

文章编号: 2095-2163(2021)07-0006-07

中图分类号: TP301

文献标志码: A

一种改进的粒子群优化算法

胡建华, 熊伟利

(上海理工大学 理学院, 上海 200093)

摘要: 粒子群优化算法(PSO)是一种群体智能进化计算方法,但在搜索过程中粒子紧跟最优粒子运动降低了粒子多样性和全局搜索能力,从而易陷入局部极值。本文提出一种新的粒子群优化算法(PSO-EWD),主要改进体现在2个方面:将惯性权重与进化因子相关联,根据种群的进化状态而改变权重大小,以平衡全局搜索能力与局部搜索能力;将时变的分布式时延引入速度更新公式中,以增加粒子的多样性。本文通过5种算法在9个基准函数上的实验对比,证明了新提出的算法相较于另外4种算法具有更优的适应度值、稳定性和收敛速度。

关键词: 分布式时延; 进化因子; 权重; 粒子群优化

An improved Particle Swarm Optimization algorithm

HU Jianhua, XIONG Weili

(College of Science, University of Shanghai for Science and Technology, Shanghai 200093, China)

【Abstract】 Particle Swarm Optimization algorithm (PSO) is a kind of evolutionary calculation method with swarm intelligence. In the search process, all particles closely follow the optimal particle's movement, which reduces the particles' diversity and global search ability. So it is easy to fall into local optima. In this paper, a new swarm optimization algorithm (PSO-EWD) has been proposed which is mainly improved in two aspects; the inertia weight is associated with the evolution factor, and the weight is changed according to the evolution state of the population to balance the global search ability and the local search ability; the distributed time-varying delays are introduced into the velocity update formula to increase diversity of the particles. In this paper, the experimental comparison of five algorithms on nine benchmark functions shows that the proposed algorithm has better fitness value, stability and convergence speed than the other four algorithms.

【Key words】 distributed time-delay; evolutionary factor; weight; Particle Swarm Optimization (PSO)

0 引言

粒子群优化算法(PSO)^[1]是一种种群随机搜索算法,其灵感来源于鸟类的群集行为。由于PSO算法原理简单、易实现的特点,在众多领域中有着广泛的应用。但PSO算法也有收敛速度慢和容易陷入局部最优等不足。一方面是由于PSO算法对其控制参数相当敏感,合理的参数配置才能提高算法的性能,PSO-LDIW(Shi和Eberhart,1998年、1999年)^[2-4]、PSO-CK(Clerc和Kennedy,2002年)^[5]、PSO-TVAC(Ratnaweera,2004年)^[6]等算法应运而生。另一方面,所有粒子紧跟最优粒子运动降低了粒子多样性和全局搜索的能力。经典的PSO算法仅关注于粒子的当前速度、当前位置、个体最优位置和全局最优位置,忽略了粒子的历史信息 and 种群的分布状态等因素。2007年,Zhan等人^[7]用聚类分析方法,分析了搜索过程中种群分布特性,提出种群进化状态这一概念。2009年,Zhan等人^[8]提出了一种自适应PSO算法(APSO),该算法根据种群实时进

化状态来实现惯性权重的自动控制,用以提高搜索效率和收敛速度。进一步考虑到历史信息对粒子当前速度的影响,时延这一概念被引入PSO算法中,SPSO(Tang等人,2011年)^[9]、SDPSO(Zeng等人,2016年)^[10]、MDPSO(Song等人,2017年)^[11]等变体相继被提出来。这些算法有效地平衡了局部搜索和全局搜索能力,提高了粒子的多样性。2019年,Liu等人^[12]引入随机分布式时延,提出RODDPSO算法,该算法充分考虑了历史个体最优和全局最优信息,但却忽略了不同阶段的历史信息对当前状态的影响是不同的。

在此基础上,本文综合考虑了种群的进化状态和不同阶段时延的影响效果,提出了一种改进的PSO算法(PSO-EWD)。该次研究的创新点在于:将惯性权重与进化因子相关联,根据种群的进化状态而改变权重大小,使全局搜索能力与局部搜索能力得到平衡;将时变的分布式时延引入速度更新公式中,以增加粒子的多样性。

