

一种新的混合粒子群优化算法

徐生兵, 蹇柯, 夏文杰

(东莞城市学院计算机与信息学院, 广东 东莞 523419)

✉xusb@ccdgt.edu.cn; jianke@ccdgt.edu.cn; xiawj@ccdgt.edu.cn



摘要: 针对粒子群算法在进化后期收敛精度低、收敛速度慢, 尤其是高维时候容易早熟等问题, 提出了一种新的混合粒子群优化算法。新算法首先设计了一种新的惯性权重, 使惯性权重取值在进化初期和后期都较为适中; 其次, 为了有效抑制粒子陷入局部极值, 引入了粒子最优速度和最差适应值的概念, 并以此为基础, 设计了粒子的一种新的自适应变异方式; 最后引入了平均收敛率和最小平均收敛代数两个概念, 可以更好地评价和比较本文算法的性能。八个标准测试函数在100维、200维进行的数值实验证实, 新算法收敛精度高, 收敛速度快, 且有效预防了早熟现象。

关键词: 粒子群优化; 惯性权重; 早熟; 变异

中图分类号: TP183 **文献标识码:** A

Algorithm of a New Hybrid Particle Swarm Optimization

XU Shengbing, JIAN Ke, XIA Wenjie

(School of Computer and Information, City College of Dongguan, Dongguan 523419, China)

✉xusb@ccdgt.edu.cn; jianke@ccdgt.edu.cn; xiawj@ccdgt.edu.cn

Abstract: This paper proposes a new hybrid particle swarm optimization algorithm to solve the problems of low convergence accuracy, slow convergence speed of particle swarm optimization algorithm in the late evolution stage, and being prone to mature early especially in the high-dimensional case. The new algorithm first proposes to design a new inertia weight, which makes the value selection of inertia weight moderate in the early and late evolution. Secondly, in order to effectively restrain the particles from falling into the local extreme value, the concepts of particle optimal velocity and the worst fitness of particles are introduced. Based on this, a new adaptive mutation method of particles is designed. Finally, the concepts of average convergence rate minimum average convergence algebra are introduced, which can better evaluate and compare the performance of the proposed algorithm. The numerical experiments of 8 standard test functions in 100 and 200 dimensions verify that the new algorithm has high convergence accuracy, fast convergence speed, and effectively prevents premature phenomenon.

Keywords: particle swarm optimization; inertia weight; premature; mutation

1 引言(Introduction)

粒子群优化(PSO)算法是由Eberhart、Kennedy于1995年提出的一种基于种群智能行为的优化算法。由于其概念明确、需要设置的参数少、编程易于实现等优点, 目前在工程领域已经得到广泛的应用。但PSO算法在优化复杂函数问题时极易陷入局部极值, 并且会出现早熟现象, 这又限制了其进一步应用。为提高PSO算法的优化性能, 许多学者提出了各种改进的PSO算法^[1-10]。其中, 在惯性权重方面, SHI等人^[1]于

1998年首次引入了惯性权重的概念, 并提出了一种线性递减的权重策略(LDWPSO); 黄轩等人^[2]提出了一种基于随机惯性权重的快速PSO算法(Faster PSO with Random inertia weight, FRPSO), 即惯性权重在0.4—0.6的随机取值; LIANG等人^[3]提出了一种利用种群质心和种群个体极值质心的PSO算法(PSO with Centroid, CPSO), 使得粒子的更新不仅与传统PSO算法中的个体极值和全局极值相关, 而且还与种群中其他粒子的位置和个体极值相关。

2 标准粒子群算法(Standard particle swarm optimization algorithm)

标准粒子群算法的数学描述过程如下：

在 d 维搜索空间中，PSO算法首先初始化 s 个随机粒子，然后通过追踪两个极值(全局极值和个体极值)位置对粒子的位置进行更新，迭代运行直至满足停机条件，最后输出最优解。其中，在第 t 代时，假设某个粒子 $i(1 \leq i \leq s)$ 的位置为 $X_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{id}^t)$ ，速度为 $V_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{id}^t)$ ，则第 $t+1$ 代粒子 i 的位置和速度分别为：

$$V_i^{t+1} = wV_i^t + c_1\zeta(P_i^t - X_i^t) + c_2\eta(P_g^t - X_i^t) \quad (1)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (2)$$

