

一种带分布式自适应时延的粒子群算法

胡建华, 熊伟利

上海理工大学理学院, 上海

Email: smilydragon2011@163.com, 1296612320@qq.com

收稿日期: 2021年2月23日; 录用日期: 2021年3月19日; 发布日期: 2021年3月26日

摘要

针对粒子群算法(PSO)容易陷入局部最优值、收敛精度低等缺陷, 提出一种新的带分布式自适应时延的粒子群算法(PSO-DW)。改进的算法主要在RODDPSO算法的基础上考虑时延的时变性和种群的进化状态, 以平衡算法的全局搜索和局部搜索能力, 降低早熟收敛的可能性, 提高算法的收敛速度和精度。主要思想: 1) 在引入了分布式时延的速度更新公式中, 每个时延项配以自适应权重, 2) 引入通过当前状态和概率转移矩阵预测下一进化状态的预测机制, 3) 分布式时延的强度因子由预测状态所确定。在九个基准函数上与四个算法作对比的实验结果表明, 改进后的算法在寻优质量、稳定性、收敛速度等方面更具优越性。

关键词

粒子群优化(PSO), 分布式时延, 进化因子, 权重

A Particle Swarm Optimization Algorithm with Distributed Adaptively Weighted Delays

Jianhua Hu, Weili Xiong

University of Shanghai for Science and Technology, Shanghai

Email: smilydragon2011@163.com, 1296612320@qq.com

Received: Feb. 23rd, 2021; accepted: Mar. 19th, 2021; published: Mar. 26th, 2021

Abstract

A new particle swarm optimization algorithm (PSO) with distributed adaptively weighted delays (PSO-DW) has been proposed to overcome the defects of the PSO algorithm, such as falling into local optimal value, low convergence accuracy. Based on the RODDPSO algorithm, the improved algorithm

further considers the time-varying delays and the evolutionary states of the population, so that it can balance the global search and local search ability of the algorithm, reduce the possibility of premature convergence, and improve the convergence speed and accuracy of the algorithm. The main ideas are: 1) each delay is equipped with adaptive weight in the velocity update formula; 2) prediction mechanism of the next evolutionary state has been introduced by the current state and probability transfer matrix; 3) intensity factor of the distributed delay is determined by the prediction state. The experimental results show that the improved algorithm has more advantages in optimizing quality, stability and convergence speed by comparing with four algorithms on nine benchmark functions.

Keywords

Particle Swarm Optimization (PSO), Distributed Time-Delay, Evolutionary Factor, Weight

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

粒子群算法(particle swarm optimization, PSO)是一种基于群体智能的随机全局优化算法[1], 最初由 Eberhart 和 Kennedy (1995) [2]受鸟群捕食行为启发而提出, 其基本思想是通过群体中个体之间的协作和信息共享来寻找最优解。由于算法简单、易于实现、快速收敛等特性, PSO 算法自从提出之日起就受到很多学者的青睐。近些年来 PSO 算法得到长足的发展并被广泛应用到实际领域的优化问题中。但粒子群算法依旧存在容易陷入局部最优从而导致早熟收敛的缺陷, 这种缺陷随搜索空间维数的增大愈发明显。为了增加粒子的多样性以减少陷入局部收敛的可能性, 学者们提出了各种改进策略来提高 PSO 算法的优化性能。基于参数调整的 PSO 算法有 PSO-LDIW (Shi, 1998, 1999) [3] [4] [5]、PSO-CK (Clerc 和 Kennedy, 2002) [6]、PSO-TVAC (Ratnaweera, 2004) [7]、NCPSO (2017) [8]、AWPSO (Liu and Wang, 2019) [9]等。基于拓扑结构和学习策略的改进算法有 APSO (Zhan 等人, 2009) [10]、SPSO (Tang 等人, 2011) [11]、SDPSO (Zeng 等人, 2016) [12]、MDPSO (Song 等人, 2017, 2018) [13] [14]等。这些算法通过考虑粒子群的四种进化状态和时延影响等因素大大提高了 PSO 算法的性能。2019 年, Liu 等人[9]提出了一种基于随机分布式时延的 PSO 算法(RODDPSO), 大大提高了算法的搜索能力, 但是这种算法忽略了不同阶段的历史信息对当前状态的影响是不同的。事实上, 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。

本文拟在 RODDPSO 算法的基础上, 充分考虑不同历史信息对当前状态的影响, 提出一种带分布式自适应时延的粒子群算法(PSO-DW)。改进算法中时延是具有时变性的, 分布式时延的强度因子由预测状态所确定。新算法能进一步平衡全局搜索能力和局部搜索能力, 降低早熟收敛的可能, 提高算法的收敛速度和精度。仿真实验在 150 维搜索空间的九个基准函数上与四个算法作对比。结果表明, 改进后的算法在寻优质量、稳定性、收敛速度等方面更具优越性。