作者简介: 胡建华(1978-),女,博士,讲师,主要研究方向:大数据;熊伟利(1994-),女,硕士研究生,主要研究方向:大数据。

收稿日期: 2021-03-08

1 PSO 算法

假设 S 为种群粒子数, $I = \{1, 2, 3, \dots, S\}$, D 代表搜索空间的维数, 则第 i 个粒子的速度用 V_i 表示, 位置用 X_i 表示, 其中 $X_i, V_i \in R^D$, $i \in I$ 。设第 k 次迭代后第 i 个粒子的速度为 $V_i^{(k)} = (V_{i1}^{(k)}, V_{i2}^{(k)}, \dots, V_{iD}^{(k)})$ 位置为 $X_i^{(k)} = (X_{i1}^{(k)}, X_{i2}^{(k)}, \dots, X_{iD}^{(k)})$ 。原始的 PSO 算法在寻找最优解的过程中所有粒子都紧跟个体最优粒子和全局最优粒子, 向着全局最优位置移动, 记第 i 个粒子的最优位置为 $P_i = (P_{i1}, P_{i2}, \dots, P_{iD})$, 全局最优粒子的位置为 $P_G = (P_{G1}, P_{G2}, \dots, P_{GD})$ 。更新迭代公式是:

$$\begin{cases} V_i^{(k+1)} = \omega V_i^{(k)} + c_1 r_1 (P_i^{(k)} - X_i^{(k)}) + c_2 r_2 (P_G^{(k)} - X_i^{(k)}), \\ X_i^{(k+1)} = X_i^{(k)} + V_i^{(k+1)} \end{cases} \quad (1)$$

其中, k 是当前迭代次数; $r_i (i = 1, 2)$ 是 D 维向量, 向量中每一个分量都是 $[0, 1]$ 上的随机数; ω 为惯性权重; c_1 是个体认知加速度系数; c_2 是社会加速度系数。

Shi 和 Eberhart 提出 PSO-LDIW 算法^[2-4], 就采用了线性下降的惯性权重, 即:

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times \frac{\text{iter}}{\max \text{ iter}} \quad (2)$$

其中, $\omega_{\max} (\omega_{\min})$ 表示在寻优过程中惯性权重的最大(最小)值; $\text{iter} (\max \text{ iter})$ 表示当前(最大)的迭代次数。

2002 年, Clerc 和 Kennedy 指出当 $\omega = 0.729$, $c_1 = c_2 = 1.49$ 时算法效果较好^[13] (PSO-CK 算法)。2004 年, 受时变惯性权重的启发, Ratnaweera 等人提出了 PSO-TVAC 算法^[14], 将加速度因子改进为:

$$c_1 = (c_{1i} - c_{1f}) \times \frac{\max \text{ iter} - \text{iter}}{\max \text{ iter}} + c_{1f} \quad (3)$$

$$c_2 = (c_{2i} - c_{2f}) \times \frac{\max \text{ iter} - \text{iter}}{\max \text{ iter}} + c_{2f} \quad (4)$$

其中, $c_{1i} (c_{1f})$ 和 $c_{2i} (c_{2f})$ 分别是个体认知加速度系数和社会加速度系数的初值(终值), 这些参数的取值为: $c_{1i} = 2.5$, $c_{1f} = 0.5$, $c_{2i} = 0.5$, $c_{2f} = 2.5$ 。

2 改进的 PSO 算法

2.1 进化状态的判断

2009 年, Zhan 等人^[8]通过分析种群的搜索行为

和分布特性提出了进化状态, 整个搜索过程种群表现出 4 种状态: 收敛状态、开发状态、勘探状态和跳出状态, 分别用 $\xi = 1, 2, 3, 4$ 表示。用 d_i 表示第 i 个粒子与其他所有粒子的平均距离, 即:

$$d_i = \frac{1}{S-1} \sum_{j=1, j \neq i}^S \sqrt{\sum_{k=1}^D (x_{ik} - x_{jk})^2} \quad (5)$$

记 d_{\min} , d_{\max} 是集合 $\{d_i | i \in I\}$ 中的最小值和最大值, 用 G 表示全局最优粒子, 显然 $d_{\min} \leq d_G \leq d_{\max}$ 。令:

$$E_f = \frac{d_G - d_{\min}}{d_{\max} - d_{\min}} \quad (6)$$

称为种群的进化因子, 式(6)表明进化因子能恰当刻画种群的分布状态, 根据进化因子 E_f 的大小不同而取不同状态^[9]:

$$\xi = \begin{cases} 1, & 0 \leq E_f < 0.25 \\ 2, & 0.25 < E_f \leq 0.50 \\ 3, & 0.50 \leq E_f < 0.75 \\ 4, & 0.75 \leq E_f \leq 1 \end{cases} \quad (7)$$

