

粒子群优化算法中的分步式策略

胡建, 李志蜀, 欧鹏, 罗思达

(四川大学计算机学院 成都 610065)

【摘要】为了解决粒子群优化算法(PSO)在处理高维多极值问题时容易陷入局部最优而早熟的问题,提出了分步式学习策略和分步式评价策略。前者让粒子每次升级只向某一个榜样学习,使粒子能在更有潜力的区域搜索;并简化了其升级规则,使粒子的搜索行为更易被控制。后者对粒子的位置矢量逐维进行评价,使粒子向目标最优位置“稳步前进”;并通过对维之间的关系进行检测,解决了维不可分解的问题。实验证明,新算法具有很好的收敛速度和抗早熟能力。

关键词 收敛性; 进化算法; 评价策略; 学习策略; 粒子群优化; 分步式策略; 群体智能
中图分类号 TP301 **文献标识码** A **doi**:10.3969/j.issn.1001-0548.2009.03.028

Stepwise Strategies in Particle Swarm Optimization

HU Jian, LI Zhi-shu, OU Peng, and LUO Si-da

(College of Computer Science, Sichuan University Chengdu 610065)

Abstract The particle swarm optimization (PSO) may be trapped in local optima and fail to converge to global optima, especially for multimodal and high-dimensional problems. To handle this problem, a stepwise learning strategy and a stepwise evaluation strategy are presented. The former makes each particle learn from only one particle's historical best information in each update progress in order to search in a potential area, and simplifies particles' update rules to easily control their convergence behaviors. The latter enables each particle to be evaluated in dimension-by-dimension order so as to step steadily toward the destination position, and settles non-separable problems by means of detecting relationships between dimensions. Application of the new PSO on several benchmark optimization problems shows a marked improvement in performance over six other recent variants of the PSO.

Key words convergence; evolutionary algorithms; evaluation strategy; learning strategy; particle swarm optimization; stepwise strategy; swarm intelligence

文献[1]提出的粒子群优化(particle swarm optimization, PSO)^[1]算法是一种全局优化进化算法,该算法形式简单、性能高效,在众多涉及参数优化的问题中广泛应用^[2-3],成为现代工程技术中一种重要的并流行的优化方法。然而,PSO容易陷入局部最优对高维多极值的复杂优化问题。针对该问题,很多研究者提出了不同的改进方法,如调整参数设置^[4]、引入基因操作符^[5]和协作学习^[6],以及采用随机取样的方法模拟粒子轨迹^[7-8]等。但是,这些方法仍然存在如下不足:(1)保留或模拟了原PSO的进化规则,仍然面临着早熟的问题;(2)增加了数学的或逻辑的操作符,增加了进化规则的复杂性,并因此为理解PSO的本质增加了难度。

本文针对粒子的学习策略和位置矢量的评价策略进行研究,提出的方法不仅较目前优秀的PSO算

法具有更好的优化性能,而且其进化规则更为简单。

1 标准PSO

文献[9]把带压缩因子PSO^[10]规定为标准PSO(standard PSO, SPSO),其伪码如下(本文中,全局最优均是指全局最小值):

- (1) {While}停止条件不满足
- (2) For群中的每个粒子*i*
- (3) 确定邻居中最优粒子*l*
- (4) 按式(1)和式(2)分别升级粒子*i*的速度和位置的所有维:

$$V_{id} = k \cdot \text{rand}_1 * (V_{id} + \text{rand}_2 * (\phi/2) * (P_{ld} - X_{id}) + \text{rand}_2 * (\phi/2) * (P_{ld} - X_{id})) \quad (1)$$

$$X_{id} = X_{id} + V_{id} \quad (2)$$

- (5) 评价*X_i*, 如果其适应度小于Pbest_{*i*}, 则更新P_{*i*}和Pbest_{*i*}

收稿日期: 2008-09-12; 修回日期: 2009-03-12

基金项目: 国家科技部中小型科技企业创新基金(06C26225101730); 四川省科技厅重点项目(05GG021-003-2)