2. 传统 PSO 算法

在 PSO 算法中, 所有的粒子构成种群, 并且每一个粒子都可能是所研究优化问题的最优解。粒子在搜索空间中根据个体的最优位置和群体最优位置更新速度和位置, 从而寻找最优解[15]。设种群由 S 个粒子组成, 记集合 $I = \{1, 2, 3, \dots, S\}$ 。设搜索空间为 R^D , 其中 D 为维数。用 $X_i \in R^D, i \in I$ 表示第 i 个粒子

的位置, 其速度为 $V_i \in R^D$ 。第 i 个粒子第 k 次迭代后的速度和位置分别是:

$$\begin{aligned} V_i^{(k)} &= (V_{i1}^{(k)}, V_{i2}^{(k)}, \dots, V_{iD}^{(k)}), \\ X_i^{(k)} &= (X_{i1}^{(k)}, X_{i2}^{(k)}, \dots, X_{iD}^{(k)}). \end{aligned}$$

令 $P_i = (P_{i1}, P_{i2}, \dots, P_{iD})$ 表示第 i 个粒子的最优位置, $P_G = (P_{G1}, P_{G2}, \dots, P_{GD})$ 表示种群全局最优位置[16]。在搜索的过程中, 所有的粒子都朝着全局最优的方向移动。第 i 个粒子在 $k+1$ 时刻的速度和位置的更新公式为:

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

在公式(1)中, ω 是惯性权重; r_1, r_2 都是 D 维向量, 其每个分量都是 $[0, 1]$ 上的随机数; c_1, c_2 分别表示个体认知加速度系数和社会认知加速度系数。

为了提高搜索速度, Clerc 和 Kennedy (2002)提出了 PSO-CK 算法[6]; 在此算法中通过大量的实验证明当 $c_1 = c_2 = 1.49, \omega = 0.729$ 时性能最优。为了平衡当前速度、认知和社会等方面的性能, Shi 和 Eberhart 博士先后在 1998 年和 1999 年提出了 PSO-LDIW 算法[3] [4] [5], 该算法采用了线性下降的惯性权重:

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

这里 $\omega_{\max}$ ($\omega_{\min}$) 为在整个搜索过程中惯性权重的最大(最小)的值; $iter$ ($maxiter$) 表示当前(最大)的迭代次数。受时变惯性权重的启发, Ratnaweera 等人在 2004 年提出了 PSO-TVAC 算法[7], 将加速度因子改进为:

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

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

上式中 c_{1i} (c_{1f}) 和 c_{2i} (c_{2f}) 分别是个体认知加速度系数和社会认知加速度系数的初值(终值), 这些参数的取值为:

$$c_{1i} = 2.5, c_{1f} = 0.5, c_{2i} = 0.5, c_{2f} = 2.5.$$

3. 改进的粒子群算法

本文提出一个带有分布式自适应时延和与由预测状态所确定的强度因子的粒子群优化(PSO-DW)算法。

3.1. 进化状态的刻画与预测

进化状态这一概念先后由 Zhan [10]、Tang [11] 等人根据种群分布信息提出。整个搜索过程包含四种状态: 收敛、开发、勘探和跳出, 分别用 $\xi = 1, 2, 3, 4$ 表示。记

$$d_i = \frac{1}{S-1} \sum_{j=1, i \neq j}^S \sqrt{\sum_{l=1}^D (x_{il} - x_{jl})^2}, i \in I \quad (5)$$

表示第 i 个粒子和其余粒子间的平均距离, $d_{\min}, d_{\max}$ 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。定义

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

显然 $E_f \in [0, 1]$ 且在收敛状态下粒子群将紧跟当前全局最优粒子飞向最优解, 此时 E_f 会逐渐减小; 而在跳出状态下全局最优粒子倾向于远离种群, 此时 E_f 相对较大。因此 E_f 能恰当刻画种群的进化状态, 称 E_f 为进化因子。根据 E_f 的取值, 规定:

$$\xi = \begin{cases} 1, & 0.00 \leq E_f < 0.25, \\ 2, & 0.25 \leq E_f < 0.50, \\ 3, & 0.50 \leq E_f < 0.75, \\ 4, & 0.75 \leq E_f \leq 1. \end{cases} \quad (7)$$

在种群进化过程中, 下一进化状态不可避免的受到当前状态的影响, 本文引入一种预测机制, 利用概率转移矩阵来通过当前进化状态对下一状态进行预测。记 $\xi^{(k)}$ 为第 k 次迭代时的当前进化状态, $\xi'^{(k)}$ 为预计的下一进化状态。取概率转移矩阵为:

$$M^{(k)} = \begin{bmatrix} \varphi & 1-\varphi & 0 & 0 \\ \frac{1-\varphi}{2} & \varphi & \frac{1-\varphi}{2} & 0 \\ 0 & \frac{1-\varphi}{2} & \varphi & \frac{1-\varphi}{2} \\ 0 & 0 & 1-\varphi & \varphi \end{bmatrix} \quad (8)$$

