大迭代次数  $k_{max}$ , 粒子维数  $d$ , 迭代计数器  $b$ , 学习因子  $c_1, c_2$ , 惯性权重  $\omega_{ini}, \omega_{end}$ , 粒子各维最小允许间距  $s_{ini}, s_{end}$ , 粒子各维速度的阈值  $v_{min}, v_{max}$ , 粒子各维位置的阈值  $x_{min}, x_{max}$ ;

输出: 全局最优  $P_g$ 。

Step 1: 随机初始化第一代种群。

Step 2: 计算粒子的适应度值。

Step 3: 寻找粒子的个体最优  $P_i$  和群体最优  $P_g$ 。

Step 4: 更新种群中粒子的速度和位置。

Step 4.1: 更新粒子速度, 如果迭代次数  $k \leq 0.5k_{max}$ , 则按式 (3) 更新粒子速度, 否则, 按式 (1) 更新速度;

Step 4.2: 更新粒子位置, 按式 (4) 对粒子位置进行更新。

Step 5: 计算状态更新后的种群中各粒子的适应度值, 并更新每个粒子的个体最优  $P_i$  和群体最优  $P_g$ 。

Step 6: 判断是否满足终止条件  $k = k_{max}$ , 若满足, 则转至 Step 7, 否则转至 Step 4, 继续迭代。

Step 7: 输出全局最优  $P_g$ , 算法运行结束。

## 3 实验分析

### 3.1 测试函数和参数设置

为了测试算法的有效性, 选取 5 个高维空间中的复杂多峰函数作为测试函数, 各函数的取值区间、理论最优值如表 1 所示。其中: Rastrigin 和 Ackley 函数具有大量的局部最优点; Schwefel 函数的全局最优与最好的局部最优相距很远, 带有一定的欺骗性。图 3 为 2 维情况下的 Schwefel 函数图像。仿真中, 设置种群规模  $N = 200$ , 最大迭代次数  $k_{max} = 500$ , 维数  $d =$

30, 迭代计数器  $b = 0$ , 学习因子  $c_1 = c_2 = 2$ , 惯性权重  $\omega_{ini} = 0.9, \omega_{end} = 0.4$ , 可接受误差为 0.1, 每个测试函数独立运行 100 次, 其他参数如表 2 所示。

表 1 测试函数的名称、取值区间和理论最优值描述

Function	Search Range	$X$	$f(X)$
Rastrigin	[-5.12, 5.12]	(0, 0, ..., 0)	0
Griewank	[-600, 600]	(0, 0, ..., 0)	0
Ackley	[-32, 32]	(0, 0, ..., 0)	0
Quartic	[-1.28, 1.28]	(0, 0, ..., 0)	0
Schwefel	[-500, 500]	(420.968, 420.968 7)	-12 569.5

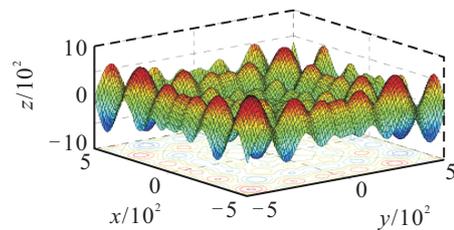


图3 Schwefel 函数的二维图像

表 2 参数设置

Function	$[u_{min}, v_{max}]$	$[x_{min}, x_{max}]$	$s_{ini}$	$s_{end}$
Rastrigin	[-2, 2]	[-5.12, 5.12]	0.5	0.01
Griewank	[-300, 300]	[-600, 600]	60	0.01
Ackley	[-16, 16]	[-32, 32]	3.2	0.01
Quartic	[-0.6, 0.6]	[-1.28, 1.28]	0.128	0.01
Schwefel	[-250, 250]	[-500, 500]	50	0.01

