【电子与信息科学 / Electronics and Information Science】

基于正态分布衰减惯性权重的粒子群优化算法

徐浩天, 季伟东, 孙小晴, 罗 强

哈尔滨师范大学计算机科学与信息工程学院, 黑龙江哈尔滨 150025

摘 要: 针对粒子群优化 (particle swarm optimization, PSO) 算法无法在提高收敛速度的同时避免早熟的缺陷, 提出基于正态分布衰减惯性权重粒子群优化 (normal distribution decay inertial weight particle swarm optimization, NDPSO) 算法. 以正态分布曲线作为惯性权重的衰减策略曲线, 通过引入控制因子对粒子的位置进行改善, 使得 NDPSO 算法能很好的在优化过程中平衡全局搜索和局部搜索能力. 使用 8 个标准函数测试分别对粒子群优化 (particle swarm optimization, PSO)、线性权重衰减粒子群优化 (linear decay inertial weight particle swarm optimization, LDWPSO)、指数权重衰减粒子群优化 (exponential decay weight particle swarm optimization, EXPPSO)、收缩因子粒子群优化 (constriction factor particle swarm optimization, CFPSO)、高斯分布衰减惯性权重粒子群优化 (Gaussian decay inertial weight particle swarm optimization, GDIWPSO)、基于动态加速度系数的粒子群优化 (particle swarm optimization based on dynamic acceleration coefficients, PSO-DAC)、惯性权重自适应粒子群优化 (inertia weight adaptive particle swarm optimization, 简称 PSO-LH) 算法以及 NDPSO 算法进行仿真, 分析他们的收敛速度和收敛精度. 结果表明, NDPSO 算法不管在单峰函数问题还是多峰函数问题上, 总体性能都优于其他算法.

关键词: 人工智能; 群体智能算法; 粒子群算法; 惯性权重; 正态分布; 衰减策略

中图分类号: TP391

文献标志码: A

doi: 10.3724/SP.J.1249.2020.02208

A PSO algorithm with inertia weight decay by normal distribution

XU Haotian, JI Weidong, SUN Xiaoqing, and LUO Qiang

School of Computer Science and Information Engineering, Harbin Normal University,
Harbin 150025, Heilongjiang Province, P. R. China

Abstract: To avoid early maturity stagnation while increasing convergence speed of particle swarm optimization (PSO) algorithm, a normal distribution decay inertial weight particle swarm optimization (NDPSO) algorithm is proposed based on the inertia weight of which the decay strategy curve is normal distribution curve. By introducing the control factor to improve the position of the particle, the NDPSO algorithm can balance the global search and local search ability in the optimization process. The PSO, linear decay inertial weight particle swarm optimization (LDWPSO), exponential decay weight particle swarm optimization (EXPPSO), constriction factor particle swarm optimization (CFPSO), Gaussian decay inertial weight particle swarm optimization (GDIWPSO), particle swarm optimization based on dynamic acceleration coefficients (PSO-DAC), inertia weight adaptive particle swarm optimization (PSO-LH) and NDPSO algorithms are simulated using 8 standard function test evaluations, and their

Received: 2019-04-06; **Accepted:** 2019-05-27

Foundation: National Natural Science Foundation of China (31971015); Harbin Science and Technology Bureau Special Fund for Scientific and Technological Innovation Research (2017RAQXJ050); Innovation Research Project of Master's Degree in Harbin Normal University (HSDSSCX2019-08)

Corresponding author: Professor JI Weidong. E-mail: Kingjwd@126.com

Citation: XU Haotian, JI Weidong, SUN Xiaoqing, et al. A PSO algorithm with inertia weight decay by normal distribution [J]. Journal of Shenzhen University Science and Engineering, 2020, 37(2): 208-213. (in Chinese)

convergence speed and convergence precision are analyzed. The results show that the NDPSO algorithm outperforms other algorithms in terms of single-peak function or multi-peak function.