2.2 新算法的构建

经典的 PSO 算法开始搜索时速度很快, 但在搜索过程中, 所有的粒子都向着当前最优粒子的方向寻找, 这使粒子失去了多样性, 在搜索后期收敛速度明显下降, 并且容易陷入局部最优。本文通过考虑不同阶段的历史信息对现在的影响, 引入具有时变性的分布式时延, 以增加粒子的多样性; 同时将惯性权重与进化因子相关联, 来平衡算法的全局搜索和局部搜索的能力。改进后的 PSO-EWD 算法的迭代公式为:

$$\begin{aligned} V_i^{(k+1)} &= \omega(E_f) V_i^{(k)} + c_1 r_1 (P_i^{(k)} - X_i^{(k)}) + \\ & c_2 r_2 (P_G^{(k)} - X_i^{(k)}) + \\ & m_i(\xi^{(k)}) c_3 r_3 \sum_{\tau=1}^N \omega_1 \alpha_{(\tau)} (P_i^{(k-\tau)} - X_i^{(k)}) + \\ & m_G(\xi^{(k)}) c_4 r_4 \sum_{\tau=1}^N \omega_1 \alpha_{(\tau)} (P_G^{(k-\tau)} - X_i^{(k)}) \\ X_i^{(k+1)} &= X_i^{(k)} + V_i^{(k+1)} \end{aligned} \quad (8)$$

其中, $\omega(E_f)$ 是和进化因子 E_f 相关的惯性权重, r_3, r_4 是如同 r_1, r_2 的 D 维随机向量; $\alpha_{(\tau)}$ 为随机数 0 或者 1; ω_1 是时延发生时的自适应权重, 决定了每个时延影响的大小; $\sum_{\tau=1}^N \omega_1 \alpha_{(\tau)} (P_i^{(k-\tau)} - X_i^{(k)})$, $\sum_{\tau=1}^N \omega_1 \alpha_{(\tau)} (P_G^{(k-\tau)} - X_i^{(k)})$ 分别是关于自我认知和社会的分布式时延项, 这里的 τ 是时延步数, N 为分布式时延步数最大值; 约定当 $\tau \geq k$ 时, $P_i^{(k-\tau)} = P_i^{(k)}$, $P_G^{(k-\tau)} = P_G^{(k)}$; $\xi^{(k)}$ 是第 k 次迭代时

种群的当前的状态; c_3 和 c_4 是分布式时延项的加速度因子; $m_l(\xi^{(k)})$ 和 $m_c(\xi^{(k)})$ 是分布式时延项的强度因子, 两者都是根据进化状态 $\xi^{(k)}$ 所确定的。

2.3 参数

惯性权重 ω 用来平衡算法的全局搜索能力和局部搜索能力, PSO-EWD 算法将 ω 与进化因子 E_f 相联系, 以适应于搜索环境。因为相对较小的 ω 有利于种群收敛和开发, 相对较大的 ω 有利于种群勘探和跳出, 而进化因子在跳出状态时相对较大而收敛状态时相对较小, 本文取 $\omega(E_f)$ 的初值为 0.9, 计算公式为:

$$\omega(E_f) = 0.5E_f + 0.4 \quad (9)$$

权重 ω_1 用来控制时延项对速度的影响, 因为离当前状态越近的历史信息对当前状态的影响较大, 而越远的历史信息对当前状态的影响相对较小, 因此 ω_1 为关于 τ 的递减函数, 本文取:

$$\omega_1 = \frac{N - \tau}{N} \quad (10)$$

取分布式时延项的加速度因子 $c_3 = c_1$, $c_4 = c_2$; 强度因子 $m_l(\xi)$ 、 $m_c(\xi)$ 根据进化状态 ξ 来确定(见表 1), 其初始值 $m_l(\xi)$ 、 $m_c(\xi)$ 都取为 0, 在 k 次迭代后, 若种群的进化状态为收敛时, 粒子将紧跟当前找到的全局最优粒子快速聚集, 忽略时延项的影响而取 $m_l(1) = m_c(1) = 0$; 在开发状态时, 粒子将利用个体最优历史信息在潜在区域仔细搜索, 而忽略全局最优信息的影响, 取 $m_l(2) = -0.01$, $m_c(2) = 0$ 。在勘探状态时, 探索全局最优解是重要任务, 鼓励粒子在全局历史最优信息的指导下探索整个搜索空间, 取 $m_l(3) = 0$, $m_c(3) = 0.01$; 在跳出状态时, 粒子群将跟随全局最优粒子飞离局部极值周围区域, 去寻求一个更好的解, 个体和全局的历史最优位置都需要综合考虑, 取 $m_l(4) = 0.01$, $m_c(4) = 0.01$ 。