其中，式(1)、式(2)分别称为速度更新方程和位置更新方程； w 称为惯性权重，表示对前一代速度的保留程度，一般取值范围为 $[0.1, 0.9]$ ； c_1 、 c_2 称为学习因子，一般取值为 2.0 ； ζ 、 η 是 $(0,1)$ 之间均匀分布的随机数； $P_i^t = (p_{i1}^t, p_{i2}^t, \dots, p_{id}^t)$ 为粒子 i 自身迄今为止找到的最好位置，即粒子 i 的个体极值位置； $P_g^t = (p_{g1}^t, p_{g2}^t, \dots, p_{gd}^t)$ 为整个种群迄今为止找到的最好位置，即种群的全局极值位置。

上述的惯性权重 w 是 PSO 算法的一个十分重要的参数。SHI 等人通过大量的数值实验表明：较大的惯性权重有利于全局搜索，而较小的惯性权重有利于局部搜索，因而可以在迭代初期设置一个较大的惯性权重，以便较为迅速地定位到一个较优的搜索区域；在迭代后期使用一个较小的惯性权重，以便在这个较优的区域进行局部搜索，进而搜索到更好的解。因此，SHI 提出了一种惯性权重线性递减(Linear-Decreased Weight, LDW)的策略，即：

$$w = w_{\max} - (w_{\max} - w_{\min}) \times t / T_{\max} \quad (3)$$

其中， w_{\max} 为最大惯性权重， w_{\min} 为最小惯性权重，一般取 $w_{\max} = 0.9$ ， $w_{\min} = 0.1$ ； t 为当前迭代次数； T_{\max} 为总迭代次数。这样， w 就会随着迭代次数的增加而线性递减。因此，使用式(3)权重策略的 PSO 算法被称为惯性权重线性递减的 PSO 算法，即 LDWPSO 算法。

3 一种新的混合粒子群优化算法(A new hybrid particle swarm optimization algorithm, NHPSO)

3.1 一种新的惯性权重策略

由于惯性权重在 PSO 算法中可以控制算法的全局搜索能力和局部搜索能力，因此，关于惯性权重的研究已经取得了不少的进展^[2,9]，其中文献[2]经过大量数值实验证实：惯性权重在 $0.4-0.6$ 随机取值时，其总体实验效果要优于 LDWPSO。与 LDWPSO 的惯性权重相比，FRPSO 的特点是：初期惯性权重较小，后期惯性权重较大。但后期惯性权重较大是不利于粒子局部搜索的(这在文献[7]FRPSO 与 LDWPSO 的对比实验中也可以看出来)。考虑到两种权重的各自特点和实验效果，本文构造了一种新的惯性权重，使其值在初期和后期均介于两者之间，具体如下：

期均介于两者之间，具体如下：

$$w = \frac{3}{7}w_1 + \frac{4}{7}w_2 \quad (4)$$

其中， $w_1 = 0.6(t/T_{\max} - 2/3)^2 + 0.1$ ， $w_2 = 1.2e^{(-t/T_{\max})} - 0.3$ 。 w_1 是一个变化范围较小的惯性权重，其变化范围约为 $[0.1, 0.37]$ ，且在迭代后期稍有回升； w_2 是一个非线性递减的权重，其变化范围约为 $[0.14, 0.9]$ ，这样 w 的变化范围约为 $[0.15, 0.67]$ 。因此，与 LDWPSO 中的惯性权重相比，新惯性权重初期较小，后期稍大；与 FRPSO 惯性权重相比，新惯性权重初期稍大，后期较小，符合设计的初衷。三种权重的变化如图1所示(以 $T_{\max} = 1,000$ 为例)。