其中

$$M_{ij}^{(k)} = P\{\xi'^{(k)} = j \mid \xi^{(k)} = i\}, \forall i, j = 1, 2, 3, 4$$

表示种群根据当前进化状态 i 去预测下一状态 j 的概率, 满足 $\sum_{j=1}^4 M_{ij}^{(4)} = 1$ 。这里 φ 是一个非常重要的参数。为既保持种群进化的惯性, 也增加其多样性, 本文参考 Zeng [12], 选取 $\varphi = 0.9$ 。这意味着种群有很大的可能维持当前的状态, 而以较小的概率转移到另一进化状态。

3.2. 新算法的构建

传统的粒子群算法专注于当前信息对寻找最优解的影响而忽略了过去的信息。但在实际搜索过程中每一个粒子对每次迭代的速度、位置、个体最优以及全局最优等信息都保留有记忆, 充分综合这些信息更有利于种群搜索最优解。2019 年 Liu 等人引入随机分布式时延项, 提出 RODDPSO 算法[9], 此算法不仅考虑种群当前状态的个体最优和全局最优信息, 还充分考虑历史上个体最优和全局最优信息, 但是忽略了不同阶段的历史信息对当前状态的影响是不同。事实上, 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。基于这一事实, 本文提出一种带有分布式自适应时延的粒子群算法(PSO-DW), 其更新公式为:

$$\begin{aligned} V_i^{(k+1)} &= \omega V_i^{(k)} + c_1 r_1 (P_i^{(k)} - X_i^{(k)}) + c_2 r_2 (P_G^{(k)} - X_i^{(k)}) \\ &\quad + m_l (\xi'^{(k)}) c_3 r_3 \sum_{\tau=1}^N \omega_l \alpha_{(\tau)} (P_i^{(k-\tau)} - X_i^{(k)}) \\ &\quad + m_G (\xi'^{(k)}) c_4 r_4 \sum_{\tau=1}^N \omega_l \alpha_{(\tau)} (P_G^{(k-\tau)} - X_i^{(k)}), \\ X_i^{(k+1)} &= X_i^{(k)} + V_i^{(k+1)}. \end{aligned} \quad (9)$$

对比式(1), 这里引进了关于个体认知和社会认知的分布式时延项。其中 τ 是时延的步数, 可取值 $1, 2, \dots, N$, N 为分布式时延步数的上限值; ω_1 是当时延发生时的自适应权重, 它决定了每个时延影响的大小; $\alpha_{(\tau)}$ 从 $\{0, 1\}$ 中随机选取, 以增加粒子的多样性; c_3, c_4 是加速度系数; r_3, r_4 是 $[0, 1]$ 上服从均匀分布的随机向量; $m_l(\xi^{(k)})$ 和 $m_G(\xi^{(k)})$ 是分布式时延项的强度因子, 由预测的进化状态 $\xi^{(k)}$ 确定, 主要用来提高算法的搜索能力和加快收敛速度。需要说明的是当 $\tau \geq k$ 时约定 $P_i^{(k-\tau)} = P_i^{(k)}$, $P_G^{(k-\tau)} = P_G^{(k)}$ 。

3.3. 参数配置

新的粒子群算法带有两个权重参数 ω 和 ω_1 。其中惯性权重 ω 由式(2)确定; 自适应权重 ω_1 用来控制时延项对速度的影响, 它是时变的, 为关于 τ 的递减函数, 本文取

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

加速度系数 c_1, c_2 由式(3)、(4)确定, c_3, c_4 与 c_1, c_2 所起作用类似, 本文取 $c_3 = c_1$, $c_4 = c_2$ 。强度因子 $m_l(\xi^{(k)})$ 和 $m_G(\xi^{(k)})$ 主要用来提高算法的搜索能力, 加快算法的收敛速度, 初值都取为 0, 在进化过程中, 其大小由预测状态所确定(见表 1)。当根据 k 次迭代后种群的状态预测到下一状态是收敛状态时, 粒子将紧跟全局最优粒子快速到达最优解, 为保持这种趋势, 忽略时延项的影响而取 $m_l(1) = m_G(1) = 0$; 当预测到下一状态是开发状态时, 粒子将在个体最优位置仔细搜索, 因此个体的历史最优位置有着重要的参考意义, 取 $m_l(2) = 0.01$, $m_G(2) = 0$ 。当预测到下一状态是勘探状态时, 粒子将在整个搜索空间尽可能的探索全局最优解, 此时需充分考虑全局历史最优信息, 取 $m_l(3) = 0$, $m_G(3) = 0.01$; 当预测到下一状态是跳出状态时, 个体和全局的历史最优位置需要综合考虑, 取 $m_l(4) = 0.01$, $m_G(4) = 0.01$ 。需指出的是参数 0.01 是个经验值。