表 1 不同进化状态下强度因子选择策略

Tab. 1 Intensity factors selection strategy under different states

状态	ξ	$m_l(\xi)$	$m_c(\xi)$
收敛	1	0	0
开发	2	0.01	0
勘探	3	0	0.01
跳出	4	0.01	0.01

3 仿真实验

3.1 基准函数

本文选取 9 个常见的基准函数来验证算法的性

能, 其中包括单峰函数、多峰函数。研究中选择的函数、名字、搜索空间等具体信息见表 2。

表 2 基准函数

Tab. 2 Benchmark function

函数	名字	搜索空间
$f_1(x)$	Penalized 1	[-50, 50]
$f_2(x)$	Schwefel 1.2	[-100, 100]
$f_3(x)$	Schwefel 2.21	[-100, 100]
$f_4(x)$	Schwefel 2.22	[-10, 10]
$f_5(x)$	Ackley	[-32, 32]
$f_6(x)$	Sphere	[-100, 100]
$f_7(x)$	Rosenbrock	[-30, 30]
$f_8(x)$	Griewank	[-600, 600]
$f_9(x)$	Step	[-100, 100]

3.2 参数 N 的训练

在搜索空间随机选取 20 个种群, 所有粒子均具有随机的初始速度 $V_i(i \in I)$ 和位置 $X_i(i \in I)$, 计算出每个粒子的适应度值, 选出初始个体最优粒子位置 P_i 和全局最优粒子位置 P_g 。为验证 PSO-EWD 算法在处理复杂问题时的优越性, 本文选定搜索空间的维数为 100 维, 最大迭代次数为 20 000 次, 同时为消除随机因素的影响, 每个实验重复 40 次, 最后取平均值。

在 PSO-EWD 算法的速度更新公式中, 分布式时延步数的最大值 N 是个训练参数, 由实验训练所确定。过大的 N 会增加计算负担, 过小的 N 不能充分发挥时延的作用, 本文将在 75、100、125、150、175 五个数中选取一个使 PSO-EWD 算法性能较好的 N 。实验结果如图 1 所示, 纵坐标表示平均适应度值的对数, 横坐标表示迭代次数。

由图 1 可看出, 在 5 个不同 N 的取值中, 当 $N = 125$ 时函数 $f_2(x)$ 、 $f_3(x)$ 、 $f_4(x)$ 、 $f_6(x)$ 以及 $f_7(x)$ 有最优的适应度值和相对快的收敛速度; 虽然函数 $f_1(x)$ 、 $f_5(x)$ 、 $f_8(x)$ 、 $f_9(x)$ 没有最优适应度值, 但有更早的收敛趋势。因此本文选取最大时延步数 $N = 125$ 。

3.3 5 种算法的对比

本文选取 PSO-CK、PSO-LDIW、MDPSO、RODDPSO 四种算法来对比验证 PSO-EWD 算法的优越性。仿真实验结果如图 2 所示。图 2 表现了 5 种算法在 100 维搜索空间中的收敛性能, 对于 9 个

基准函数来说, PSO-EWD 算法以更快速度收敛于最优的适应度值。5 种算法在 100D 的搜索空间中的性能比较见表 3。由表 3 可知, 从最小值、均值、方差三个评价指标来对比 5 种算法。从表 3 中可以看出, PSO-EWD 算法在 9 个基准函数上都有最小的适应度均值, 这说明 PSO-EWD 算法有较好的寻优质量和收敛精度。从方差来看, PSO-EWD 算法仅在 $f_5(x)$ 上次于 RODDPSO 算法, 这说明 PSO-EWD 算法具有较好的稳定性。就最小值而言, PSO-EWD 算法仅在 $f_6(x)$ $f_7(x)$ 上次于 RODDPSO

算法, 这说明 PSO-EWD 有较好的跳出局部极值、寻找全局最优解的能力。进一步仔细比较, 算法性能表现次优的是 RODDPSO 算法, 这说明引入了分布式时延的 2 种算法由于充分考虑了历史个体最优信息和全局最优信息而更能增加粒子的多样性, 增强跳出局部最优的能力。而采用与进化状态关联的惯性权重和具有时变性的时延的 PSO-EWD 算法更能平衡全局搜索能力和局部搜索能力, 从而提升了收敛精度。