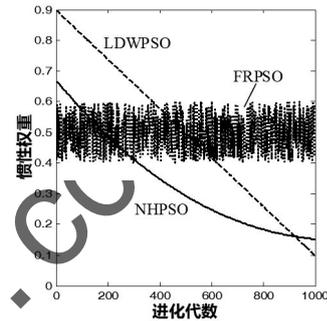


图1 三种PSO算法权重变化示意图

Fig.1 Schematic diagram of weight variation of three PSO algorithms

3.2 一种新的变异策略

PSO 算法在陷入局部极值的时候，粒子之间的差异性很小，因此，让部分粒子变异将使得算法有机会搜索到更好的解。本文利用粒子的速度信息和种群适应值的信息，构造了一种新的粒子位置的变异策略。为了叙述方便，先对粒子最优速度 V_{best} 和最差适应值 f_{worst} 做如下定义：

定义1：PSO 算法中，某个粒子 i 在第 t 代的最优速度

$$V_{\text{best}}^t = \begin{cases} V_i^t & \text{if } f(X_i^t) < f(X_i^{t-1}) \\ V_i^0 & \end{cases}, V_i^0 \text{ 表示粒子 } i \text{ 的初始速度。}$$

定义2：PSO 算法中，某个粒子 i 在第 t 代的最差适应值

$$f_{\text{worst}}^t = \begin{cases} f(X_i^t) & \text{if } f(X_i^t) > f(X_i^{t-1}) \\ f(X_i^0) & \end{cases}, f(X_i^0) \text{ 表示粒子 } i \text{ 的初始适应值。}$$

那么，对粒子 i 的变异操作是：

$$X_i^t = |f_{\text{worst}}^t - f_i^t| \times [V_{\text{best}}^t + f(P_g)] \quad (5)$$

其中， f_i^t 表示粒子 i 在第 t 代的适应值； V_{best}^t 表示全局极值粒子的最优速度， $f(P_g)$ 表示整个种群的当前最优适应值，用来给 V_{best}^t 一个速度扰动。由于 $|f_{\text{worst}}^t - f_i^t|$ 是一个较大的数，因此粒子可以以很大概率远离当前位置。选取哪些粒子变异也是提高 PSO 算法性能的关键，本文变异的准则如下：

设种群在第 t 代各粒子的适应值分别为 $f_1^t, f_2^t, \dots, f_s^t$ ，则第 t 代种群的平均适应值 $f_{\text{avg}}^t = \sum_{i=1}^s f_i^t / s$ ，粒子 i 与 f_{avg}^t 绝对偏

差 $\delta_i^t = |f(P_i^t) - f_{avg}^t|$ ，检验值 $\lambda_i^t = f_{avg}^t / s$ 。其中， $f(P_i^t)$ 为粒子 i 的个体极值， s 为粒子个数。对每个需要进行变异粒子的选取准则是 $\delta_i^t < \lambda_i^t$ 。为了有效防止粒子陷入局部极值和确保算法的稳定性，在每一次迭代中对每一个满足准则且不是全局极值的粒子按式(5)进行变异操作。

3.3 NHPSO算法流程

NHPSO算法的计算流程与LDWPSO算法的计算流程基本一致，不同之处有：(1)需要计算初始化粒子的最优速度和最差适应值，在每次更新的时候也要更新粒子的最优速度和最差适应值；(2)在每一次迭代过程中，对满足变异条件的粒子均要进行变异操作。NHPSO算法的计算流程图如图2所示。

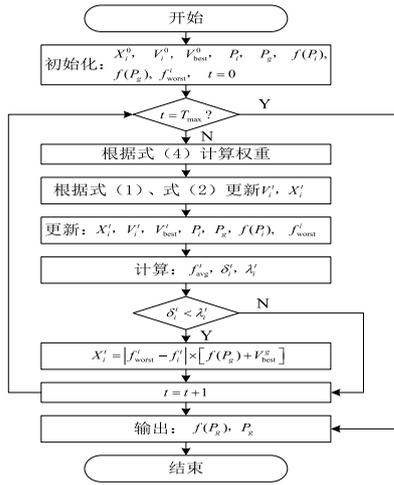


图2 NHPSO计算流程图

Fig.2 Flow chart of NHPSO calculation

4 数值仿真实验(Numerical simulation experiments)