定义判定标准: 1) 成功率 (SuccR), 规定最终解达到可接受误差为成功, 成功率为有效求解次数在总运行次数中的比例; 2) 平均最优值 (MeanBst), 100 次运行完成后得到的算法平均最优值, 可以衡量粒子寻优的平均质量; 3) 最终适应值 (FinalBst), 表示最终收敛时函数的最优值; 4) 平均运行时间 (MeanT/s), 可衡量粒子寻优的时效性; 5) 标准差, 表示每次迭代得到的适应值与平均最优值的差值, 可以衡量粒子寻优的稳定性。

### 3.2 测试和结果分析

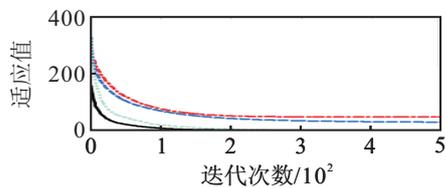
将 RFRPSO 算法的测试结果与基本 PSO 算法、FIPS 算法<sup>[3]</sup>、DBPSO 算法<sup>[13]</sup>进行比较, 4 种算法的测试指标对比如表 3 所示。由表 3 可见, RFRPSO 算法的平均最优值、最终适应值和平均运行时间均小于其他 3 种算法, 且算法成功率显著提高, 充分表明了本文算法的优越性。其主要原因在于通过增加反向预测因子和斥力因子, 使粒子在多样化与集中化之间达到平衡, 能够有效求解复杂多峰函数, 尤其对于 Rastrigin 和 Ackley 函数优化效果十分突出, 不仅获得了全局最优解, 而且成功率和粒子寻优的时效性也优于其他 3 种算法。由于 Quartic 函数本身含有一个随机噪声变量, 本文算法未找到全局最优解, 但也非常接近。

表 3 4 种算法测试指标对比

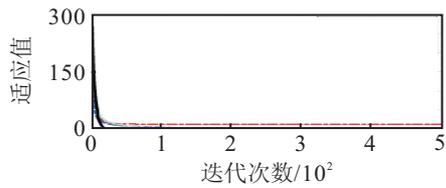
Function	Algorithm	SuccR	MeanBst	FinalBst	MeanT/s
Rastrigin	PSO	0.0000	30.2502	15.8947	8.9301
	FIPS	0.6900	11.7235	6.3971	7.0457
	DBPSO	0.0500	1.1995	0.0000	5.2238
	本文算法	0.8700	0.4595	0.0000	4.6152
Griewank	PSO	0.3700	3.4592	1.6373	23.2084
	FIPS	0.7100	1.1335	1.0238	18.7360
	DBPSO	0.3900	0.9652	0.6358	15.1379
	本文算法	0.9300	0.0000	0.0000	10.9752
Ackley	PSO	0.1800	5.0569	2.7842	10.9132
	FIPS	0.4600	3.6688	0.0890	12.4034
	DBPSO	0.6000	0.5893	0.0160	13.7580
	本文算法	0.8900	0.0000	0.0000	7.0335
Quartic	PSO	0.0000	0.3993	0.1605	5.8854
	FIPS	0.7600	0.0807	0.0129	6.0573
	DBPSO	0.8200	0.0647	0.0163	4.2361
	本文算法	1.0000	0.0010	0.0001	2.5404
Schwefel	PSO	0.0000	-7605.9569	-10014.7961	14.7073
	FIPS	0.2200	-9803.9169	-11285.8006	10.1566
	DBPSO	0.3500	-8698.4179	-11732.0042	12.3019
	本文算法	0.6500	-11309.443	-12569.4845	5.7334

DBPSO 算法和 FIPS 算法差别不大, 在细微的差别中, DBPSO 算法通过动态调整搜索空间边界, 引导粒子在更有效的区域内进行搜索, 找到了 Rastrigin 函数的全局最优解, 且除 Schwefel 函数外, 其平均最优值和最终适应值均小于 FIPS 算法, 但成功率偏低. 需要指出, 4 种算法对 Griewank 函数都取得了较好的效果, 因为当维数空间超过 15 维后, Griewank 函数特性趋向于单峰, 而其他测试函数均是具有大量局部最优点的复杂多峰函数, 所以容易陷入局部最优而难以取得理想的效果.