Table 1. Intensity factor selection strategy

表 1. 强度因子选择策略

状态	状态值	$m_l(\xi^{(k)})$	$m_G(\xi^{(k)})$
收敛	$\xi^{(k)} = 1$	0	0
开发	$\xi^{(k)} = 2$	0.01	0
勘探	$\xi^{(k)} = 3$	0	0.01
跳出	$\xi^{(k)} = 4$	0.01	0.01

4. 仿真实验与结果分析

4.1. 实验环境

为验证新算法的性能, 本文引入九个基准函数进行测试, 其中既有单峰函数又有多峰函数, 基准函数的具体信息见表 2; 本文实验在高维搜索空间进行, 取维数 $D = 150$, 粒子数 $S = 20$, 最大迭代次数为 20,000 次, 为消除随机因素的影响, 每次实验重复 40 次, 最后取平均值。每次实验之初随机初始化粒子的速度 V_i 和位置 $X_i, i \in I$, 然后计算出每一个粒子的适应度值, 根据适应度值确定个体最优粒子的位置 P_i 和全局最优粒子的位置 P_G 。

4.2. 上限值 N 的选取

在 PSO-DW 算法中, 分布式时延步数的上限值 N 是一个非常重要的参数, 不同的 N 将影响算法的收敛速度和最终的收敛值。本文将通过实验, 对比 N 取 75、100、125、150、175、200 时, 根据算法在 150

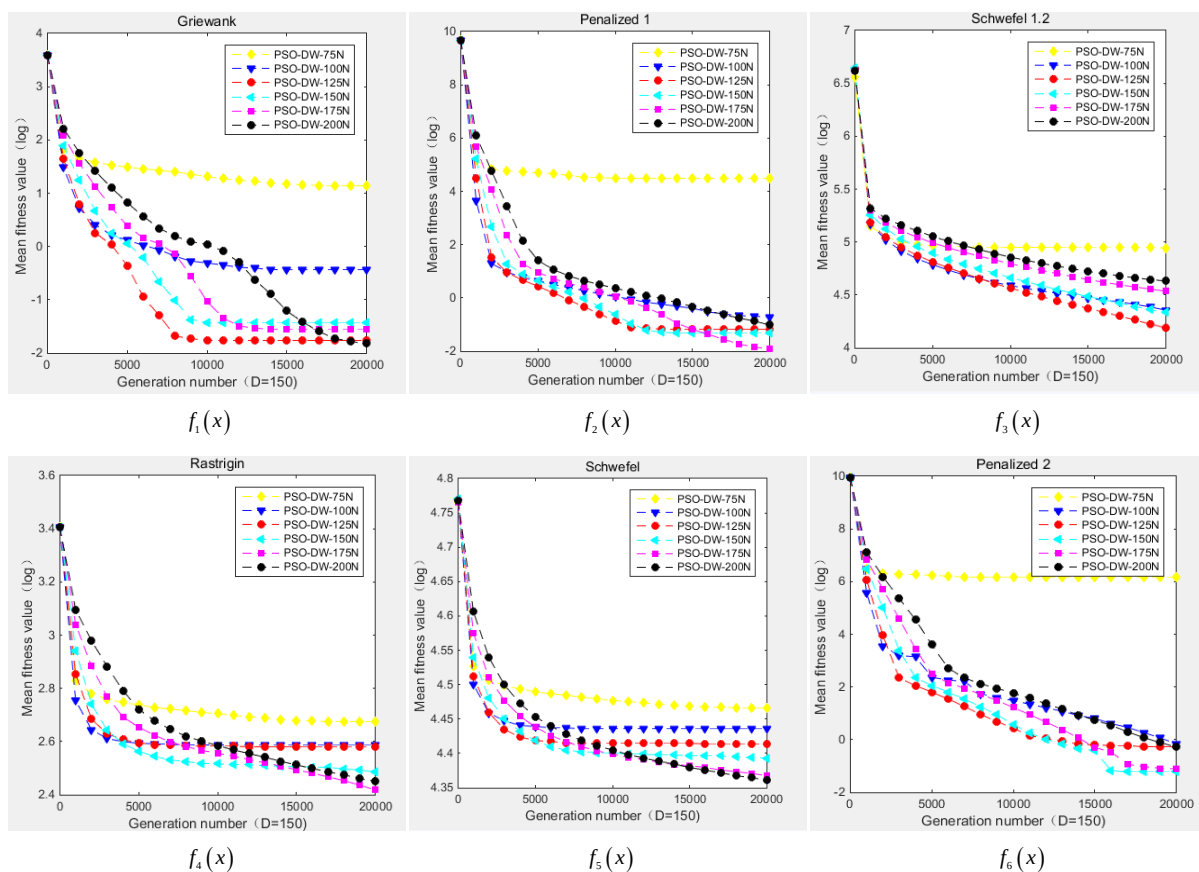
维空间中的九个基准函数上的不同表现, 选取鲁棒性最强的 N 。实验结果如图 1 所示。在图 1 中, 纵坐标是平均适应度值的对数值, 横坐标是迭代次数。