4.1 标准测试函数

在数值仿真实验中，选取如表1所示的八个标准测试函数对LDWPSO、FRPSO、CPSO和NHPSO进行性能对比实验。

表1 标准测试函数列表

Tab.1 List of standard test functions

测试函数表达式	搜索区间	理论最优解	预设精度
$f_1(X) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	$x_i = 0$	$\leq e-10$
$f_2(X) = \sum_{i=1}^n (x_i - i)^2$	$[-n, n]$	$x_i = i$	$\leq e-4$
$f_3(X) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]$	$x_i = 0$	$\leq e-5$
$f_4(X) = \sum_{i=1}^n \frac{1}{4000} x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$	$x_i = 0$	$\leq e-8$
$f_5(X) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-100, 100]$	$x_i = 0$	$\leq e-5$
$f_6(X) = \sum_{i=1}^{n-1} 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2$	$[-5, 5]$	$x_i = 1$	$\leq e-6$
$f_7(X) = 418.9829n + \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	$[-500, 500]$	$x_i = 420.96$	$\leq e-4$
$f_8(X) = 78.3323n + \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$	$[0, -50]$	$x_i = -2.9035$	$\leq e-6$

上述测试函数的理论全局最优解均为0。其中， f_1 、 f_3 容易优化； f_2 低维容易优化，高维很难优化； f_4 、 f_5 的优化难度

较大； f_6 、 f_7 、 f_8 优化难度最大， f_6 是典型的极难优化非凸病态函数， f_7 是带有一定欺骗性的函数，因为全局最优与最好的局部最优相距很远，因此搜索算法往往朝着错误的方向收敛， f_8 是一个 2^n 最小化问题，一般的PSO算法极易陷入局部极值且极难优化。

4.2 实验参数设置及实验方案设计

4.2.1 实验参数设置

在实验中，所有算法的公共参数设置为： $s = 80$ ， V_{max} 为搜索区间长度的0.1，粒子的初始化区间为如表1所示的搜索区间，但 f_2 的初始区间为 $[-300, 300]$ 。LDWPSO算法中， $w_{max} = 0.9$ ， $w_{min} = 0.1$ ；FRPSO和CPSO的参数设置均按参考文献设置，且算法优化性能也与参考文献一致。

4.2.2 实验方案设计

为了叙述方便，先给出算法平均收敛率和平均最小收敛代数的定义。

定义3：如果算法在指定 T_{max} 代的运算中，最终优化结果 $f(P_g) < prec$ ，其中 $prec$ 为预设精度，则称该次算法收敛。若算法在 M 次运算中，有 M_s 次收敛，则平均收敛率 $\bar{p} = M_s / M$ 。

定义4：当算法收敛时，第一次满足预设精度的迭代次数 t_{min} 称为该次算法的最小收敛代数；如果算法在 T_{max} 次运算中不收敛，则 $t_{min} = T_{max}$ 。设算法在 M 次运算中，最小收敛代数分别为 $t_{min}^1, t_{min}^2, \dots, t_{min}^M$ ，则最小平均收敛代数 $\bar{t}_{min} = \sum_{k=1}^M t_{min}^k / M$ 。

实验方案1：在固定 T_{max} 和 $prec$ 的条件下，比较 \bar{p} 和 \bar{t}_{min} 。该方案中， $d = 100$ ， $T_{max} = 2,500$ ， $M = 50$ ， $prec$ 如表1所示，实验结果如表2(其中 \bar{p} 、 \bar{t}_{min} 在表2中分别用I、II表示)所示。