Key words: artificial intelligence; swarm intelligence algorithm; particle swarm optimization; inertia weight; normal distribution; attenuation strategy

粒子群优化 (particle swarm optimization, PSO) 算法^[1]在群体智能算法领域中发展相当成熟, 与遗传算法 (genetic algorithm, GA)^[2]、改进的 GA 算法^[3]以及蚁群算法 (ant colony optimization algorithm, ACO)^[4]相比, PSO 算法具有原理简单、参数少和容易实现等优点. 虽然目前已被广泛用于系统辨识^[5]和神经网络训练^[6]等热门领域, 但是仍存在各种缺陷. 针对收敛速度慢和易早熟收敛的缺陷, 研究人员提出了 3 种改进方法: ① 改进种群的多样性. KENNEDY 等^[7-8]提出不同的拓扑结构 PSO 算法, 通过改变种群大小、粒子速度更新的来源和粒子分布结构, 使粒子能更好找到个体极值和全局最优解; VESTERSTROM 等^[9]设计的分工 PSO 和 BERGH 等^[10]提出的合作 PSO, 都是使用不同的父子代更新策略丰富种群的多样性, 达到阻止早熟收敛的目的; DENG 等^[11]提出子种群协作的改进 PSO 算法, 通过不同子种群之间的相互协作, 达到提高收敛性能的效果. ② 改进参数设置. SHI 等^[12]提出线性权重衰减粒子群优化 (linear decay inertial weight particle swarm optimization, LDWPSO) 算法, 随着迭代次数的增加进行线性衰减的惯性权重, 由于权重衰减步长固定, 所以 LDWPSO 的初始化需要根据使用者的经验来设置; YAN 等^[13]提出指数权重衰减粒子群优化 (exponential decay weight particle swarm optimization, EXPPSO) 算法, 基于指数形式对惯性权重进行了衰减; CLERC 等^[14]提出的收缩因子粒子群优化 (constriction factor particle swarm optimization, CFPSO) 算法引入了一个收缩因子 χ , 可以平衡传统粒子群优化算法局部和全局搜索的能力, 但是, 收缩因子作用有限, 无惯性权重那样能直接影响局部和全局搜索的能力; ZHANG 等^[15]提出高斯分布衰减惯性权重粒子群优化 (Gaussian decay inertial weight particle swarm optimization, GDIWPSO) 算法, 受高斯分布函数启发, 将其与粒子群算法结合, 并设置阈值 k 来控制权重曲线的变化率, 使惯性权重动态变化来满足粒子群算法在不同时期对粒子搜索的需求. ③ 与智能方法结合, 混合不同的优化策略. 通过借鉴其他优化算法的策略和机制来提高 PSO 算法性能. 例如, ZHOU

等^[16]提出基于邻域重心进行反向学习的 PSO 算法, 不仅拓展了粒子空间搜索范围, 还增加了得到高质量解的概率; SONG 等^[17]提出结合阵列信号模型和随机最大似然算法的 PSO, 解决了传统 PSO 算法中粒子数多和迭代次数多、计算复杂度高的问题.

本研究使用改进参数设置方式, 提出正态分布衰减惯性权重粒子群优化 (normal distribution decay inertial weight particle swarm optimization, NDPSO) 算法, 针对惯性权重进行改进. NDPSO 算法以正态分布曲线进行权重衰减, 该正态分布曲线是一种非线性衰减曲线, 早期衰减缓慢, 中期衰减迅速, 后期衰减趋于平缓, 且权重衰减步数动态变化. 该衰减策略充分利用了惯性权重可以提升传统 PSO 性能的作用. 通过计算机仿真对比 PSO、LDWPSO、EXPPSO、CFPSO、GDIWPSO、基于动态加速度系数的粒子群算法 (particle swarm optimization based on dynamic acceleration coefficients, PSO-DAC)^[18]和 LUO^[19]提出的惯性权重自适应粒子群算法 (inertia weight adaptive particle swarm optimization, 本研究简称 PSO-LH) 以及 NDPSO 算法在 8 个标准测试函数中的收敛速度和收敛精度, 证明 NDPSO 算法综合性能最佳.