从图 1 中可以看出, 当分布时延项数的上限值 N 取 125 时算法在 $f_1(x)$, $f_3(x)$, $f_8(x)$, $f_9(x)$ 相对于其他的 N 性能更优, 不仅拥有较好的适应度值, 而且相对而言较早开始收敛。对于函数 $f_2(x)$, $f_4(x)$, $f_5(x)$, $f_6(x)$, $f_7(x)$, 虽然 N 取 125 没有最优的适应度值, 但与其它适应度值相差不大, 而 $N = 125$ 时有更早的收敛趋势, 因此本文选取 $N = 125$ 。

Table 2. Benchmark function

表 2. 基准函数

函数	名字	搜索空间
$f_1(x)$	Griewank	$[-600, 600]$
$f_2(x)$	Penalized 1	$[-50, 50]$
$f_3(x)$	Schwefel 1.2	$[-100, 100]$
$f_4(x)$	Rastrigin	$[-5.12, 5.12]$
$f_5(x)$	Schwefel	$[-500, 500]$
$f_6(x)$	Penalized 2	$[-50, 50]$
$f_7(x)$	Step	$[-100, 100]$
$f_8(x)$	Schwefel2.22	$[-10, 10]$
$f_9(x)$	Schwefel2.21	$[-100, 100]$



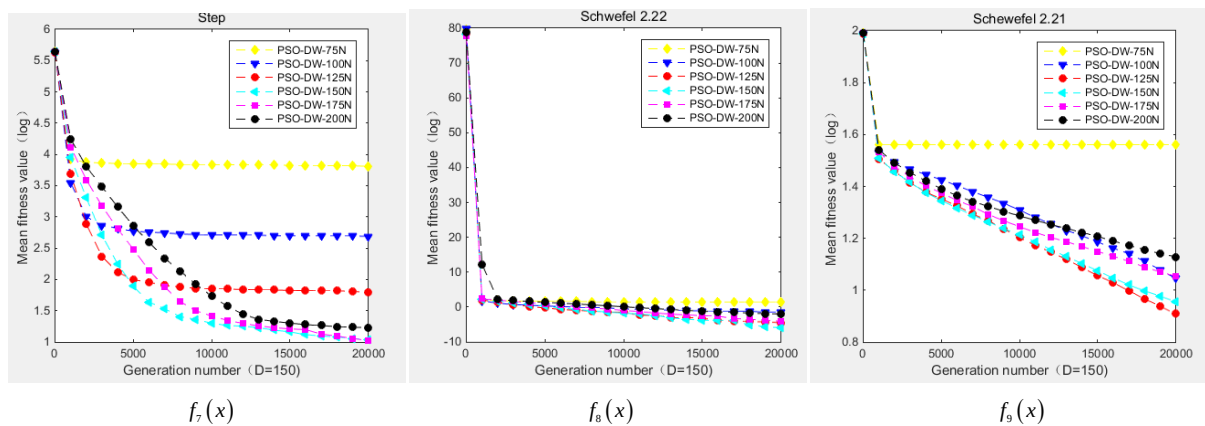


Figure 1. Training N by PSO-DW algorithm

图 1. 由 PSO-DW 算法训练 N

4.3. 实验结果与分析

为验证算法性能在高维搜索空间中的优越性,本文选用 PSO-CK、PSO-TVAC、PSO-LDIW、RODDPSO 四种算法作对比,评价指标为最小值、均值、方差。实验结果如表 3 和图 2 所示。表 3 表明,在相同的实验环境下,PSO-DW 算法在九个基准函数上表现出明显优势。首先九个平均适应度值都是最小的,这表明 PSO-DW 算法有较好的寻优质量,其收敛精度更高;其次除了在 $f_4(x)$ 上,PSO-DW 算法的方差也是最小的,这说明 PSO-DW 算法的稳定性较好;就最小值而言 PSO-DW 算法的仅在 $f_3(x)$ 上略逊于 PSO-CK 算法。进一步仔细比较,算法性能表现次优的是 RODDPSO 算法,而 PSO-CK、PSO-TVAC、PSO-LDIW 算法精确度普遍较差,这表明在高维搜索空间中,PSO-CK、PSO-TVAC、PSO-LDIW 算法容易收敛于局部极值,而引入时延项的 PSO 算法,充分考虑了种群的个体最优历史信息 and 全局历史信息,更加有利于增加粒子的多样性,表现出更好的跳出局部最优的能力。