作者简介: 胡建(1980-), 男, 博士生, 主要从事智能优化及图像融合方面的研究。

- (6) End For
(7) End While}。

其中,下标*i*、*l*和*d*分别表示粒子、邻居中最优粒子和维的序号; *V*和*X*分别表示速度和位置; *P*表示历史最优位置; *Pbest*表示历史最优位置的适应度; *khi*和*phi*是常数,通常分别取0.729 84和4.1; *rand*₁和*rand*₂是服从均匀分布的范围为[0,1]的随机数。

2 学习策略

文献[1, 4]用社会心理学阐释PSO,认为粒子速度和位置的升级过程就是粒子学习的过程。在SPSO中,如式(1)所示,粒子*i*的学习策略是:同时向“自身经验”*P_i*和“社会经验”*P_l*学习。

2.1 SPSO中的“经验中和”现象

将式(1)变形可得:

$$V_{id} = khi * (V_{id} + rand_3 * (phi/2) * (P_{md} - X_{id})) \quad (3)$$

式中 $rand_3 = rand_1 + rand_2$

$$P_{md} = \frac{rand_1 P_{id} + rand_2 P_{ld}}{rand_1 + rand_2}$$

式(3)表明,SPSO中的学习策略等价于向“自身经验”*P_{id}*和“社会经验”*P_{ld}*二者的“加权经验”*P_{md}*学习。为了讨论方便,下面只考虑一维情形,并去掉下标*d*,但这不失一般性。

由于*rand*₁和*rand*₂是范围为[0,1]的随机数,所以 $\min(P_i, P_l) \leq P_m \leq \max(P_i, P_l)$ 。于是,虽然*P_i*和*P_l*具有较好的适应度,但是如果在*P_i*和*P_l*之间是一个较差的区域,那么*P_m*较差,就会导致经验在加权后质量下降。其实,由于*P_i*和*P_l*分别是粒子*i*的个体历史最优位置和邻居历史最优位置,是“久经考验”的经验,即使没有“在*P_i*和*P_l*之间是一个较差的区域”的条件,*P_m*通常也会比*P_i*和*P_l*差。这种经验质量下降的现象称为“经验中和”。

文献[11]分析了*P_i*和*P_l*不变并且随机数取固定值的情形,理论证明了SPSO中粒子会以*P_m*为中心振荡,且其振幅会越来越小并最终收敛到*P_m*。文献[7, 8]用实验证明了SPSO中粒子轨迹可以用Gaussian或lévy随机取样的方法来模拟,其标准差为 $|P_i - P_l|$ (在lévy中采用有效标准差),其期望值是 $(P_i + P_l)/2$,正是*P_m*的期望值。这些研究结果表明,*P_m*是粒子搜索密度最高的地方。

故“经验中和”现象的存在将会导致粒子集中在一个较差的区域搜索,并相应地减少了在有潜力的区域的搜索,是一种不被期望的现象。

2.2 分步式学习策略

“经验中和”现象存在的原因是粒子同时向多种经验学习,导致经验之间在加权过程中相互牵制或抵消。为了避免该现象,且基于文献[12]中的“fully informed”思想,本文采用新的学习策略:粒子在一次升级过程中只向群体中任意一粒子*j*的历史最优位置学习。该新策略称为分步式学习策略一,即stepwise learning strategy-1(SLS-1),其使粒子*i*在*d*维上的速度升级规则变为:

$$V_{id} = C_1 V_{id} + C_2 \times rand * (P_{jd} - X_{id}) \quad (4)$$

相应地,SPSO中的策略称为同步式学习策略。由于学习策略的变化,式(4)没有采用带压缩因子^[10]的表述形式,其中*C*₁和*C*₂是待定参数,rand是服从均匀分布的范围为[0,1]的随机数。