实验方案2：固定 T_{max} ，比较算法在 $d = 200$ 时的平均值、最小值、方差。此方案中， $T_{max} = 2,500$ ， $M = 20$ 。实验结果如表3和表4所示，从 f_1 至 f_8 在 $d = 200$ 时的进化曲线如图3至图10所示。

4.3 实验结果

表2、表3、表4是八个标准测试函数在实验方案1和方案2下得到的实验结果。

表2 实验方案1实验结果

Tab.2 The results of experimental scheme 1

函数 算法	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
	LDWPSO	I 0	I 0.02	I 0	I 0	I 0	I 0	I 0
	II 2,500	II 2,489	II 2,500					
CPSO	I 0	I 0	I 0	I 0	I 0	I 0	I 0	I 0
	II 2,500							
FRPSO	I 0	I 0.06	I 0	I 0	I 0	I 0	I 0	I 0
	II 2,500	II 2,499	II 2,500					
NHPSO	I 0.98	I 1.0	I 0.98	I 0.96	I 1.0	I 0.98	I 0.98	I 1.0
	II 249.1	II 2,068	II 374.4	II 1,232	II 279.9	II 382.4	II 1,988	II 2,230

表3 实验方案2实验结果(f_1-f_4)

Tab.3 The results of experimental scheme 2(f_1-f_4)

函数	f_1	f_2	f_3	f_4
LDWPSO 平均值	3.032e+1	5.905e+4	1.106e+1	7.988e-1
LDWPSO 最小值	5.844	1.949e+4	3.001	2.260e-1
LDWPSO 方差	4.553e+2	1.295e+9	5.988e+1	8.734e-2
CPSO 平均值	4.191e-1	1.613e+5	1.550e-1	2.143e-1
CPSO 最小值	1.225e-1	8.112e+4	8.816e-2	4.216e-2
CPSO 方差	1.469e-1	1.865e+9	2.390e-3	6.072e-2
FRPSO 平均值	1.282e+1	6.967e+3	2.055	1.012
FRPSO 最小值	2.642	3.900e+1	3.953e-1	7.669e-1
FRPSO 方差	4.662e+1	1.930e+8	1.351e+1	1.315e-2
NHPSO 平均值	2.173e-5	7.640	1.783e-3	3.787e-5
NHPSO 最小值	0	2.250	0	0
NHPSO 方差	8.580e-10	2.968e+1	1.563e-6	2.518e-9

表4 实验方案2实验结果(f_5-f_8)

Tab.4 The results of experimental scheme 2(f_5-f_8)

函数	f_5	f_6	f_7	f_8
LDWPSO 平均值	1.429e+3	4.813e+2	4.213e+4	1.376e+4
LDWPSO 最小值	1.194e+3	2.527e+2	3.925e+4	1.292e+4
LDWPSO 方差	1.727e+4	1.008e+4	3.071e+6	1.184e+5
CPSO 平均值	2.346e+3	2.994e+2	5.391e+4	9.574e+3
CPSO 最小值	1.974e+3	2.101e+2	5.085e+4	7.481e+3
CPSO 方差	2.423e+4	3.479e+3	3.446e+6	3.241e+5
FRPSO 平均值	1.225e+3	5.127e+2	4.222e+4	4.388e+6
FRPSO 最小值	8.555e+2	2.973e+2	3.682e+4	7.268e+5
FRPSO 方差	1.626e+4	7.887e+3	1.089e+7	4.776e+12
NHPSO 平均值	3.625e-3	2.950e-4	4.988	3.848e-1
NHPSO 最小值	0	0	3.222e-1	8.836e-2
NHPSO 方差	5.732e-5	2.198e-7	4.336e+1	4.690e-2