Table 3. Comparison of performance of five algorithms in 150 D search space

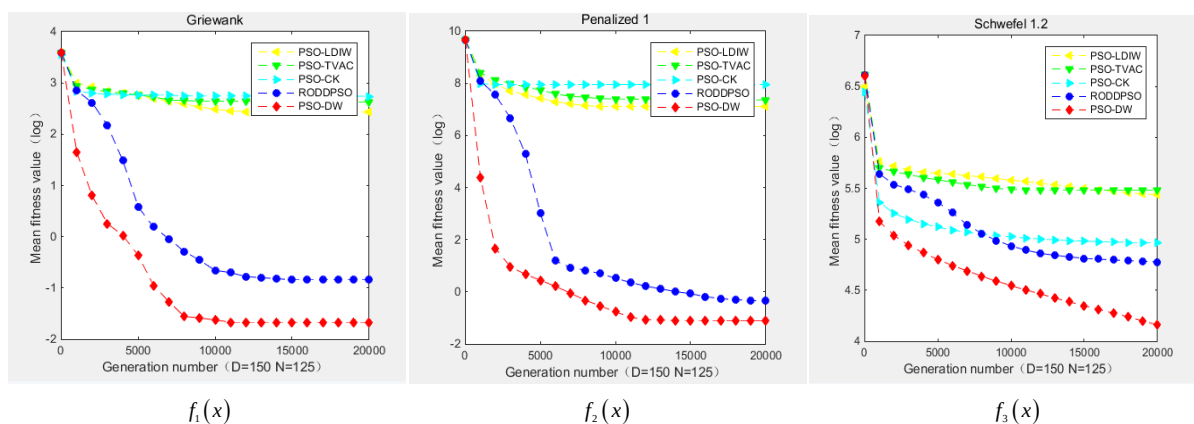
表 3. 五种算法在 150D 的搜索空间中的性能比较

函数	评价指标	PSO-CK	PSO-LDIW	PSO-TVAC	RODDPSO	PSO-DW
$f_1(x)$	最小值	2.71×10^2	3.43×10^{-2}	2.99×10^2	2.32×10^{-11}	8.66×10^{-15}
	均值	5.50×10^2	3.27×10^2	4.13×10^2	1.43×10^{-1}	2.15×10^{-2}
	方差	1.83×10^2	1.63×10^2	4.13×10^2	1.43×10^{-1}	2.15×10^{-2}
$f_2(x)$	最小值	4.17×10^{-2}	2.51×10^{-4}	5.71×10^6	4.93×10^{-9}	1.88×10^{-15}
	均值	8.96×10^7	1.92×10^7	2.23×10^7	4.42×10^{-1}	7.98×10^{-2}
	方差	1.79×10^8	6.83×10^7	9.90×10^6	5.66×10^{-1}	1.51×10^{-1}
$f_3(x)$	最小值	3.30×10^3	1.54×10^5	2.12×10^5	1.70×10^4	8.04×10^3
	均值	9.21×10^4	2.29×10^5	3.07×10^5	6.00×10^4	1.45×10^4
	方差	5.69×10^4	6.01×10^4	4.69×10^4	2.30×10^4	4.01×10^3
$f_4(x)$	最小值	7.26×10^2	5.49×10^2	1.23×10^3	3.69×10^2	2.71×10^2
	均值	9.01×10^2	7.72×10^2	1.40×10^3	5.24×10^2	3.80×10^2
	方差	1.20×10^2	1.14×10^2	8.44×10^1	7.11×10^1	8.17×10^1

Continued

$f_5(x)$	最小值	2.55×10^4	2.56×10^4	3.88×10^4	2.20×10^4	2.18×10^4
	均值	3.09×10^4	3.07×10^4	4.44×10^4	3.05×10^4	2.59×10^4
	方差	2.28×10^3	2.50×10^3	4.52×10^3	9.51×10^3	1.98×10^3
$f_6(x)$	最小值	1.59×10^0	6.42×10^{-1}	4.17×10^7	4.61×10^{-8}	1.40×10^{-12}
	均值	5.33×10^8	1.33×10^8	9.14×10^7	1.81×10^0	3.51×10^{-1}
	方差	4.17×10^8	2.35×10^8	3.44×10^7	2.31×10^0	9.70×10^{-1}
$f_7(x)$	最小值	4.13×10^4	1.00×10^4	2.42×10^4	2.70×10^1	1.20×10^1
	均值	7.75×10^4	3.53×10^4	4.18×10^4	2.04×10^2	4.06×10^1
	方差	1.93×10^4	1.65×10^4	7.92×10^3	1.05×10^2	3.03×10^1
$f_8(x)$	最小值	9.07×10^1	1.60×10^2	3.55×10^2	2.40×10^{-1}	9.20×10^{-9}
	均值	1.89×10^2	2.79×10^2	4.67×10^2	3.41×10^1	4.19×10^{-6}
	方差	4.52×10^1	9.75×10^1	3.45×10^2	2.13×10^1	1.51×10^{-5}
$f_9(x)$	最小值	2.93×10^1	2.98×10^1	5.52×10^1	1.64×10^1	6.34×10^0
	均值	3.31×10^1	3.48×10^1	6.13×10^1	2.36×10^1	8.17×10^0
	方差	1.87×10^0	2.32×10^0	3.42×10^0	2.77×10^0	1.42×10^0