根据社会心理学可知,式(4)中,*P_{jd}* - *X_{id}*表示粒子*i*向*P_{jd}*学习,*C*₂ × rand是为了阻止粒子*i*瞬间运动到*P_{jd}*的扰动因子,以扩大粒子*i*的搜索范围;*C*₁ *V_{id}*表示粒子的行为受到其原习惯的影响,目的也是为了对粒子的学习造成扰动,以扩大搜索范围^[4]。基于这些含义,式(4)可化简为:

$$V_{id} = P_{jd} - X_{id} + D_{id} \quad (5)$$

式中 *D_{id}*表示对粒子*i*向*P_{jd}*学习的扰动。

本文采用常用的Gaussian变异进一步计算*D_{id}*。根据上述社会心理学含义可知,变异应在*P_{jd}* - *X_{id}*两边对称地进行,且变异大小应正比于*P_{jd}* - *X_{id}*,因此,该Gaussian变异的期望值为0,标准差为*C(P_{jd}* - *X_{id}*),其中*C*为待定系数。把该Gaussian变异用randn(0, *C(P_{jd}* - *X_{id}*))表示,式(5)变为:

$$V_{id} = P_{jd} - X_{id} + randn(0, C(P_{jd} - X_{id})) \quad (6)$$

其对应的学习策略被称为分步式学习策略二(SLS-2)。

合并式(6)和式(2),粒子*i*在*d*维上的位置升级规则最终变为如下形式:

$$X_{id} = P_{jd} + randn(0, C(P_{jd} - X_{id})) \quad (7)$$

式(7)中只含有一个参数*C*,较SPSO的位置升级规则简单很多,而且粒子的搜索行为也更易理解:当*C*值增大,粒子趋向于全局探索;当*C*值减小,粒子趋向于局部开发。本文中,分步式学习策略均指SLS-2。

3 评价策略

如伪码所示,SPSO中位置矢量的评价策略是:粒子*i*的所有维*X_{id}*全部升级后,评价其新的位置*X_i*;如果该位置优于其历史最优位置,则取代之。

3.1 SPSO中的“两进一退”及“两退一进”现象

SPSO中的评价策略把整个位置矢量作为评价对象, 只在意位置矢量每一次升级的始末状态, 而没有深入升级过程内部去注重位置矢量各维的变化, 这会造成下面两个现象: 粒子找到了适应度更好的位置, 但是其中一部分维仍在退化, 即并不是所有维都在进化; 粒子没有找到适应度更好的位置, 但是其中一部分维仍在进化, 即并不是所有维都在退化。它们分别可以被形象地称为“两进一退”和“两退一进”, 都是不被期望的现象, 前者使得一些维被误认为是粒子的历史最优维, 并误导群体的进一步进化; 后者会使粒子在运动过程中“放弃”一些可进化之处。

3.2 分步式评价策略

为了克服上述缺点, 本文采用对位置矢量的每维的每次升级都进行评价的策略, 以对每维的变化提供一种更为频繁的反馈。该策略称为分步式评价策略一(stepwise evaluation strategy-1, SES-1), 相应地, SPSO中的策略被称为同步式评价策略。由于适应度的评价需要全部维, 因此, 在评价位置矢量的某一维时, 还需要确定其他维的取值。本文取当前粒子的历史最优位置的对应维作为其他维的值。为简便起见, 本文只提及对某维进行评价, 隐含其他维的取值。

但是, SES-1面临文献[13]研究遗传算法时所提及的问题, 即现实世界中的待优化问题常常是不可分解的(non-separable), 位置矢量的一些维常常是相关的, 对相关的维中任何一维单独变化将会是低效的。分步式评价策略必须解决“哪些维可以单独评价, 哪些维必须一起评价”的问题。为了解决该问题, 本文将对各维进行检测。