1 粒子群优化算法及其改进算法

基本 PSO 算法中, 速度和位移更新公式^[1]为

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 [pbest(t) - x_{id}(t)] + c_2 r_2 [gbest(t) - x_{id}(t)] \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2)$$

其中, $v_{id}(t+1)$ 和 $x_{id}(t+1)$ 分别为粒子 i 在第 $t+1$ 代的第 d 维速度和位置; c_1 和 c_2 为学习率; r_1 和 r_2 为 $[0, 1]$ 间均匀分布的随机数; $pbest(t)$ 为粒子个体在第 t 代所得的最优位置信息; $gbest(t)$ 为种群在第 t 代所得的最优位置信息; 惯性权重 ω 计算式^[12]为

$$\omega = \omega_{\max} - t \frac{\omega_{\max} - \omega_{\min}}{t_{\max}} \quad (3)$$

其中, $\omega_{\max}$ 和 $\omega_{\min}$ 分别为最大和最小加权系数; $t_{\max}$ 为最大迭代次数. 文献[12]的实验证明该参数能加

快算法的收敛速度,且动态变化的 ω 比固定的 ω 效果更好,当 ω 从0.9动态衰减到0.4时性能最佳.

EXPPSO 算法^[13]中的 ω 与LDWPSO算法中的 ω 不同, ω 被递减的指数函数 $e^{-\alpha_E}$ 替代,其中, α_E 为影响移动步长的大小,由当前迭代次数控制,则速度公式修改为

$$v_{id}(t+1) = e^{-\alpha_E} v_{id}(t) + c_1 r_1 [pbest(t) - x_{id}(t)] + c_2 r_2 [gbest(t) - x_{id}(t)] \quad (4)$$

由于指数函数的特点,算法前期搜索步长较大,扩大了搜索区域,保证了前期的全局搜索能力.后期较小的步长减缓粒子更新速度,避免极值问题.YAN等^[13]通过实验证明,指数衰减策略可以很快找到最优解.

受非线性指数权重衰减的启发,本研究提出的NDPSO算法将从线性衰减和非线性衰减两个角度进行研究.其中, ω 是以正态分布曲线进行衰减.在前期一定次数的迭代过程中, ω 始终保持较大的数值,使算法的全局搜索能力持续最大,令算法能快速收敛到最优解所在的区域内.中期 ω 迅速衰减,全局搜索能力慢慢减弱,局部搜索能力逐渐增强,并最终使得两种搜索能力保持动态平衡.后期 ω 减小到一定值后,算法不断迭代,令其局部搜索能力达到最强.可见,该算法充分利用了 ω 对PSO算法收敛速度和能力的影响,动态变化的 ω 引导算法避免陷入局部最优,并且保证了收敛的速度.

2 NDPSO 算法

针对现有的惯性权重线性衰减PSO算法存在的衰减机制落后、速度较慢和易陷入局部最优等缺点,本研究引入正态分布曲线提出NDPSO算法,既保证了粒子群优化算法速度快的特性,还较好的克服了易陷入局部最优的缺点.

正态分布也叫常态分布,是连续随机变量概率分布的一种,在自然界和人类社会普遍存在.一般正态分布的函数公式为

$$f(x) = \frac{1}{\sqrt{2\pi\theta}} e^{-\frac{(x-\mu)^2}{2\theta^2}} \quad (5)$$

其中, x 为随机变量; μ 为正态分布的位置参数,描述正态分布的集中趋势位置; θ 描述了正态分布数据分布的离散程度.

为寻找合适的正态分布曲线,满足衰减机制,首先针对正态分布曲线的核心参数进行仿真分析.