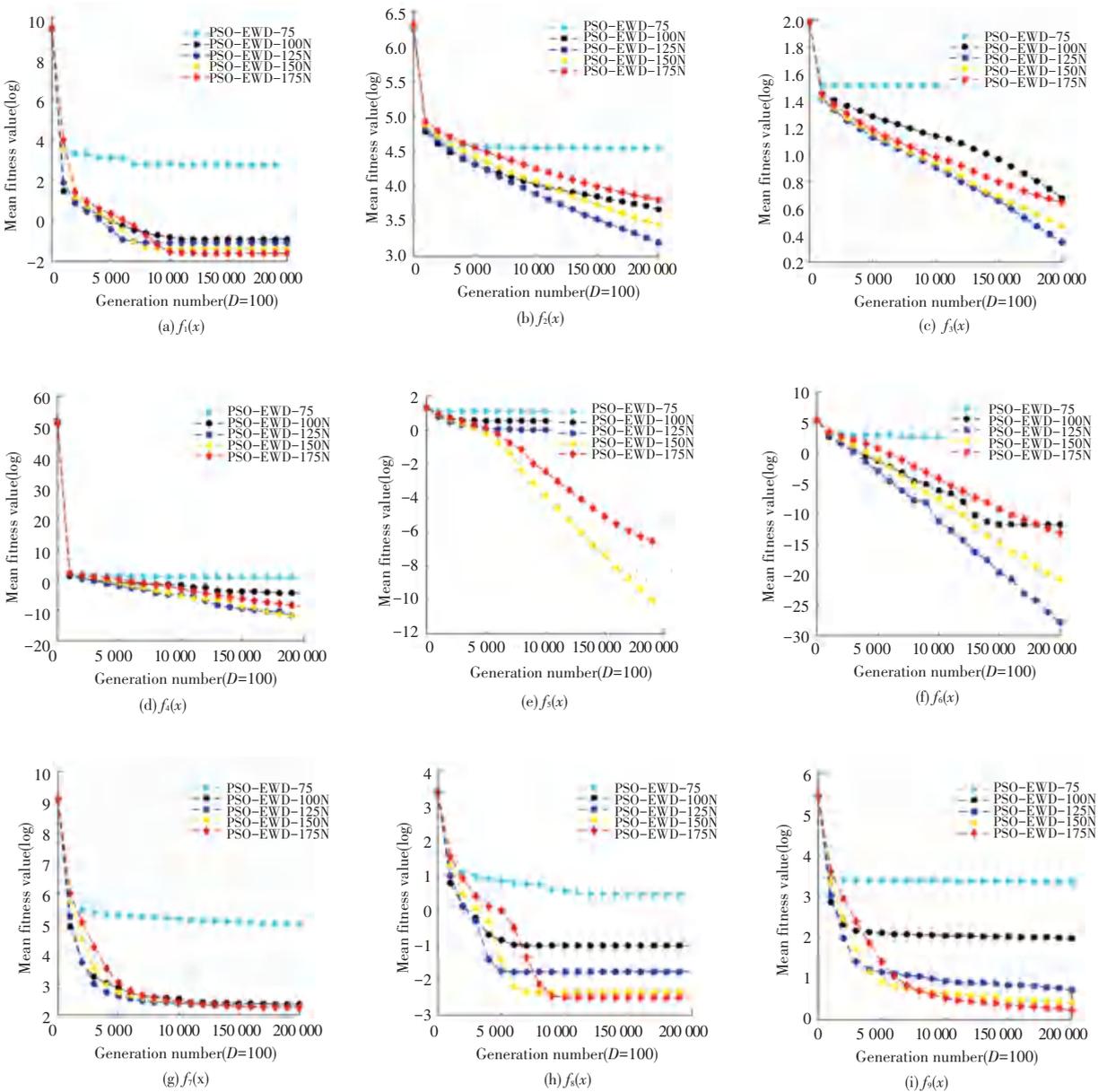


图 1 由 PSO-EWD 算法训练 N

Fig. 1 Training N by PSO-EWD algorithm

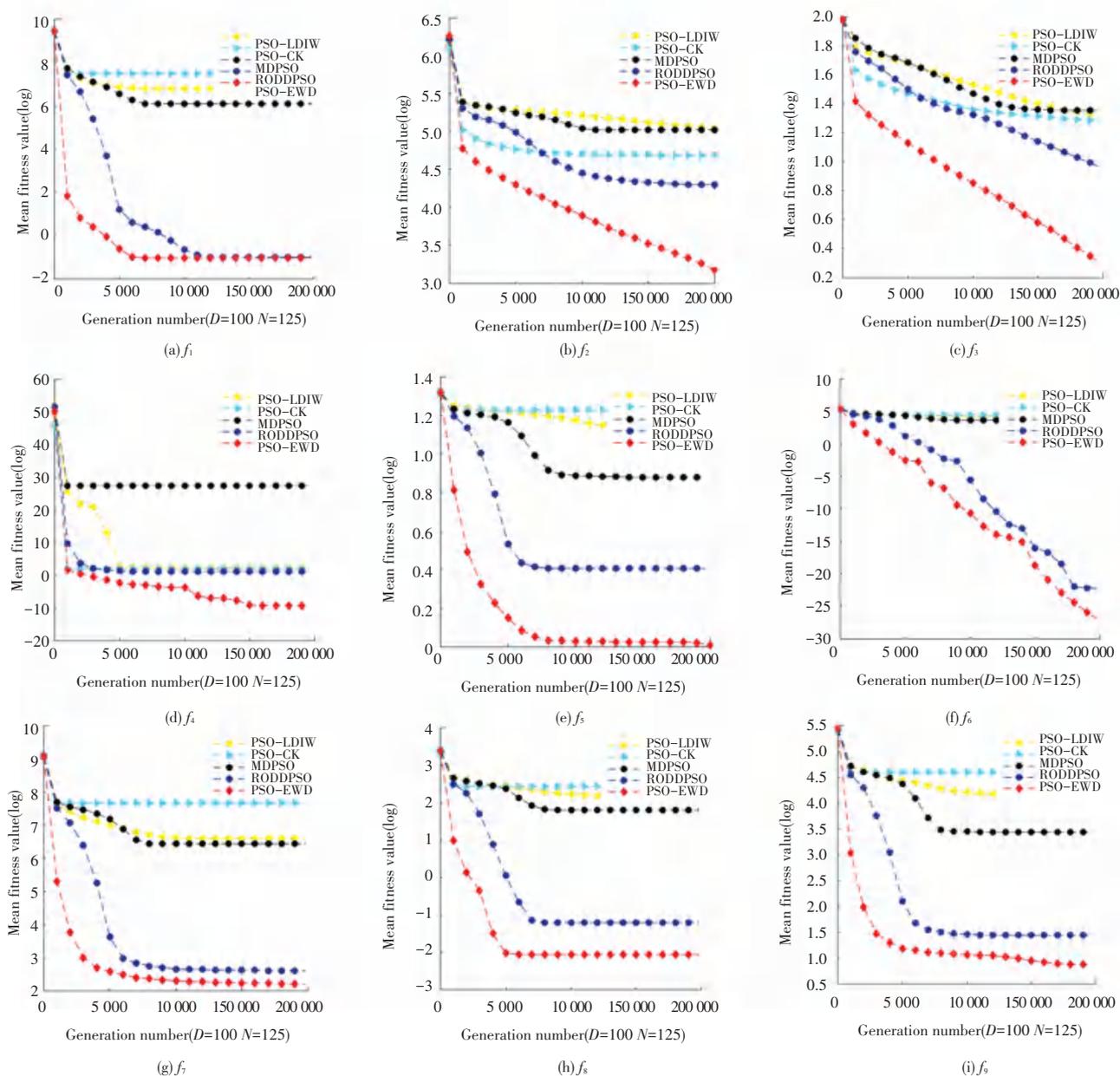


图 2 各种算法在 100D 的搜索空间中的对比图

Fig. 2 Comparison of algorithms in a 100D search space

4 结束语

本文考虑到所有粒子紧跟最优粒子运动降低了粒子多样性和全局搜索能力,提出了一种新的分布式权重粒子群优化算法(PSO-EWD)。通过考虑不同阶段的历史信息对当前速度的影响,引入具有时变性的分布式时延,以增加粒子的多样性;同时,将

惯性权重与进化因子相关联,来平衡算法的全局搜索和局部搜索的能力。实验在 100 维的搜索空间中迭代 20 000 次,并且为了减少随机因素的影响实验重复 40 次而取平均值。新算法在 9 个基准函数上与 4 种经典的 PSO 算法对比,实验结果证明 PSO-EWD 算法具有更优的稳定性、收敛速度与适应度值。