设待优化问题是 $\min f(\mathbf{X})$, 矢量 $\mathbf{X}=[X_1, X_2, \dots, X_D]$, 根据文献[13]中可分解函数的概念, 推理可得如下检测方法, 即对维 X_i , 如果有:

$$\frac{\partial f(\mathbf{X})}{\partial X_i} = g(X_i) \quad (8)$$

式中 $\mathbf{X} \in$ 解空间; $g(X_i)$ 表示 X_i 的任意函数, 则 X_i 可以单独评价, 否则不可单独评价, 而与其他不可单独评价的维一起评价。由于 $f(\mathbf{X})$ 通常在第 i 维上不可偏微分, 式(8)过于苛刻且不具实用性。根据式(8)的物理含义, 即对解空间中的任意 \mathbf{X} , 当只变化 X_i 时, 适应度的变化只与 X_i 有关, 而与其他维无关, 可把式(8)用差分形式表示为:

$$f([X_1, \dots, X_i + \Delta X_i, \dots, X_D]) - f([X_1, \dots, X_i, \dots, X_D]) = f([X'_1, \dots, X_i + \Delta X_i, \dots, X'_D]) - f([X'_1, \dots, X_i, \dots, X'_D]) \quad (9)$$

式中 $\mathbf{X}, \mathbf{X}' \in$ 解空间。

由于检测整个解空间是不现实的, 且通常只关注适应度的变化方向, 因此, 在解空间中随机地取 N 组 \mathbf{X}, \mathbf{X}' , 把式(9)放宽为:

$$f([X_1, \dots, X_i + \Delta X_i, \dots, X_D]) - f([X_1, \dots, X_i, \dots, X_D]) / f([X'_1, \dots, X_i + \Delta X_i, \dots, X'_D]) - f([X'_1, \dots, X_i, \dots, X'_D]) > 0 \quad (10)$$

本文把增加上述检测过程的分步式评价策略称为分步式评价策略二(stepwise evaluation strategy-2, SES-2, SES-2), 由于其较SES-1更具一般性, 本文中, 分步式评价策略均特指SES-2。

4 分布式PSO

分步式学习策略和分步式评价策略的核心思想是一样的, 都是把经验或评价对象“分而治之”, 以避免各成份之间“内耗”。本文把这两种策略统称为分步式策略(stepwise strategy, SwS), 把同时应用这两种策略的PSO称为分步式PSO(stepwise PSO, SwPSO)。在伪码表示上, SwPSO和SPSO的不同之处在于把SPSO的伪码(3)~(5)变化如下:

“随机产生榜样粒子 j ”

用等式(7)对粒子 i 的所有维升级

分别评价所有可以单独评价的维, 如果其适应度小于 $Pbest_i$, 则更新 P_i 的对应维和 $Pbest_i$;

评价需要一起评价的维, 如果其适应度小于 $Pbest_i$, 则更新 P_i 的对应维和 $Pbest_i$ ”。

5 实验及讨论

5.1 实验设计

实验选用5个基准函数^[9,14], 如表1所示, x^* 表示理论全局最优位置, D 表示维数, 在实验中 $D=30$ 。

实验选用9种PSO算法: (1) SPSO; (2) UPSO^[15]; (3) FDRPSO^[16]; (4) FIPS^[12]; (5) CPSO-H^[6]; (6) CLPSO^[14]; (7) 采用分步式学习策略的PSO(stepwise learning PSO, SLPSO); (8) 采用分步式评价策略的PSO(stepwise evaluation PSO, SEPSO); (9) SwPSO。

按照文献[9], 所有算法设置如下: 50个粒子; 允许粒子越界, 越界时不评价; 速度范围设置为搜索空间大小(SLPSO和SwPSO中无此设置, 因为它们

直接采用式(7)而没有速度);最大适应度评价次数(fitness evaluations, FEs)设置为300 000;SPSO和SEPSO采用Ring结构,FIPS采用文献[12]中的获得最高成功率的U-Ring结构;在分步式学习策略中,榜样粒子 j 是在 M 个适应度最小的粒子中均匀随机选取产生的, M 和参数 C 均取变化值,当FEs为0至50 000时, M 和 C 分别从50和1.5线性递减至20和0.5;当FEs

为50 000至250 000时, M 和 C 分别恒为20和0.5;当FEs为250 000至300 000时, M 和 C 分别从20和0.5线性递减至1和0。在分步式评价策略中,如 N 取2,当FEs为5 000 $k(k=0,1,2,\dots)$ 时均作检测,且把检测的空间实时地缩小到粒子群所在的空间范围;最终结果采用独立运行30次后的平均值;其他参数见对应文献。

