

# 一种改进的均值粒子群算法<sup>①</sup>

张 亮, 谢富强, 陈 立

(南华大学 电气工程学院, 衡阳 421001)

**摘 要:** 为了解决粒子群算法存在“早熟”现象和收敛速度慢的问题, 本文提出一种改进的均值粒子群算法. 该算法采用非线性惯性权重, 同时在每个迭代步, 将粒子历史最优和种群全局最优取均值再乘以一个非线性权重的方法, 以提高算法的全局搜索能力和收敛速度. 通过 4 个标准函数的测试, 实验结果表明该算法的有效性.

**关键词:** 粒子群算法; 非线性惯性权重; 权重均值; 均值粒子群算法; 优化算法

## Improved Mean Particle Swarm Optimization Algorithm

ZHANG Liang, XIE Fu-Qiang, CHEN Li

(University of South China, College of Electric Engineering, Hengyang 421001, China)

**Abstract:** In order to solve premature phenomenon and slow convergence problems in particle swarm algorithm, an improved mean particle swarm algorithm is provided. The algorithm apply to the nonlinear weight. At the same time in each iteration step, the history optimal particle and the global optimal population were averaged and multiplied a nonlinear weight so that it improve the global search capacity and convergence speed. Through four standard function test, The results show that the effectiveness of the proposed algorithm.

**Key words:** particle swarm algorithm; nonlinear inertia weight; mean weight; mean particle swarm algorithm; optimization algorithms

① 收稿时间:2013-09-25;收到修改稿时间:2013-10-18

粒子群优化(Particle Swarm Optimization, PSO)算法是一种基于群体智能方法的演化计算技术, 它是由美国的 J. Kennedy 和 R. C. Eberhart 受鸟群觅食行为的启发于 1995 年提出的一种智能优化算法, 该算法因其算法简单, 调节参数少, 易于实现, 一经提出就受到许多专家学者的关注. 但由于粒子群算法存在易陷入局部极值点或早熟收敛等问题, 许多研究者提出了改进的办法, 如文献[1]和文献[2]中 R.Eberhart 等人先后提出的线性惯性权重的 PSO 算法以及模糊惯性权重的 PSO 算法, 该算法提高收敛速度. 文献[3]中 M.Clerc 等人提出了具有收缩因子的 PSO 算法. 文献[4]中添加变异算子, 改善种群多样性. 文献[5]中将 PSO 算法与模拟退火法结合, 提出了一种基于模拟退火的 PSO 算法, 该算法能提高收敛精度, 但在搜索过程中需要调整退火结果的条件, 计算量大, 不便于实际应用. 文

献[6]用平均值的概念提出均值粒子群算法, 该算法采用线性惯性权重提高了收敛的速度, 但算法的稳定性不好, 本文提出一种非线性惯性权重和均值相结合的算法, 以提高收敛速度、稳定性.

### 1 标准的粒子群算法

PSO 算法是一种群体智能算法, 它利用  $m$  个粒子组成的粒子群在  $D$  维目标搜索空间中以迭代的方式寻找最优解. 在每步迭代中, 第  $i$  个粒子速度与位置的更新表达式如下:

$$V_{id}(t+1) = WV_{id}(t) + C_1 r_{1d}(t)(p_{id}(t) - X_{id}(t)) + C_2 r_{2d}(t)(p_{gd}(t) - X_{id}(t)) \quad (1)$$

$$X(t+1) = X(t) + V(t+1) \quad (2)$$

其中  $i=1, 2, \dots, m$ ,  $d=1, 2, \dots, D$ , 在  $D$  为空间中,  $V_{id}(t)$  和

$X_{id}(t)$  表示第  $i$  个粒子在  $t$  时刻的速度与位置.  $p_{id}(t)$  表示第  $i$  个粒子经历过的最优位置,  $p_{gd}(t)$  表示第  $i$  个粒