取 $\theta = 0.2000 \sim 0.7000$,由正态分布的算法机制可知, θ 越大,图形越扁平,如图1.从图1可见,当 $\theta = 0.2000$ 时,曲线十分陡峭,随着 θ 增至0.4000~0.5000时,曲线基本满足NDPSO算法的要求.文献[11]通过大量实验证明,若 ω 以线性或非线性衰减,则0.9~0.4是较佳衰减值.本研究经大量仿真实验得到 $\mu = 0$, $\theta = 0.4433$ 为最佳值.故为使 $x = 0$ 时, $f(x) = 0.9$,设 $\theta = 0.4433$.此时,正态分布曲线 ω 从0.9开始递减.将式(5)应用到标准PSO算法中,得到 ω 的非线性变化公式,如式(6),速度更新则沿用式(1).

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \frac{1}{\sqrt{2\pi\theta}} e^{-\frac{(t/t_{\max})^2}{2\theta^2}} \quad (6)$$

其中, $t_{\max}$ 为最大迭代次数; t 为当前迭代次数; $\theta = 0.4433$.

图1 正态分布曲线图

Fig. 1 (Color online) Normal distribution picture

为适应 ω 的变化特点,引入如式(7)的 λ ,并采用式(8)为粒子群位置公式,使算法的搜索步长随着粒子的适应度值变化而动态变化.

$$\lambda = \frac{1}{1 + e^{\frac{f(x_{id})}{\bar{f}(x)}}} \quad (7)$$

$$x_{id}(t+1) = x_{id}(t) + \lambda v_{id}(t+1) \quad (8)$$

其中, $f(x_{id})$ 为当前粒子的适应度; $\bar{f}(x)$ 为当前所有粒子适应度的平均值; $x_{id}(t)$ 为当前粒子的位置; $v_{id}(t)$ 为当前粒子的速度.

从图1可见,当 $x > 0$ 时,正态分布曲线呈“Z”型.在迭代刚开始时,由于初期整个解空间很大,为了加快收敛速度, ω 保持0.9或比0.9略小的值,令算法以一个较大的步长搜索整个解空间,全局搜索能力此时达到最大,且收敛速度增加,也不易陷入局部最优.在经过一定的迭代次数后, ω

开始迅速衰减, 全局搜索能力逐步减弱; 随着 ω 减小, 搜索步长减小, 算法达到全局搜索和局部搜索的平衡. 当迭代将要结束时, ω 的值已经极小, 此时算法进入局部搜索, 步长小, 搜索不会遗漏最优解, 由于后期能搜索的解空间已经相当小, 所以收敛速度不会降低, 这就令算法在保证速度的同时, 能精确找到最优解.

算法步骤描述为:

- 1) 初始化种群中粒子的位置和速度;
- 2) 将粒子的 pbest 设置为当前个体最优值, 将 gbest 设置为初始群体中全局最优;
- 3) 计算每个粒子的适应度值;
- 4) 对每个粒子, 若粒子适应度值优于 pbest, 则更新 pbest;
- 5) 对每个粒子, 若粒子适应度值优于 gbest, 则更新 gbest;
- 6) 根据当前迭代次数和式(6)计算 ω 的值, 根据式(7)更新 λ 的值;
- 7) 用式(1)和式(9)更新粒子的速度和位置;
- 8) 判断算法终止准则是否满足, 如果满足, 算法停止; 否则转向3).

NDPSO 算法的核心部分是 ω 的更新衰减策略, 每次迭代都使用新的 ω 值来更新粒子的速度和位置. 这样局部搜索能力和全局搜索能力在算法优化的每个阶段都能发挥应有的作用. 这种“Z”型曲线使算法不易陷入局部最优, 且收敛速度很快.

3 仿真实验

通过对8个典型标准测试函数(表1, 各函数具体表达式请扫描论文末页右下角二维码查看)的测试, 比较 NDPSO 算法与其他改进 PSO 算法的性能. 其中, f_1 到 f_4 为单峰函数, $f_1(x)$ 是一个简单的单峰二次函数, 最小值为0, $f_3(x)$ 的全局最小值位于抛物线形山谷中, 很难找到最小值; $f_5(x)$ 到 $f_8(x)$ 为多峰函数, $f_6(x)$ 和 $f_7(x)$ 都具有多个局部最小值, 很难找到最小值. 对比算法分别为 PSO、LDWPSO、CFPSO、EXPPSO、PSO-DCA 和 PSO (LH).