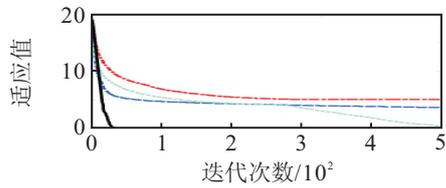
图 4 和图 5 分别为 4 种算法对测试函数的收敛特性曲线图和标准差波动图. 图中: 实线为 RFRPSO 算法, 虚线为 FIPS 算法, 点线为 DBPSO 算法, 点划线为 PSO 算法. 可以看出, RFRPSO 算法通过增加反向测试因子和斥力因子, 及时修正了粒子的速度和位置更新策略, 使其在求解复杂多峰函数时, 具有很好的跳出局部最优和快速收敛到最优解的能力, 无论是解的质量还是算法的稳定性均优于其他算法.



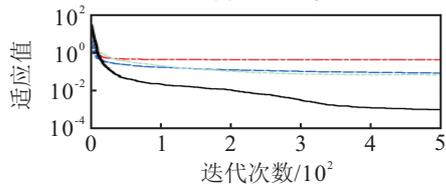
(a) Rastrigin



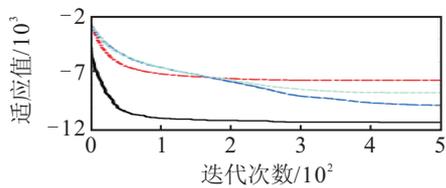
(b) Griewank



(c) Ackley

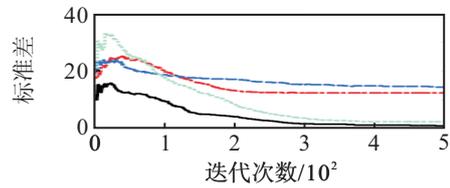


(d) Quartic

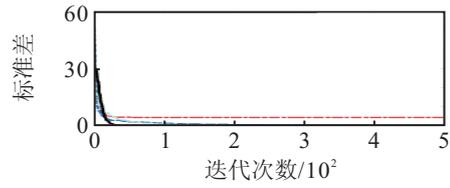


(e) Schwefel

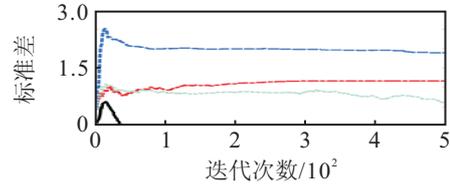
图 4 测试函数收敛特性曲线



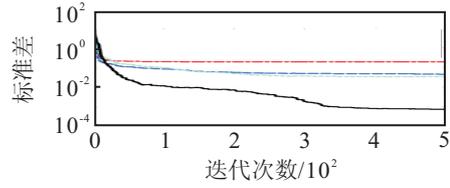
(a) Rastrigin



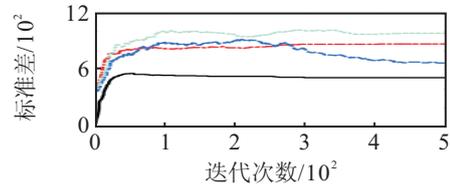
(b) Griewank



(c) Ackley



(d) Quartic



(e) Schwefel

图 5 测试函数标准差波动曲线

## 4 结 论

本文为解决高维空间中复杂多峰函数的优化问题,基于粒子运动的特点,通过增加反向测试因子和斥力因子,改进粒子的速度和位置更新策略,提出RFRPSO优化算法,有效地避免了传统PSO算法在求解复杂多峰函数的过程中容易陷入早熟收敛的问题。由算法性能测试和对比分析可见,较传统PSO算法,RFRPSO算法在整体性能上有了显著提高,寻优能力、收敛速度、求解精度和鲁棒性均有较大改善,适合于处理工程应用中的复杂函数优化问题。如何将RFRPSO算法应用于更复杂的实际多目标优化问题中,以及选取不同参数对优化结果的影响均是尚待探究的问题。