子的所有邻域个体曾经历过的全局历史最优位置,  $c_1$  与  $c_2$  为学习因子,  $r_{1d}(t)$  与  $r_{2d}(t)$  为在  $[0,1]$  内均匀分布的随机数, 惯性权重  $w$  是线性递减, 如(3)所示

$$w(t) = w_{end} + (1 - t / T_{max})(w_{start} - w_{end}) \quad (3)$$

其中  $T_{max}$  为最大的迭代次数,  $t$  为当前的迭代次数,  $w_{start}$  和  $w_{end}$  分别为初始惯性权重和进化最大的迭代次数时的惯性权重, 当  $w_{start}=0.9$ ,  $w_{end}=0.4$  时算法的性能会大大提高, 与基本的 PSO 算法相比较其收敛速度快且求解精度高.  $V_{id} \in [V_{min}, V_{max}]$ , 是当前设定的最大粒子速度(由用户设定). 粒子在解空间内不断跟踪个体极值与全局极值进行搜索. 直到达到规定的迭代次数或满足规定的误差标准为止.

## 2 非线性惯性权重的PSO算法

标准的粒子群算法中惯性权重  $w$  为线性递减, 线性递减的惯性权重只对某些问题有效, 当待求解的问题为难优化的复杂函数时, PSO 算法在迭代后期局部搜索能力不强, 易陷入局部最优, 导致不能找到所要求的最优解. 非线性惯性权重[7-8]是指惯性权重的变化曲线为非线性. 针对上面的问题提出一种非线性的惯性权重如式(4)所示.

$$w(t) = w_{end} + (w_{start} - w_{end}) * \exp(-25 * (t / T_{max})^k) \quad (4)$$

式中  $w_{start}=0.9$ ,  $w_{end}=0.4$ ,  $k$  为调节因子, 本文中  $k=6$ , 惯性权重  $w$  曲线如图 1 所示.

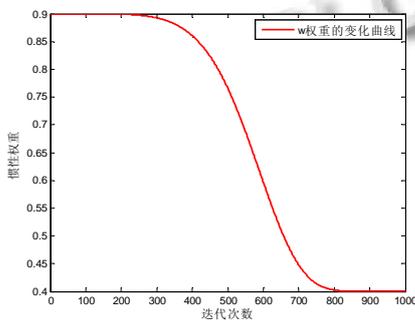


图 1 惯性权重的变化曲线

对于难优化的复杂函数, 在迭代初期需要加强全局搜索能力, 相应的会增加全局搜索时间, 一旦确定最优解的大致范围, 就需要加强局部搜索能力, 相应

的增加局部搜索时间. 在式(3)的线性递减权重策略中,  $w$  从  $w_{start}$  线性递减到  $w_{end}$ , 全局搜索和局部搜索的斜率没有变化, 全局搜索和局部搜索时间太短, 用式(3)作 PSO 的惯性权重去优化复杂的函数很难找到要求的最优解.

从图 1 的惯性权重变化曲线可以看出, 本文提出的惯性权重  $w$  在迭代初期持续一段时间保持一个较大的值, 在迭代后期持续一段时间保持了一个较小的值, 从而增加迭代初期的全局搜索时间和迭代后期的局部搜索时间, 提高在迭代初期的全局搜索能力, 以及在迭代后期的局部搜索能力, 更好的平衡全局搜索能力与局部搜索能力. 另一种表述方法如式(5).

$$w = \begin{cases} w_{start} & t \in [0, 0.25T_{max}] \\ w(t) = w_{end} + (1 - t / T_{max})(w_{start} - w_{end}) & t \in (0.25T_{max}, 0.8T_{max}) \\ w_{end} & t \in [0.8T_{max}, T_{max}] \end{cases} \quad (5)$$

惯性权重  $w$  在迭代前期以  $w=0.9$  持续一段时间来增加全局搜索能力, 在迭代中期以线性递减的方式进行搜索, 在迭代后期  $w=0.4$  持续一段时间来增加局部搜索能力, 通过这种方式也能更好的平衡全局搜索能力与局部搜索能力.