表1 实验中的基准函数

函数名称及其形式	x^*	$f(x^*)$	搜索范围	初始化范围
Sphere: $f_1(x) = \sum_{i=1}^D x_i^2$	0^D	0	$[-100,100]^D$	$[50,100]^D$
Ackley's: $f_2(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	0^D	0	$[-32,32]^D$	$[16,32]^D$
Rastrigin's: $f_3(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	0^D	0	$[-5.12,5.12]^D$	$[2.56,5.12]^D$

f_4 和 f_5 : 分别是 f_2 和 f_3 的用Salomon方法^[13]进行坐标旋转后的函数 分别与函数 f_2 和 f_3 相同

5.2 实验结果及讨论

实验结果如表2和图1所示。

表2 30次独立实验的平均结果

算法	Sphere	Ackley's	Rastrigin's	Rotated Ackley's	Rotated Rastrigin's
SPSO	$3.89 \times 10^{-57} \pm 5.73 \times 10^{-57}$	$1.55 \times 10^1 \pm 7.91 \times 10^0$	$1.41 \times 10^2 \pm 2.72 \times 10^1$	$1.44 \times 10^1 \pm 8.74 \times 10^0$	$1.81 \times 10^2 \pm 2.72 \times 10^1$
UPSO	$8.18 \times 10^{-105} \pm 2.49 \times 10^{-104}$	$1.97 \times 10^1 \pm 7.57 \times 10^{-2}$	$2.30 \times 10^2 \pm 2.12 \times 10^1$	$1.90 \times 10^1 \pm 3.59 \times 10^0$	$2.57 \times 10^2 \pm 3.03 \times 10^1$
FDRPSO	$4.78 \times 10^{-138} \pm 2.58 \times 10^{-137}$	$8.05 \times 10^{-15} \pm 2.46 \times 10^{-15}$	$2.88 \times 10^1 \pm 7.26 \times 10^0$	$4.31 \times 10^{-1} \pm 6.51 \times 10^{-1}$	$5.12 \times 10^1 \pm 1.42 \times 10^1$
FIPS	$3.00 \times 10^{-15} \pm 1.35 \times 10^{-15}$	$1.44 \times 10^{-8} \pm 2.67 \times 10^{-9}$	$6.54 \times 10^1 \pm 1.04 \times 10^1$	$1.82 \times 10^{-8} \pm 5.55 \times 10^{-9}$	$1.16 \times 10^2 \pm 1.46 \times 10^1$
CPSO-H	$3.62 \times 10^{-32} \pm 5.33 \times 10^{-32}$	$5.20 \times 10^{-14} \pm 1.30 \times 10^{-14}$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$2.06 \times 10^0 \pm 8.64 \times 10^{-1}$	$9.89 \times 10^1 \pm 2.92 \times 10^1$
CLPSO	$1.15 \times 10^{-23} \pm 4.08 \times 10^{-24}$	$3.52 \times 10^{-11} \pm 4.28 \times 10^{-11}$	$6.71 \times 10^{-12} \pm 9.70 \times 10^{-12}$	$9.19 \times 10^{-2} \pm 2.16 \times 10^{-1}$	$4.76 \times 10^1 \pm 7.54 \times 10^0$
SLPSO	$3.15 \times 10^{-132} \pm 1.19 \times 10^{-131}$	$2.05 \times 10^{-14} \pm 3.10 \times 10^{-14}$	$4.12 \times 10^1 \pm 1.29 \times 10^1$	$1.20 \times 10^{-1} \pm 3.73 \times 10^{-1}$	$4.48 \times 10^1 \pm 8.06 \times 10^0$
SEPSO	$2.59 \times 10^{-11} \pm 2.69 \times 10^{-11}$	$8.90 \times 10^{-7} \pm 3.41 \times 10^{-7}$	$2.97 \times 10^{-9} \pm 2.89 \times 10^{-9}$	$1.57 \times 10^0 \pm 2.95 \times 10^0$	$6.64 \times 10^1 \pm 1.02 \times 10^1$
SwPSO	$1.90 \times 10^{-88} \pm 5.32 \times 10^{-88}$	$7.11 \times 10^{-15} \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$8.05 \times 10^{-15} \pm 2.46 \times 10^{-15}$	$4.20 \times 10^1 \pm 8.36 \times 10^0$