3.1 参数设置

PSO 算法的参数有3种, ω 、 r_1 和 r_2 、 c_1 和 c_2 . 其中, ω 是控制当前速度对新产生速度的影响, 用于平衡局部搜索和全局搜索能力; r_1 和 r_2 是 $[0, 1]$ 内的随机数; c_1 和 c_2 为学习率, 用来控制自身记忆和整个种群之间的相对影响.

表1 八个标准测试函数

Table 1 Eight benchmark test functions

测试函数	函数名	定义域范围	最优值
$f_1(x)$	Sphere	$[-100, 100]$	0
$f_2(x)$	Schwefel	$[-10, 10]$	0
$f_3(x)$	Rosenbrock	$[-30, 30]$	0
$f_4(x)$	Step	$[-100, 100]$	0
$f_5(x)$	Ackley	$[-32, 32]$	0
$f_6(x)$	Rastrigin	$[-5.12, 5.12]$	0
$f_7(x)$	Griewank	$[-600, 600]$	0
$f_8(x)$	Quartic	$[-1.28, 1.28]$	0

为保持实验的一致性, 设置搜索维数为30, 种群规模为30, 最大迭代次数为1 000, $c_1 = c_2 = 2$, $\pi = 3.141 59$, $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$, 其余参数分别按照对应论文设置. 每个算法计算20次, 最优解取平均值. 文献[12]指出, ω 从0.9到0.4进行线性与非线性曲线衰减最佳, 故为充分发挥正态分布曲线的优势, 本研究设 NDPSO 的 $\mu = 0$, $\theta = 0.443 3$.

3.2 仿真结果及比较

通过 Matlab 仿真, 得到不同算法在8个测试函数中的收敛曲线如图2. 8种算法的最优解如表2. 其中, 灰底数据为8个算法中精度最高的解.

函数 $f_1(x)$ 到 $f_4(x)$ 是简单的单峰测试函数, 对比图2(a)至(d)和表2可见, NDPSO 算法在保证收敛速度的同时, 收敛精度也远大于其他改进 PSO 算法. 由于 $f_1(x)$ 至 $f_4(x)$ 测试函数的单峰性, 函数求解相对简单, EXPPSO 和 CFPSO 算法的收敛速度可得到最大发挥, NDPSO 算法的收敛速度次之. 但是, EXPPSO 和 CFPSO 算法是通过牺牲收敛精度换取速度, 导致性能不如 NDPSO 和 LDWPSO 算法. 而且从图2(a)至(d)可见, 随着迭代次数的增加, NDPSO 算法的收敛曲线下落明显, 收敛速度和步长随着迭代次数动态调整, 保证了算法在加快收敛速度的同时仍能保持最优解的精度. LDWPSO 算法的收敛精度虽略次于 NDPSO 算法, 但收敛速度大不如 NDPSO 算法. 与 NDPSO 算法机理相似的 GDIWPSO 算法, 虽然较其他算法有兼顾速度和精度的优势, 但 NDPSO 算法综合效果更优. PSO-DAC 和 PSO (LH) 算法无论是收敛速度还是收敛精度都远逊于 NDPSO 算法, 这两种算法的惯性权重机制都未能发挥 ω 的作用. 在单峰函数求解上, NDPSO 算法能同时满足收敛速度和收敛精度要求.

图 2 不同算法在 8 个测试函数中的收敛曲线图

Fig. 2 (Color online) Convergence curves of different algorithms in eight test functions

表 2 不同算法在 8 个测试函数中的平均最优解¹⁾Table 2 Average optimal solution of different algorithms in eight test functions¹⁾