图3至图10分别是八个测试函数分别在LDWPSO、FRPSO、CPSO和NHPSO下的迭代进化图。

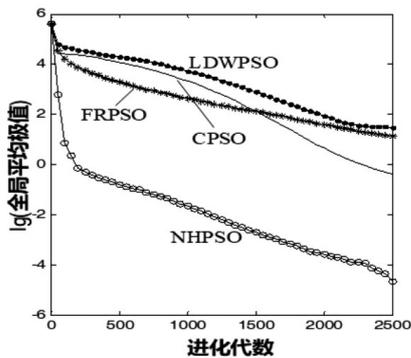


图3 f_1 进化过程

Fig.3 Evolutionary process of f_1

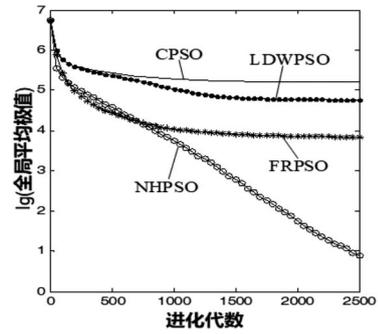


图4 f_2 进化过程

Fig.4 Evolutionary process of f_2

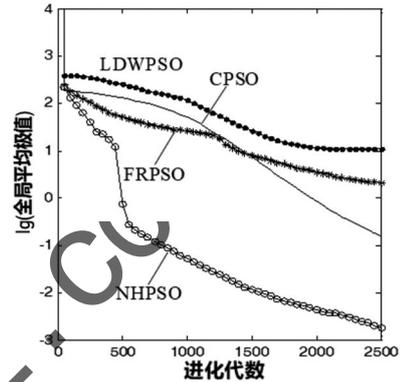


图5 f_3 进化过程

Fig.5 Evolutionary process of f_3

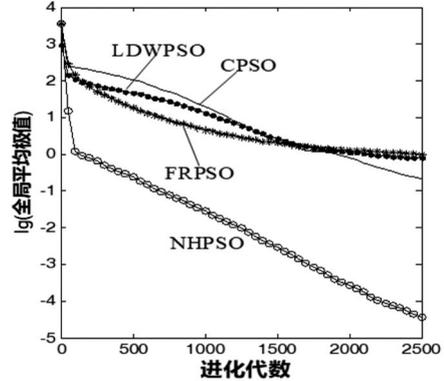


图6 f_4 进化过程

Fig.6 Evolutionary process of f_4

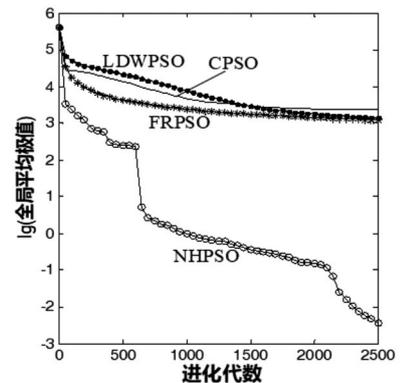


图7 f_5 进化过程

Fig.7 Evolutionary process of f_5

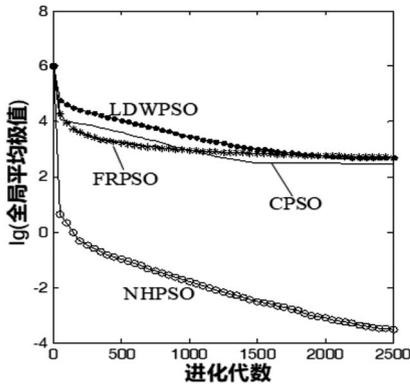


图8 f_6 进化过程

Fig.8 Evolutionary process of f_6

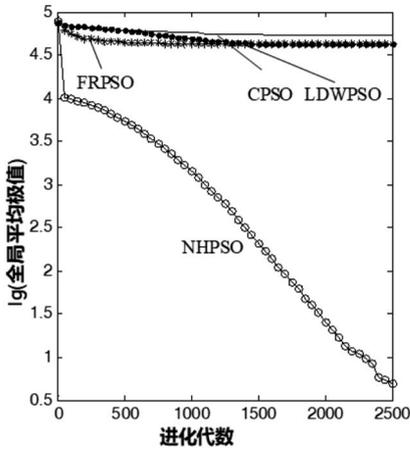


图9 f_7 进化过程

Fig.9 Evolutionary process of f_7

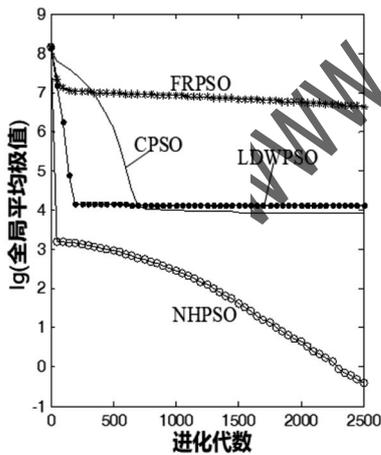


图10 f_8 进化过程

Fig.10 Evolutionary process of f_8

由上述实验结果可以看出, NHPSO无论在收敛精度、收敛速度还是收敛率方面, 都要显著优于其他三种算法, 并且在后期仍有很强的全局搜索能力, 有效抑制了粒子早熟。这主要得益于NHPSO有较为适中的惯性权重, 使之能在进化初期搜索到一个较好的候选优化位置, 在进化后期收敛的同时仍具有较强的全局搜索能力; 在每一次迭代中, 对满足条件的粒

子采取了一种新的自适应变异策略, 提高了种群的多样性。

5 结论(Conclusion)

本文针对PSO算法的早熟收敛问题, 提出了一种新的混合粒子群优化算法——NHPSO算法。在NHPSO中, 使用新惯性权重策略有效平衡了粒子的全局搜索能力和局部搜索能力; 在每次迭代中, 按确定条件选择(而不是随机选择)粒子并对其进行新的自适应变异, 不仅有利于算法的稳定性, 而且有效抑制了粒子早熟。数值仿真实验证实了以上两点。