### 参考文献(References)

- [1] Kennedy J, Eberhart R. Particle swarm optimization[C]. Proc of IEEE Int Conf on Neural Networks. Piscataway: IEEE Service Center, 1995: 1942-1948.
- [2] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[C]. Proc of Int Symposium on Micro Machine and Human Science. Piscataway: IEEE Service Center, 1995: 39-43.
- [3] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: Simpler, maybe better[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 204-210.
- [4] Chen Wei-neng, Zhang Jun. A novel set based particle swarm optimization method for discrete optimization problems[J]. IEEE Trans on Evolutionary Computation, 2010, 14(2): 278-300.
- [5] Clerc M, Kennedy J. The particle swarm: Explosion, stability and convergence in multidimensional complex space[J]. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73.
- [6] 张顶学, 关治洪, 刘新芝. 一种动态改变惯性权重的自适应粒子群算法[J]. 控制与决策, 2008, 23(11): 1254-1257. (Zhang D X, Guan Z H, Liu X Z. Adaptive particle swarm Optimization algorithm with dynamically changing inertia weight[J]. Control and Decision, 2008, 23(11): 1254-1257.)
- [7] 黄泽霞, 俞攸红, 黄德才. 惯性权自适应调整的量子粒子群优化算法[J]. 上海交通大学学报, 2012, 46(2): 228-232. (Huang Z X, Yu Y H, Huang D C. Quantum-behaved particle swarm algorithm with self-adapting adjustment of inertia weight[J]. J of Shanghai Jiaotong University, 2012, 46(2): 228-232.)
- [8] Li Ming-wei, Kang Hai-gui, Zhou Peng-fei. Hybrid optimization algorithm based on chaos, cloud and particle swarm optimization algorithm[J]. J of Systems Engineering and Electronics, 2013, 24(2): 324-334.
- [9] Wang Pan-pan, Shi Li-ping, Zhang Yong. A hybrid simplex search and modified bare-bones particle swarm optimization[J]. Chinese J of Electronics, 2013, 22(1): 104-108.
- [10] 刘衍民, 隋常玲, 赵庆祯. 基于K-均值聚类的动态多种群粒子群算法及其应用[J]. 控制与决策, 2011, 26(7): 1019-1025. (Liu Y M, Sui C L, Zhao Q Z. Dynamic multi-swarm particle swarm optimizer based on K-means clustering and its application[J]. Control and Decision, 2011, 26(7): 1019-1025.)
- [11] Chen Yin-yann, Cheng Chen-yang, Wang Li-chih. A hybrid approach based on the variable neighborhood search and particle swarm optimization for parallel machine scheduling problems-A case study for solar cell industry[J]. Int J Production Economics, 2013, 141(1): 66-78.
- [12] 伍大清, 郑建国. 基于混合策略自适应学习的并行粒子群优化算法[J]. 控制与决策, 2013, 28(7): 1087-1093. (Wu D Q, Zheng J G. Improved parallel particle swarm optimization algorithm with hybrid strategy and self-adaptive learning[J]. Control and Decision, 2013, 28(7): 1087-1093.)
- [13] 李迎秋, 迟玉红, 温涛. 一种基于动态边界的粒子群优化算法[J]. 电子学报, 2013, 41(5): 865-870. (Li Y Q, Chi Y H, Wen T. A dynamic boundary based particle swarm optimization[J]. Acta Electronic Sinica, 2013, 41(5): 865-870.)
- [14] 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 X J, Yang Z Z, Zhao M. A uniform searching particle swarm optimization algorithm[J]. Acta Electronic Sinica, 2011, 39(6): 1261-1266.)
- [15] Liang J J, Qin A K. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Trans on Evolutionary Computation, 2006, 10(3): 281-295.
- [16] Zhang H B, Wang H B, Hu Z J. Analysis of particle swarm optimization algorithm global convergence method[J]. Computer Engineering and Applications, 2011, 47(34): 61-63.
- [17] Chen Dong-hui, Liu Zhi-jing, Wang Zong-hu. A fuzzy similarity-based clustering optimized by particle swarm optimization[J]. Chinese J of Electronics, 2013, 22(3): 461-465.

(责任编辑: 郑晓蕾)