算法	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$
PSO	2.22×10^2	1.62×10^2	2.33×10^4	8.42×10^3	$2.740 0 \times 10^2$	1.63×10^2	1.38×10^2	$6.650 0 \times 10^{-2}$
LDWPSO	1.41×10^2	7.179 0	2.17×10^4	1.04×10^2	6.322 4	1.05×10^2	1.870 0	$8.530 0 \times 10^{-2}$
CFPSO	1.85×10^3	2.69×10	7.17×10^5	1.46×10^3	9.476 7	1.45×10^2	2.22×10	$7.300 0 \times 10^{-1}$
EXPPSO	6.93×10^3	2.23×10	3.54×10^6	5.77×10^3	13.799 9	1.63×10^2	4.00×10	1.167 9
GDIWPSO	6.04×10^2	4.22×10	1.23×10^4	8.92×10^2	6.728 1	9.08×10	4.993 2	$1.210 0 \times 10^{-1}$
PSO-DAC	3.19×10^2	4.47×10	1.46×10^6	7.47×10^3	$1.360 0 \times 10$	1.35×10^2	5.50×10	1.039 1
PSO(LH)	2.12×10^3	2.89×10	5.60×10^5	3.85×10^3	$1.280 0 \times 10$	1.36×10^2	3.00×10	1.388 6
NDPSO	8.57×10	1.60×10	1.45×10^3	1.50×10^2	8.690 1	4.84×10	2.438 7	$5.250 0 \times 10^{-2}$

¹⁾带阴影的数据为 8 个算法中精度最高的解

函数 $f_5(x)$ 到 $f_8(x)$ 是较为复杂的多峰函数, 算法很容易陷入局部最优, 导致无法收敛. 结合图 2 (e) 至 (h) 和表 2 发现, NDPSO 算法在多峰函数上性能依然出色. 在收敛速度上, NDPSO 算法的表

现比其他改进的 PSO 算法更好, 且收敛精度也大幅提高. 从图 2 (f) 和 (h) 可见, NDPSO 算法收敛速度最快且找到的最优解精度最高. 与之相比, 线性权重衰减策略的衰减机制在复杂的多峰函数只能保

证收敛精度, 不能保证快速收敛. 由于多峰函数的特殊性, 求解较为复杂, 极易陷入局部最优. 而NDPSO算法几乎不会陷入局部最优, 其 ω 衰减机制保证了前期的收敛步长, 防止陷入局部最优导致算法无法进行.

综上所述, NDPSO算法不管在单峰函数问题还是多峰函数问题上, 总体性能都优于PSO、LDWPSO、CFPSO、EXPPSO、GDIWPSO、PSO-DAC和PSO(LH)算法.

结 语

本研究利用正态分布的曲线特征, 提出了一种基于正态分布衰减惯性权重的粒子群算法NDPSO. 该算法前期缓慢收敛, 中期加速收敛, 后期趋于平缓, 保证了算法在前期的全局搜索能力, 中期局部搜索能力与全局搜索能力的平衡, 后期强化了局部搜索能力. 理论分析和实验数据比较发现, 其他线性衰减权重PSO、收缩因子PSO和高斯分布PSO相比, NDPSO算法在简单的单峰函数测试中有着不俗的收敛速度和收敛精度, 而在多峰函数测试中也始终能收敛到最优解和保持较快的收敛速度, 说明NDPSO充分利用了 ω 对PSO算法收敛机制的影响, 能解决多种优化问题.

基金项目: 国家自然科学基金资助项目(31971015); 哈尔滨市科技创新人才研究专项资助项目(2017RAQXJ050); 哈尔滨师范大学硕士研究生创新科研资助项目(HSDSSCX2019-08)

作者简介: 徐浩天(1996—), 哈尔滨师范大学硕士研究生. 研究方向: 群体智能算法. E-mail: 15189871428@163.com

引 文: 徐浩天, 季伟东, 孙小晴, 等. 基于正态分布衰减惯性权重的粒子群优化算法[J]. 深圳大学学报理工版, 2020, 37(2): 208-213.