(1) 先比较SLPSO和SPSO。从表2可以看出,SLPSO在5个函数上均具有较好的收敛精度。从图1可看出,当评价次数约为50 000次之前时,SPSO较SLPSO具有较快的收敛速度;而当评价次数约为50 000次之后时,SLPSO具有较快的收敛速度;特别地,当评价次数约为50 000次与100 000次之间时,SLPSO的收敛速度明显地快于SPSO。这是因为在进化初期,SLPSO为了加强全局探索能力,把 C 值和榜样粒子 j 的选取范围均设置得较大,虽然较SPSO减慢了早期的收敛速度,但是阻止了算法的早熟;在进化的中后期,随着 C 值和 j 的选取范围逐渐减小,SLPSO从全局探索逐渐趋向局部开发,因没有“经验中和”现象而较SPSO能够在更有潜力的区域搜索,可达到更快的收敛速度。

(2) 再比较SEPSO和SPSO。从表2可以看出,除Sphere函数外,而SEPSO在4个多峰函数上具有较好的收敛精度。从图1可以看出,SPSO较SEPSO在进

化前期具有较快的收敛速度,而在多峰函数的进化

后期具有较慢的收敛速度, 并容易早熟而陷于局部最优; SEPSO在整个进化过程中具有较强抗早熟的能力。这是因为当粒子远离目标最优位置时, SPSO中的位置矢量整体评价的“大踏步”策略对粒子快速飞向目标最优位置是有益的。但是当粒子靠近目标最优位置时, 特别是当解决高维多极值的复杂问题时, 这种“大踏步”的策略就会使粒子因“两进一退”或“两退一进”而缺乏精细搜索的能力, 并导致早熟; 相反地, 在SEPSO中的位置矢量逐维评价的“稳步前进”的策略使粒子仔细搜索, 虽降低了进化前期的收敛速度, 但阻止了进化后期的早熟。

(3) 最后把SwPSO与其他的PSO算法进行比较。在所有算法中, SwPSO在除Sphere函数外的另外4个函数上均具有最好的收敛精度。其原因是SwPSO集成了分步式学习策略和分步式评价策略二者的优点, 使粒子在有潜力区域稳步前进, 不仅具有分步式学习策略的收敛速度, 而且具有分步式评价策略精细搜索和抗早熟的能力。