表 3 5 种算法在 100D 的搜索空间中的性能比较

Tab. 3 Comparison of performance of five algorithms in 100D search space

函数	评价指标	PSO-LDIW	PSO-CK	MDPSO	RODDPSO	PSO-EWD
$f_1(x)$	最小值	2.31×10^{-14}	5.83×10^{-10}	1.85×10^{-1}	5.28×10^{-28}	1.45×10^{-30}
	均值	8.34×10^{-2}	3.20×10^7	1.26×10^6	1.01×10^{-1}	6.08×10^{-2}
	方差	1.58×10^{-1}	1.04×10^8	5.41×10^6	1.81×10^{-1}	1.36×10^{-1}
$f_2(x)$	最小值	4.21×10^4	5.02×10^3	1.99×10^4	1.52×10^3	5.65×10^2
	均值	1.23×10^5	4.23×10^4	1.06×10^5	1.99×10^4	1.51×10^3
	方差	4.11×10^4	2.98×10^4	1.06×10^5	1.14×10^4	5.94×10^2
$f_3(x)$	最小值	1.70×10^1	1.48×10^1	1.34×10^1	6.72×10^0	1.22×10^0
	均值	2.13×10^1	1.93×10^1	2.28×10^1	9.02×10^0	1.89×10^0
	方差	2.01×10^0	2.52×10^0	9.81×10^0	1.66×10^0	4.66×10^{-1}
$f_4(x)$	最小值	9.00×10^1	3.22×10^1	5.59×10^0	2.80×10^{-5}	1.44×10^{-18}
	均值	1.76×10^2	9.02×10^1	3.71×10^{27}	1.01×10^1	3.32×10^{-14}
	方差	4.12×10^1	3.32×10^1	2.35×10^{28}	1.04×10^1	1.04×10^{-13}
$f_5(x)$	最小值	2.29×10^{-7}	1.36×10^1	2.43×10^0	1.42×10^0	1.48×10^{-13}
	均值	1.44×10^1	1.73×10^1	7.60×10^0	2.55×10^0	1.11×10^0
	方差	3.73×10^0	1.33×10^0	4.02×10^0	4.29×10^{-1}	8.60×10^{-1}
$f_6(x)$	最小值	1.33×10^{-13}	1.47×10^{-11}	4.98×10^{-1}	7.07×10^{-35}	4.40×10^{-32}
	均值	2.10×10^4	2.83×10^4	4.26×10^3	5.70×10^{-23}	7.22×10^{-28}
	方差	1.17×10^4	1.50×10^4	1.21×10^4	3.58×10^{-22}	3.47×10^{-27}
$f_7(x)$	最小值	1.11×10^2	7.67×10^1	4.99×10^2	1.07×10^1	7.81×10^1
	均值	4.02×10^6	3.40×10^7	2.79×10^6	3.86×10^2	1.75×10^2
	方差	1.17×10^7	5.09×10^7	9.40×10^6	8.06×10^2	4.98×10^1
$f_8(x)$	最小值	1.00×10^{-13}	4.72×10^{-1}	1.96×10^{-1}	7.22×10^{-15}	0
	均值	1.76×10^2	2.91×10^2	6.32×10^1	6.04×10^{-2}	1.77×10^{-2}
	方差	1.19×10^2	1.31×10^2	1.25×10^2	7.52×10^{-2}	4.19×10^{-2}
$f_9(x)$	最小值	0	4.47×10^3	9.20×10^1	2.00×10^0	0
	均值	1.75×10^4	3.41×10^4	2.76×10^3	2.70×10^1	8.43×10^0
	方差	1.33×10^4	1.38×10^4	7.88×10^3	2.73×10^1	1.08×10^1

参考文献

[1] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory [C] //Proceedings of the 6th International Symposium on Micro Machine and Human Science. Nagoya, Japan;IEEE, 1995: 39-43.

[2] SHI Y, EBERHART R C. Parameter selection in particle swarm optimization [C]//Proceedings of the 7th International Conference on Evolutionary Programming. San Diego, CA, USA;IEEE, 1998: 591-600.

[3] SHI Y, EBERHART R C. A modified particle swarm optimizer [C] //Proceedings of IEEE International Conference on Evolutionary Computation. Anchorage, AK;IEEE, 1998: 69-73.

[4] SHI Yuhui, EBERHART R C. Empirical study of particle swarm optimization [C]// Proceedings of the 1999 Congress on Evolutionary Computation. Washington DC, USA; IEEE, 1999, 3:1945-1950.

[5] 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.

[6] RATNAWEERA A, HALAAMUGE S, WATSON H C. Self organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients [J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 240-255.

[7] ZHAN Zhihui, XIAO Jing, ZHANG Jun, et al. Adaptive control of acceleration coefficients for particle swarm optimization based on clustering analysis [C] //2007 IEEE Congress on Evolutionary Computation. Singapore;IEEE, 2007:3276-3282.

[8] ZHAN Zhihui, ZHANG Jun, LI Yun, et al. Adaptive particle swarm optimization [J]. IEEE Transactions on systems, Man, and Cybernetics, Part B (Cybernetics), 2009, 39 (6): 1362-1381.

[9] TANG Y, WANG Z, FANG J. Parameters identification of unknow delayed genetic regulatory networks by a switching

- particle swarm optimization algorithm [J]. *Expert Systems with Applications*, 2011, 38(3): 2523–2535.
- [10] ZENG Nianyin, WANG Zidong, ZHANG Hong, et al. A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay [J]. *Cognitive Computation*, 2016, 8(2): 143–152.
- [11] SONG Baoye, WANG Zidong, ZOU Lei. On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm [J]. *Cognitive Computation*, 2017, 9(1): 5–17.
- [12] LIU Weibo, WANG Zidong, LIU Xiaohui. A novel particle swarm optimization approach for patient clustering from emergency departments [J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(4): 632–644.
- [13] CLERK 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.
- [14] RATNAWEERA A, HALAAMUGE S, WATSON HC. Self organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients [J]. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3): 240–255.