参考文献 / References:

- [1] KENNEDY J, EBERHART R C. Particle swarm optimization [C]// Proceedings of 4th IEEE International Conference on Neural Networks. Perth, Australia: IEEE, 1995: 1942-1948.
- [2] GOLDBERG D E. Genetic algorithms in search optimization and machine learning [M]. Reading Mass, USA: Addison-Wesley, 1989: 95-99.
- [3] JI Weidong, WANG Jianhua, ZHANG Jun. An improved real hybrid genetic algorithm [J]. Tehnicki Vjesnik-Tehnicka Gazette, 2014, 21(5): 979-986.
- [4] DORIGO, MANIEZZO M, COLORIN V. A ant system: optimization by a colony of cooperating agents [J]. IEEE Transactions on SMC, 1996, 26(1): 8-41.
- [5] KARABOGA N. A new design method based on artificial bee colony algorithm for digital IIR filters [J]. Journal of the Franklin Institute-engineering and Applied Mathematics, 2009, 346(4): 328-348.
- [6] PANG Hui, LIU Fan, XU Zeren. Variable universe fuzzy

control for vehicle semi-active suspension system with MR damper combining fuzzy neural network and particle swarm optimization [J]. Neurocomputing, 2018, 306: 10-140.

- [7] KENNEDY J. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance [C]// Proceedings of IEEE Congress on Evolutionary Computation. Piscataway, USA: IEEE, 1999: 1931-1938.
- [8] KENNEDY J, Mendes R. Population structure and particle swarm performance [C]// Proceedings of IEEE Congress on Computational Intelligence. Hawaii, USA: IEEE, 2002: 1671-1676.
- [9] VEDTERDTROM J S, RIGET J, KRINK T. Division of labor in particle swarm optimization [C]// Proceedings of the Congress on Evolutionary Computation. Honolulu, USA: IEEE, 2002: 1570-1575.
- [10] BERGH F V D, ENGELBRECHT A P. A cooperative approach to particle swarm optimization [C]// Proceedings of IEEE Transactions on Evolutionary Computation. Portland, USA: IEEE, 2004: 225-239.
- [11] DENG Xianli, WEI Bo, ZENG Hui, et al. A multi-population based self-adaptive migration PSO [J]. Acta Electronica Sinica, 2018, 46(8): 1858-1865.
- [12] SHI Y, EBERHART R C. A modified particle swarm optimizer [C]// Proceedings of IEEE Congress on Evolutionary Computation. Piscataway, USA: IEEE, 1998: 69-73.
- [13] YAN Chunman, LU Genyuan, LIU Yingting, et al. A modified PSO algorithm with exponential decay weight [C]// Proceedings of the 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery. Changsha, China: IEEE, 2018: 239-242.
- [14] CLERC M, KENNEDY J. The particle swarm-explosion, stability and convergence in a multidimensional complex space [J]. Proceedings of IEEE Transactions on Evolutionary Computation, 2002, 6(1): 58-73.
- [15] ZHANG Xun, WANG Ping. Particle swarm optimization algorithm based on Gaussian function decreasing inertia weight [J]. Application Research of Computers, 2012, 29(10): 3710-3724.
- [16] ZHOU Lingyun, DING Lingxin, PENG Hu, et al. Neighborhood centroid opposition-based particle swarm optimization [J]. Acta Electronica Sinica, 2017, 45(11): 2815-2823.
- [17] SONG Huajun, LIU Feng, CHEN Haihua, et al. A stochastic maximum likelihood algorithm based on improved PSO [J]. Acta Electronica Sinica, 2017, 45(8): 1990-1994.
- [18] 滕志军, 吕金玲, 郭力文, 等. 基于动态加速因子的粒子群优化算法研究[J]. 微电子学与计算机, 2017, 34(12): 125-129.
- [19] TENG Zhijun, LV (Lyn) Jinling, GUO Liwen, et al. Research on particle swarm optimization based on dynamic acceleration coefficients [J]. Microelectronics & Computer, 2017, 34(12): 125-129. (in Chinese)
- [19] LUO Hua. An inertia weight adaptive particle swarm optimization algorithm [J]. Electronic Science and Technology, 2017, 30(2): 30-36.

【中文责编: 英 子; 英文责编: 木 柯】

补充材料

<http://journal.szu.edu.cn>