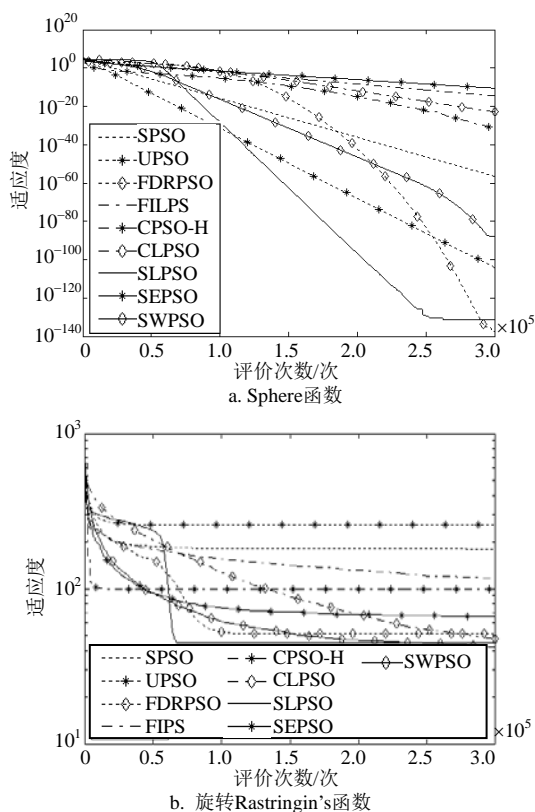


图1 部分函数的适应度进化曲线

6 结论

本文研究了PSO中粒子的学习策略和位置矢量的评价策略, 认为原算法存在“经验中和”和“两进一退, 两退一进”缺陷, 提出了分步式学习策略

和分步式评价策略, 前者让粒子每次升级只向某一经验学习, 使其能在更有潜力的区域搜索, 加快了算法的收敛速度; 后者对粒子的位置矢量逐维进行评价, 并提出了对维不可分解问题进行维之间关系的检测方法, 使粒子“稳步前进”, 很好地解决算法的早熟问题。本文还把分步式学习策略和分步式评价策略统一起来, 充分利用了前者的快速收敛能力和后者的抗早熟能力, 使算法达到了很好的优化性能。

新算法的另一个优点是升级规则更为简单, 对粒子轨迹的理解更为容易, 并能很方便地和其他的PSO改进方法结合。

参 考 文 献

- [1] KENNEDY J, EBERHART R C. Particle swarm optimization[C]//Proceedings of the IEEE International Conference on Neural Networks. Perth, Australia: [s.n.], 1995: 1942-1948.
- [2] MAO Yan, QIU Jie, LIU Xin-ying, et al. Structural parameter optimization of multilayer conductors in HTS cable[J]. Journal of Electronic Science and Technology of China, 2008, 6(2): 112-118.
- [3] 刁 鸣, 高洪元, 马 杰, 等. 应用神经网络粒子群法的多用户检测[J]. 电子科技大学学报, 2008, 37(2): 178-180, 281. DIAO Ming, GAO Hong-yuan, MA Jie, et al. Multi-user detection based on particle swarm optimization algorithm with neural network[J]. Journal of University of Electronic Science and Technology of China, 2008, 37(2): 178-180, 281.
- [4] SHI Y, EBERHART R C. A modified particle swarm optimizer[C]//Proceedings of the 1998 IEEE Conference on Evolutionary Computation. [S.l.]: [s.n.], 1998: 66-73.
- [5] WANG H, LI H, LIU Y, et al. Opposition-based particle swarm algorithm with Cauchy mutation[C]//Proceedings of the 2007 IEEE Congress on Evolutionary Computation. [S.l.]: [s.n.], 2007: 4750-4756.
- [6] van den Bergh F, ENGELBRECHT A P. A cooperative approach to particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 225-239.
- [7] KENNEDY J. Bare bones particle swarms[C]//Proceedings of the 2003 IEEE Swarm Intelligence Symposium. Indianapolis, USA: [s.n.], 2003: 80-87.
- [8] RICHER T J, BLACKWELL T M. The lévy particle swarm[C]//Proceedings of 2006 IEEE Congress on Evolutionary Computation. Vancouver, BC, Canada: IEEE, 2006: 808-815.
- [9] BRATTON D, KENNEDY J. defining a standard for particle swarm optimization[C]//Proceedings of the 2007 IEEE Swarm Intelligence Symposium. [S.l.]: [s.n.], 2007: 120-127.
- [10] CLERC M, KENNEDY J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(1): 58-73.

- [11] van den Bergh F, ENGELBRECHT A P. A study of particle swarm optimization particle trajectories[J]. Information Sciences, 2006, 176(8): 937-971.
- [12] MENDES R, KENNEDY J, NEVES J. The fully informed particle swarm: simpler, maybe better[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 204-210.
- [13] SALOMON R, Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions[J]. Bio Systems, 1996, 39: 263-278.
- [14] LIANG J J, QIN A K, SUGANTHAN 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.
- [15] PARSOPOULOS K E, VRAHATIS M N. UPSO: A unified particle swarm optimization scheme[C]//Proceedings of the International Conference of Computational Methods in Sciences and Engineering 2004. [S.l.]: Attica, 2004: 868-873.
- [16] PERAM T, VEERAMACHANENI K, MOHAN C K. Fitness-distance-ratio based particle swarm optimization [C]//Proceedings of Swarm Intelligence Symposium, [S.l.]: [s.n.], 2003: 174-181.

编辑 熊思亮