## 3 改进的均值粒子群算法

在粒子群算法中, 粒子根据个体历史最优  $p_{id}$  和整个粒子全局最优  $p_{gd}$  不断的调整自己的速度和位置, 使粒子朝全局最优位置飞去. 均值粒子群算法<sup>[9-10]</sup>是用局部最优  $p_{id}(t)$  和全局最优  $p_{gd}(t)$  的线性组合  $\frac{p_{id}(t) + p_{gd}(t)}{2}$  和  $\frac{p_{id}(t) - p_{gd}(t)}{2}$  取代标准粒子群算法中的  $p_{id}(t)$  和  $p_{gd}(t)$ , 如式(6)(7)所示.

$$V_{id}(t+1) = WV_{id}(t) + c_1 r_{1d} (\frac{p_{id}(t) + p_{gd}(t)}{2} - X_{id}(t)) + c_2 r_{2d} (\frac{p_{id}(t) - p_{gd}(t)}{2} - X_{id}(t)) \quad (6)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (7)$$

式(6)中第一项为粒子先前的速度, 第二项为粒子的“认知”部分, 它吸引粒子从当前位置指向由全局最优和粒子个体最优构成的平均位置方向, 第三部分为“社会”部分, 它吸引粒子从当前位置指向由全局最优位置和粒子个体最优位置构成的平均位置负方向.

均值粒子群算法中将(6)中的  $p_{id}(t)$  和  $p_{gd}(t)$  的系数

$$V_{id}(t+1) = WV_{id}(t) + \frac{c_1 r_{1d}}{2} (\frac{p_{id}(t) + p_{gd}(t)}{2} - X_{id}(t)) - (c_1 r_{1d} + c_2 r_{2d}) X_{id}(t)$$

合并得到式(8).

(8)

当  $c_1r_{id} \gg c_2r_{2d}$  时,  $p_{id}(t)$  和  $p_{gd}(t)$  前的系数为正, 粒子收敛到全局最优点.

当  $c_1r_{id} = c_2r_{2d}$  时, 式(8)中的第三项为零, 此时粒子只有“认知”部分而没有“社会”部分, 粒子之间互不影响, 每个粒子都在各自范围内搜索, 没有信息的共享和相互合作, 粒子收敛到全局最优点的能力不足.

当  $c_1r_{id} \ll c_2r_{2d}$  时,  $p_{id}(t)$  前的系数为正,  $p_{gd}(t)$  前的系数为负, 粒子不能收敛到全局最优点.

本文提出改进的算法, 其表达式如式(9)和(10).

$$V_{id}(t+1) = WV_{id}(t) + c_1r_{id}(t) \left( \frac{1}{W} \frac{p_{id}(t) + p_{gd}(t)}{2} - X_{id}(t) \right) + (W - \frac{1}{2}) \frac{p_{id}(t) - p_{gd}(t)}{2} - X_{id}(t) + c_2r_{2d}(t) \left( \frac{1}{W} \frac{p_{id}(t) - p_{gd}(t)}{2} - (W - \frac{1}{2}) \frac{p_{id}(t) + p_{gd}(t)}{2} - X_{id}(t) \right) \quad (9)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (10)$$

改进的均值粒子群算法是在均值粒子群算法的基础上添加系数  $1/w$  和  $w-1/2$ , 将式(9)式中的  $p_{id}(t)$  和  $p_{gd}(t)$  的系数合并得到式(11).

$$V(t+1) = WV(t) + \left( \frac{1}{W} - W + \frac{1}{2} \right) \frac{c_1r_{id} + c_2r_{2d}}{2} p_{id} + \left( \frac{1}{W} + W - \frac{1}{2} \right) \frac{c_1r_{id} - c_2r_{2d}}{2} p_{gd} - (c_1r_{id} + c_2r_{2d}) X_{id}(t) \quad (11)$$

将式(8)与式(11)比较可以知道  $p_{id}$  与  $p_{gd}$  的系数前添加了  $(1/w-w+1/2)$  和  $(1/w+w-1/2)$ , 由式(4)得  $W \in [0.9, 0.4]$  是非线性的递减, 那么  $(1/w-w+1/2) \in [1.2, 2.1]$ ,  $(1/w+w-1/2) \in [1, 2.9]$ , 都是非线性递增, 即在迭代前期提高粒子全局搜索能力, 粒子迅速飞行到最优位置附近, 在迭代后期粒子的局部搜索能力增强, 粒子飞行到最优点.