图 2 表明在 150 维搜索空间中, 五种算法都有较好的收敛, 但 PSO-DW 的优势更加明显。由于分布式时延的引入, 历史最优信息被充分考虑, PSO-DW 和 RODDPSO 算法在寻优前期都表现出较好的全局搜索能力, 快速朝最优解方向搜索, 并且在寻优中期和晚期中表现出较好的局部搜索和跳出局部最优的能力, 实现收敛能力的提升。对比 RODDPSO 算法, 一方面由于考虑了时延的时变性, PSO-DW 算法具有更加准确的搜索性能, 大大提升了收敛精确度; 另一方面, 由预计状态确定的强度因子的引入, 促使种群由当前状态快速向下一进化状态转化, 加快了算法的收敛速度。综上所述, 在 150 维的搜索空间中, 改进的 PSO-DW 算法在搜索过程中较好地平衡了全局搜索和局部搜索的能力, 降低了陷入局部最优的可能性, 实现了收敛速度和收敛精度的提升。



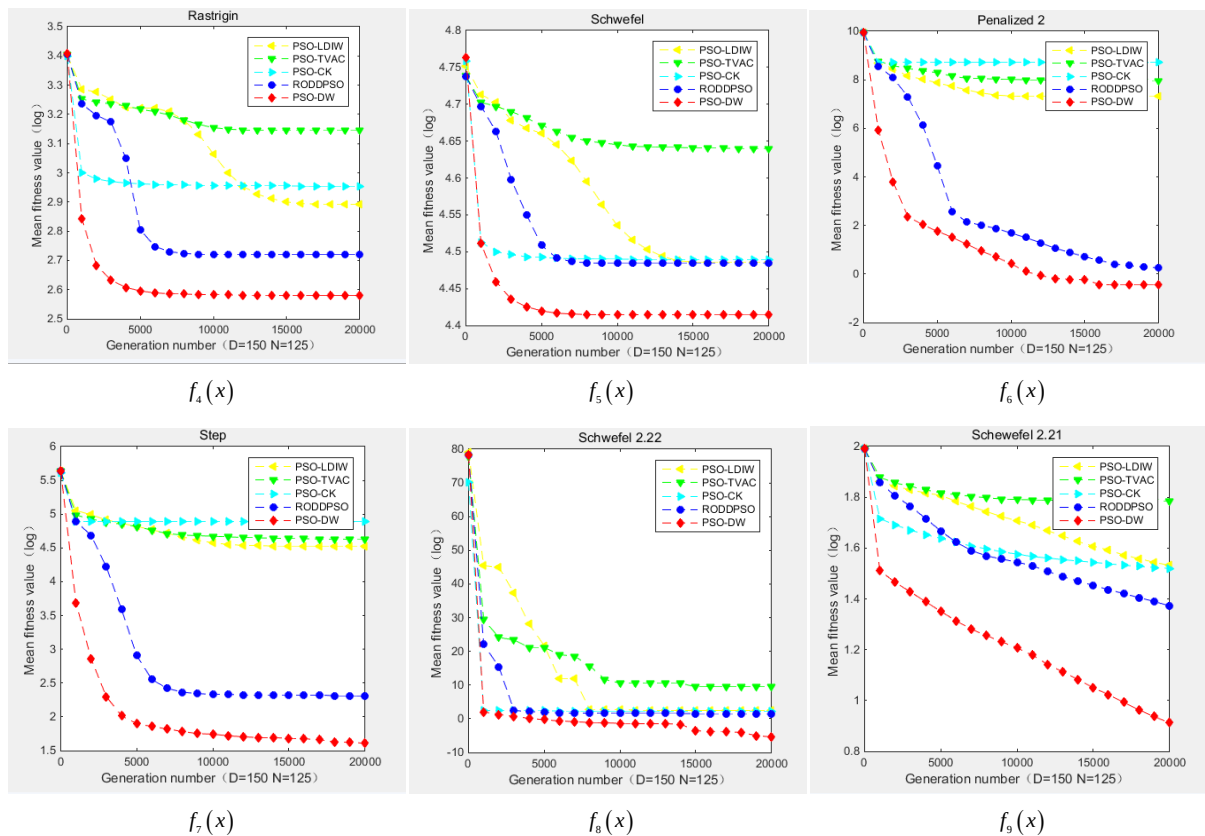


Figure 2. Comparison of five algorithms in 150-dimensional search space

图 2. 五种算法在 150 维的搜索空间中的对比图

5. 结论

考虑到种群的进化状态和历史最优信息, 本文提出一种带分布式自适应时延的粒子群优化算法 (PSO-DW), 该算法大大降低了种群在搜索过程中陷入局部最优的可能性, 提升了收敛速度和收敛精度。实验在 150 维搜索空间进行, 迭代 20,000 次, 每次实验重复 40 次, 工作量较大; 实验结果表明在高维搜索空间中新算法有较好的寻优能力和稳定性。