(上接第5页)

脸是正向清晰的,能够确保人脸识别的准确率和系统的效率,同时本文的人脸识别精度优于 Dlib 人脸识别库。通过对多个人脸数据集的实验证明了本文人脸识别算法的良好性能,不过对于佩戴口罩等存在遮挡的人脸场景的识别依然不够精准,这仍是一个具有挑战的方向,下一步还应深入研究,寻找更有效的方法。

参考文献

- [1] 十大民生热词 2017: 校园安全排名第一[N]. 人民日报, 2017-12-27(002).
- [2] 潘玉娇. 推动新时代校园安全工作再上新台阶[N]. 中国教育报, 2019-05-18(012).
- [3] RAVEENDRA K, RAVI J. Performance evaluation of face recognition system by concatenation of spatial and transformation domain features, *International Journal of Computer Network and Information Security*, 2021, 13(1): 47–60.
- [4] REDMON J, FARHADI A. YOLOv3: An incremental improvement [J]. *arXiv preprint arXiv:1804.02767*, 2018.
- [5] SZEGEDY C, TOSHEV A, ERHAN D. Deep Neural Networks for object detection [J]. *Advances in neural information processing systems*, 2013, 26: 2553–2561.
- [6] ARTHUR D, VASSILVITSKII S. K-Means++: The advantages of careful seeding [C]// *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007*. New Orleans, Louisiana, USA: ACM, 2007: 1027–1035.
- [7] BABU B G, MURTY M. A near optimal initial seed value selection in kmeans algorithm using a genetic algorithm [J]. *Pattern Recognition Letters*, 1993, 14(10): 763–769.
- [8] SZEGEDY C, VANHOUCHE V, IOFFE S, et al. Rethinking the inception architecture for computer vision [C]// *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE Computer Society, 2016: 2818–2826.
- [9] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition [C]// *2016 IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, NV, USA: IEEE, 2016: 770–778.
- [10] 颜伟超. 基于 LIBSVM 的简单图识别研究与实现 [J]. *电子技术*, 2020, 49(1): 44–46.
- [11] CHEN Tian, JU Sihang, REN Fuji, et al. EEG emotion recognition model based on the LIBSVM classifier [J]. *Measurement*, 2020, 164: 108047.
- [12] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]// *IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, USA: IEEE Computer Society, 2014: 580–587.
- [13] KRIZHEVSKY A, SUTSKEVER I, HINTON G. ImageNet classification with Deep Convolutional Neural Networks [C]// *NIPS*. Nevada: Curran Associates Inc., 2012, 141: 1097–1105.
- [14] GIRSHICK R. Fast R-CNN [C]// *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago: IEEE, 2015: 1440–1448.
- [15] LIN T Y, DOLLÁR P, GIRSHICK R, et al. Feature pyramid networks for object detection [C]// *Proceedings of the IEEE conference on computer vision and pattern recognition*. Honolulu, HI, USA: IEEE, 2017: 2117–2125.
- [16] MILAN A, LEALTAIXE L, REID I, et al. MOT16: A benchmark for multi-object tracking [EB/OL]. (2016-03-02). <https://arxiv.org/pdf/1603.00831.pdf>.
- [17] SUN Tao, ZHU Xingjie, PAN Shyang, et al. No-reference image quality assessment in spatial domain [M]// *SUN H, YANG C Y, LIN C, et al. Genetic and Evolutionary Computing. Advances in Intelligent Systems and Computing*. Cham: Springer, 2015, 329: 381–388.
- [18] LASMAR N E, STITOU Y, BERTHOUMIEU Y. Multiscale skewed heavy tailed model for texture analysis [C]// *IEEE International Conference on Image Processing*. Cairo, Egypt: IEEE, 2010: 2281–2284.
- [19] 尹旭东. 基于并行 SVM 算法的中文文本分类方法研究 [D]. 长春: 吉林大学, 2018.
- [20] LUKIN V V. Estimation of parameters for generalized Gaussian distribution [C]// *2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP)*. Athens, Greece: IEEE, 2014: 376–379.
- [21] TRIGUEROS D S, MENG L, HARTNETT M. Face recognition: From traditional to deep learning methods [J]. *arXiv preprint arXiv:1811.00116*, 2018.
- [22] SZEGEDY C, IOFFE S, VANHOUCHE V, et al. Inception-v4, Inception-ResNet and the impact of residual connections on learning [J]. *arXiv preprint arXiv:1602.07261*, 2016.
- [23] SENGUPTA S, CHEN J C, CASTILLO C, et al. Frontal to profile face verification in the wild [C]// *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Lake Placid, NY, USA: IEEE, 2016: 1–9.
- [24] 李洋, 郭红利, 管树林, 等. OpenCV+Dlib 技术在课堂教学中的应用与研究 [J]. *电脑知识与技术*, 2020, 16(19): 159–162.