权重均值粒子群算法实现步骤如下:

**Step1** 初始化种群中粒子的位置与速度, 设定最大速度  $V_{max}$  与位置边界  $X_{max}, X_{min}$

**Step2** 计算每个粒子的适应度值, 以及个体极值  $p_{id}$  和全局极值  $p_{gd}$ .

**Step3** 根据公式(9)和(10)来更新粒子的速度和位置.

**Step4** 评价每一个粒子, 粒子  $i$  的适应度值为  $f_{id}$ , 若  $f_{id}$  优于  $p_{id}$  的适应度值, 则更新  $p_{id}$  为粒子  $i$  的当前位置, 反之  $p_{id}$  保持不变; 若  $f_{id}$  优于  $p_{gd}$  的适应度值, 则新  $p_{gd}$  为粒子  $i$  的当前位置, 反之  $p_{gd}$  保持不变.

**Step5** 判断是否满足结束条件, 若满足则执行 step6, 若不满足转到 step3

**Step6** 输出全局最优粒子  $p_{gd}$  和最优适应度值, 算法运行结束.

## 4 仿真实验与分析

### 4.1 测试函数

为了验证本算法的有效性, 将本算法与基本粒子群算法对下列测试函数进行仿真计算并对结果进行比较分析.

$$f_1(x) = \sum_{i=1}^n x_i^2 \quad |x_i| \leq 5.12 \quad (12)$$

$$f_2(x) = -20e^{-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}} - e^{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)} + 20 + e \quad |x_i| \leq 30 \quad e=2.71282 \quad (13)$$

$$f_3(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10] \quad |x_i| \leq 5.12 \quad (14)$$

$$f_4(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 + \prod_{i=1}^n \cos \frac{x_i}{\sqrt{i}} + 1 \quad |x_i| \leq 600 \quad (15)$$

其中  $f_1(x)$  是 sphere 函数, 这是一个单峰值函数, 该函数在  $x_i=0$  达到最小值 0.

$f_2(x)$  是 Ackley 函数是多峰值函数, 该函数在  $x_i=0$  时达到最小值 0.

$f_3(x)$  是多峰值的 Rastrigrin 函数, 该函数具有大量局部最小点, 在  $x_i=0$  时达到全局最小值 0.

$f_4(x)$  是多峰值的 Griewank 函数, 在  $x_i=0$  时达到全局最小 0.

### 4.2 仿真结果与分析

该实验平台操作系统为 Windows XP, 编译环境为 Matlab2010A, CPU 为 Intel(R)Core(TM)2 Duo T5470, 1.6GHz, 内存为 2.0G, 我们将标准的 PSO 算法记为 PSO, 将线性递减的惯性权重算法记为 LDWPSO, 本文的算

法记为 WMPSO, 算法的参数设置及各个测试函数的搜索空间如表 1 所示.

表 1 各测试函数的搜索空间

函数	$(X_{min}, X_{max})$
$f_1(x)$	(-5.12, 5.12)
$f_2(x)$	(-30,30)

$f_3(x)$  (-5.12,5.12)