由于NHPSO在100维和200维有很好的优化效果, 因此是一种很有实用价值的智能优化算法。

参考文献(References)

- [1] SHI Y, EBERHART R. A modified particle swarm optimizer[C]// IEEE.1998 Proceedings of the IEEE International Conference on Evolutionary Computation(CEC). Piscataway, USA: IEEE, 1998:69-73.
- [2] 黄轩,张军,詹志辉.基于随机惯性权重的快速粒子群优化算法[J].计算机工程与设计,2009,30(03):55-57.
- [3] LIANG S J, SONG S L, LI K, et al. An improved particle swarm optimization algorithm and its convergence analysis[C]// IEEE. 2nd International Conference on Computer Modeling and Simulation(ICCMS). New York, USA: IEEE, 2010: 138-141.
- [4] 杨英杰.粒子群算法及其应用研究[M].北京:北京理工大学出版社,2017:1-186.
- [5] 李根.基于云任务调度及粒子群算法的网络安全系统设计[J].软件工程,2018,21(05):28-30.
- [6] 赵红超,周洪庆,王书湖.无人机三维航迹规划的量子粒子群优化算法[J].航天控制,2021,39(01):40-45.
- [7] 陈强,王宇嘉,梁海娜,等.目标空间映射策略的高维多目标粒子群优化算法[J].智能系统学报,2021,16(02):362-370.
- [8] 田梦丹,梁晓磊,符修文,等.具有博弈概率选择的多子群粒子群算法[J].计算机科学,2021,48(10):67-76.
- [9] 杨博雯,钱伟懿.多惯性权重的自适应粒子群优化算法[J].渤海大学学报,2021,42(01):45-52.
- [10] 王建丽.基于随机微粒群算法的开放实验室规划研究[J].软件工程,2021,24(10):28-30.

作者简介:

徐生兵(1980-), 男, 硕士, 讲师.研究领域: 智能计算及其应用.

蹇柯(1983-), 男, 硕士, 讲师.研究领域: 智能优化, 盲源分离.

夏文杰(1981-), 男, 硕士, 讲师.研究领域: 计算机专色油墨配色的理论与实现.