基金项目

本项目由国家自然科学基金项目: 61873169 资助。

参考文献

- [1] 陈贵敏, 贾建援, 韩琪. 粒子群优化算法的惯性权重递减策略研究[J]. 西安交通大学学报, 2006, 40(1): 53-56.
- [2] Eberhart, R. and Kennedy, J. (1995) A New Optimizer Using Particle Swarm Theory. *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, Nagoya, 4-6 October 1995, 39-43.
- [3] Shi, Y. and Eberhart, R.C. (1998) Parameter Selection in Particle Swarm Optimization. *Proceedings of the 7th International Conference on Evolutionary Programming*, San Diego, 25-27 March 1998, 591-600. <https://doi.org/10.1007/BFb0040810>
- [4] Shi, Y. and Eberhart, R. (1998) A Modified Particle Swarm Optimizer. *Proceedings of IEEE International Conference on Evolutionary Computation*, Anchorage, 4-9 May 1998, 69-73.
- [5] Shi, Y. and Eberhart, R.C. (1999) Empirical Study of Particle Swarm Optimization. *Proceedings of the 1999 Congress on Evolutionary Computation*, Washington DC, Vol. 3, 1945-1950.

- [6] Clerck, M. and Kennedy, J. (2002) The Particle Swarm: Explosion, Stability, and Convergence in a Multidimensional Complex Space. *IEEE Transactions on Evolutionary Computation*, **6**, 58-73. <https://doi.org/10.1109/4235.985692>
- [7] Ratnaweera, A., Halaamuge, S. and Watson, H.C. (2004) Self Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients. *IEEE Transactions on Evolutionary Computation*, **8**, 240-255. <https://doi.org/10.1109/TEVC.2004.826071>
- [8] 蔡燕敏. 一种新的混沌粒子群优化算法[J]. 智能计算机与应用, 2017, 7(2): 63-66.
- [9] Liu, W., Wang, Z. and Liu, X.H. (2019) A Novel Particle Swarm Optimization Approach for Patient Clustering from Emergency Departments. *IEEE Transactions on Evolutionary Computation*, **23**, 632-644. <https://doi.org/10.1109/TEVC.2018.2878536>
- [10] Zhan, Z., Zhang, J., Li, Y. and Chung, H. (2009) Adaptive Particle Swarm Optimization. *IEEE Transactions on Systems*, **39**, 1362-1381. <https://doi.org/10.1109/TSMCB.2009.2015956>
- [11] Tang, Y., Wang, Z. and Fang, J. (2011) Parameters Identification of Unknown Delayed Genetic Regulatory Networks by a Switching Particle Swarm Optimization Algorithm. *Expert Systems with Applications*, **38**, 2523-2535. <https://doi.org/10.1016/j.eswa.2010.08.041>
- [12] Zeng, N.Y., Wang, Z., Zhang, H. and Alsaadi, F.E. (2016) A Novel Switching Delayed PSO Algorithm for Estimating Unknown Parameters of Lateral Flow Immunoassay. *Cognitive Computation*, **8**, 143-152. <https://doi.org/10.1007/s12559-016-9396-6>
- [13] Song, B.Y., Wang, Z. and Zou, L. (2017) On Global Smooth Path Planning for Mobile Robots Using a Novel Multimodal Delayed PSO Algorithm. *Cognitive Computation*, **9**, 5-17. <https://doi.org/10.1007/s12559-016-9442-4>
- [14] Song, B.Y., Xu, J.W. and Xu, L. (2018) PSO-Based Extended Kalman Filtering for Speed Estimation of an Induction Motor. 2018 37th Chinese Control Conference (CCC), Wuhan, 25-27 July 2018, 3803-3807. <https://doi.org/10.23919/ChiCC.2018.8482581>
- [15] 张文成, 周穗华, 吴志东, 等. 改进策略的粒子群算法及其实验测试[J]. 计算机工程与设计, 2012, 33(10): 3945-3948.
- [16] 杜江, 袁中华, 王景芹. 动态改变惯性权重的新模式粒子群算法[J]. 安徽大学学报(自然科学版), 2018, 42(2): 60-62.