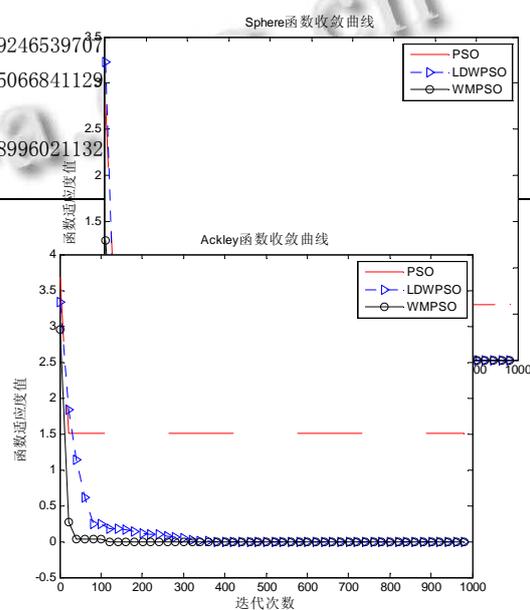
$f_4(x)$  (-600,600)

学习因子  $c_1 = c_2 = 2$ , 非线性惯性权重  $W \in [0.4,0.9]$ , 调节因子  $k=6$ , 种群规模为 60, 迭代次数 1000 次, 在 10 维空间中对各测试函数进行 100 次独立实验, 对比结果如表 2 所示.

表 2 LDWPSO 算法和 WMPSO 算法的结果对比

函数	算法	最优值	最差值	平均最优值
$f_1(x)$	PSO	0.020051983705929	0.091563567057143	0.052723041568453
	LDWP	7.498784932713408e-043	7.445106952981053e-025	7.446467756906439e-026
	WMPS	0	0	0
$f_2(x)$	PSO	0.526166193248407	1.532184515570019	1.001112939954039
	LDWP	0.5461828459041e-002	2.311386963388542e-002	0.506360264307041
	WMPS	0.5461828459045e-002	0.5461828459045e-002	0.5461828459045e-002
$f_3(x)$	PSO	12.408443062850179	34.377798952248177	20.874012161630546
	LDWP	3.979836228373161	13.929411685848844	7.362692992235095
	WMPS	0	0	0
$f_4(x)$	PSO	1.278718542443634	92.549246539707	3.5
	LDWP	1.561994820784768	33.225066841129	2.5
	WMPS	0.955488800333389	7.1298996021132	1.5

为了说明本算法的性能本文同时绘出了各个函数基于标准的 PSO 算法(PSO)、惯性权重线性递减的 PSO 算法(LDWPSO)和本文的算法(WMPSO)的优化曲线, 如图 2-图 5, 通过这些函数的优化曲线图可以看到均值权重的 PSO 算法相对其他 PSO 算法收敛速度更快, 同时收敛的精度提高.



Computation. 1998. 69-73.

图 3  $f_2$  函数的优化曲线

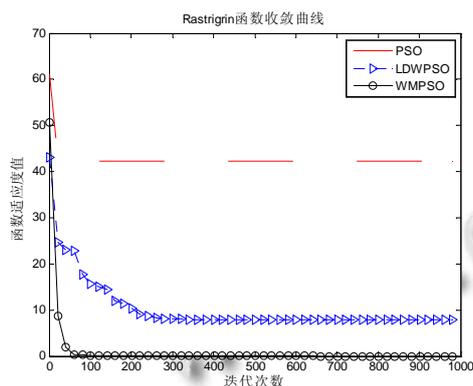


图 4  $f_3$  函数的优化曲线

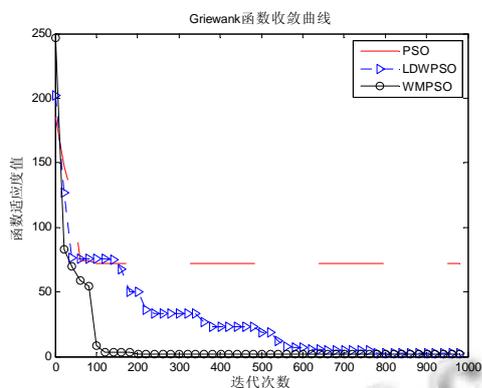


图 5  $f_4$  函数的优化曲线

### 5 结论

本文针对粒子群算法存在“早熟”现象和收敛速度慢的问题,提出一种改进的均值 PSO 算法,通过与传统的均值粒子群算法相比较,克服了传统均值粒子群算法的缺点,同时提高了粒子群算法收敛速度、全局收敛能力、算法的稳定性.

### 参考文献

1 Shi SY, Eberhart R. A modified particle swarm optimizer. Proc. of the IEEE International Conference on Evolutionary

2 Shi SY, Eberhart R. Fuzzy adaptive particle swarm optimization. Proc. of the IEEE Congress on Evolutionary Computation. 2001. 101-106.  
 3 Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evolutionary Computation. 2002, 6(1): 58-73.  
 4 焦巍,刘光斌,王凯.基于粒子多样性研究的改进 PSO 算法.系统仿真学报,2009,21(20):6483-6486.  
 5 王华秋,曹长修.基于模拟退火的并行粒子群优化研究.控制与决策.2005,20(5):500-504.  
 6 Deep K, Bansal JC. Mean particle swarm optimisation for function optimization. Computer Intelligence Studies, 2009, (1).  
 7 刘伟,周育人.一种改进惯性权重的 PSO 算法.计算机工程与应用,2009,45(7):46-48.  
 8 Chatterjee A, Siarry P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. Computers & Operations Research (S0305-0548), 2006, 33(3): 859-871.  
 9 刘红霞,周永权.一种基于均值的云适应粒子群算法.计算机工程与科学,2011,33(5):97-101.  
 10 杨艳,李树波.一种改进权均值粒子群优化算法.软件导

刊,2013,12(5):54-56.

## 数值优化改进的 BP 网络的模式分类对比<sup>①</sup>

丁 硕, 常晓恒, 巫庆辉

(渤海大学 工学院, 锦州 121013)

**摘 要:** 研究了共轭梯度算法、拟牛顿算法、LM 算法三类常用的数值优化改进算法, 基于这三类数值优化算法分别对 BP 神经网络进行改进, 并构建了相应的 BP 神经网络分类模型, 将构建的分类模型应用于二维向量模式的分类, 并进行了泛化能力测试, 将不同 BP 网络分类模型的结果进行对比. 仿真结果表明, 对于中小规模的网络而言, LM 数值优化算法改进的 BP 网络的分类结果最为精确, 收敛速度最快, 分类性能最优; 共轭梯度数值优化算法改进的 BP 网络的分类结果误差最大, 收敛速度最慢, 分类性能最差; 拟牛顿数值优化算法改进的 BP 网络的分类结果误差值、收敛速度及分类性能介于上述两种算法之间.

**关键词:** 数值优化; BP 神经网络; 模式分类; 泛化能力; 对比

### Comparative of Pattern Classification of BP Neural Networks Improved by Numerical Optimization Approach

DING Shuo, CHANG Xiao-Heng, WU Qing-Hui

(College of Engineering, Bohai University, Jinzhou 121013, China)

**Abstract:** Three common numerical optimization algorithms are first studied, including conjugate gradient algorithm, quasi-newton algorithm and LM algorithm. The three kinds of algorithms are used to improve BP neural network respectively and the corresponding classification models based on BP neural network are established. Then the models are used in pattern classification of two-dimensional vectors, and their generalization abilities are also tested. The classification results of different classification models based on BP network are compared with each other. Simulation results show that for small or medium scale networks, BP neural network improved by LM algorithm has the most precise classification result, the fastest convergence speed and the best classification ability. The one improved by conjugate gradient algorithm has the biggest error, slowest convergence speed and worst classification ability. While the classification precision, convergence speed and classification ability of quasi-newton algorithm lie between the above two algorithms.

**Key words:** numerical optimization; BP neural networks; pattern classification; generalization ability; comparison

射能力、大规模并行协同处理能力、鲁棒性和容错性, 这为构造神经网络模式分类系统提供了有力手段. 在众多神经网络模型中, 应用得最为广泛的模型为 BP 神经网络具有较强的自学习能力、非线性映