

微粒群算法的参数选择及收敛性分析

崔红梅,朱庆保

CUI Hong-mei,ZHU Qing-bao

南京师范大学 数学与计算机科学学院,南京 210097

School of Mathematics and Computer Science,Nanjing Normal University,Nanjing 210097,China

E-mail:rose_chm@sina.com

CUI Hong-mei,ZHU Qing-bao.Convergence analysis and parameter selection in particle swarm optimization.Computer Engineering and Applications,2007,43(23):89-91.

Abstract: Particle Swarm Optimization(PSO) is a novel optimization technology and has been applied successfully to various optimization problems.But the convergence of the algorithm has been studied insufficiently.The author studies the particle's trajectory and convergence of the existing PSO algorithm,on the basis of their characteristics,proposes a new PSO algorithm with a new inertia weight along broken line.This algorithm preserves the advantages of the incremental and reduced inertia weight,and overcomes their shortcomings.Simulated experiments achieve good results.

Key words: Particle Swarm Optimization(PSO);parameter selection;convergence;particle's trajectory;inertia weight

摘要:微粒群算法是相对较新颖的优化算法,已经成功应用于许多优化问题。然而算法的参数选择及收敛性分析研究不足,为此首先认真研究了现有微粒群算法粒子轨迹及其收敛性的文献,在此基础上,根据递减惯性权重和递增惯性权重微粒群算法各自的特点,结合算法的收敛区间,提出了一种具有先增后减惯性权重的新的微粒群算法,既保留了具有递增和递减惯性权重的优点,也克服了它们的缺点,取得了比较好的效果。

关键词:微粒群算法;参数选择;收敛性;粒子轨迹;惯性权重

文章编号:1002-8331(2007)23-0089-03 文献标识码:A 中图分类号:TP18

1 引言

1995年,美国社会心理学家 James Kennedy 与电气工程师 Russell Eberhart 在他们提交给 IEEE 神经网络国际会议的论文中首次提出了粒子群优化算法(Particle Swarm Optimization,简称 PSO 算法)^[1]。该算法源于对鸟类捕食行为的模拟,与遗传算法相似,粒子群优化算法也是一种基于群体和适应度的进化优化算法。与其它进化算法相比,粒子群优化算法的优势在于收敛速度快且容易实现。此外,粒子群优化算法需要调整的参数比较少,可以直接采用实数编码,算法结构相对简单。

粒子群优化算法已得到较广泛的应用,如何提高其收敛速度或保证收敛成为该算法的一个研究热点。例如,为了提高收敛速度,有学者提出了惯性权重 w 线性减少的算法^[2-3],也有文献报道了惯性权重 w 线性递增的算法^[4],这些算法在一定程度上改善了收敛速度和最优解的精确性。然而作者通过实验发现,较大的 w 具有较好的全局收敛能力,而较小的 w 则具有较强的局部搜索能力。具有递减惯性权重的算法在前期阶段具有较好的全局搜索能力,后期收敛性也较好,但收敛速度比较慢;而具有递增惯性权重的算法虽然前期收敛速度很快,但后期的局部搜索能力较差。结合 Ioan Cristian Trelea 和 Frans van den Berth 提出的算法的收敛区间^[5],本文提出了具有沿折线

先增后减的动态惯性权重的微粒群算法,这样在算法的前期阶段具有较快的收敛速度,而且在算法后期局部搜索能力也不错,既保留了具有递增惯性权重和递减惯性权重微粒群算法的优点,也克服了它们的缺点,取得了比较好的实验效果。

2 标准微粒群算法简介

粒子群优化算法首先初始化一群随机粒子(随机解),然后通过进化(迭代)找到最优解。每个粒子通过跟踪两个“极值”来更新自己:一个极值是粒子本身找到的最优位置,这个位置被称作个体极值 $pBest$;另一个极值是整个粒子群到目前为止找到的最优位置,通常被称作全局极值 $gBest$ 。

设第 i 个粒子的位置表示为:

$$X_i=(x_{i1},x_{i2},\dots,x_{id}) \quad (1)$$

其中, D 表示问题的维度。相应地,第 i 个粒子的速度可以表示为:

$$V_i=(v_{i1},v_{i2},\dots,v_{id}) \quad (2)$$

第 i 个粒子的当前最优位置可以表示为:

$$Y_i=(y_{i1},y_{i2},\dots,y_{id}) \quad (3)$$

同样,粒子群目前找到的全局最优位置可以表示为:

$$\hat{Y}_g=(\hat{y}_{g1},\hat{y}_{g2},\dots,\hat{y}_{gd}) \quad (4)$$

基金项目:江苏省自然科学基金(the Natural Science Foundation of Jiangsu Province of China under Grant No.2006218)。

作者简介:崔红梅(1977-),硕士研究生,主要从事智能算法与智能控制等方面的学习与研究;朱庆保(1955-),研究生导师,教授,主要从事机器人路径规划、人工智能与智能控制等方面的教学与研究。

在找到个体极值和全局极值后,粒子根据式(5)更新自己的速度和位置:

$$\begin{cases} v_{i,d}(t+1)=wv_{i,d}(t)+c_1r_1(y_{id}(t)-x_{i,d}(t))+c_2r_2(\hat{y}_{id}(t)-x_{i,d}(t)) \\ x_{i,d}(t+1)=x_{i,d}(t)+v_{i,d}(t+1) \end{cases} \quad (5)$$

式(5)中 d 的取值范围是 $[1, D]$ 。 $0 < w \leq 1$ 称作惯性权重,使得粒子保持运动惯性,具有扩展空间的趋势,有能力探索新的区域,用来平衡全局搜索能力和局部搜索能力。Y.Shi 和 R.Eberhart 在文献[2,3]中指出,较大的 w 有较好的全局收敛能力,而较小的 w 则有较强的局部搜索能力。 c_1 和 c_2 被称作学习因子,通常 $c_1=c_2=2$, $rand$ 表示 $(0, 1)$ 之间的随机数。在算法运行过程中,粒子的个体极值和粒子群的全局极值都不断更新。算法运行结束时,输出全局极值 $gBest$ 。

基本微粒群算法的流程如下:

- (1)依照初始化过程,对粒子群的随机位置和速度进行初始设定;
- (2)计算每个粒子的适应值;
- (3)对每个粒子,将其适应值与所经历过的最好位置 $pBest$ 的适应值进行比较,若较好,则将其作为当前的最好位置;
- (4)对每个粒子,将其适应值与全局所经历过的最好位置 $gBest$ 的适应值进行比较,若较好,则将其作为当前的全局最好位置;
- (5)根据式(5)对粒子的速度和位置进行进化;
- (6)如未达到结束条件通常为足够好的适应值或达到一个预设最大迭代次数,则返回步骤(2)。

3 算法改进、参数选择和收敛性分析

3.1 具有递减惯性权重的微粒群算法

文献[3]提出了一种递减惯性权重的 PSO 算法,令式(5)中 $c_1r_1=\phi_1, c_2r_2=\phi_2, \phi_1, \phi_2 \in [0, 2]$, 与标准算法的主要不同点在于:惯性权重 w 是迭代次数的函数,沿直线从 0.9 线性递减到 0.4, 即 w 满足:

$$w(k)=-0.5 \times \frac{k}{MaxNumber} + 0.9 \quad (6)$$

其中 k 为迭代序号, $MaxNumber$ 为最大迭代次数。

3.2 具有递增惯性权重的微粒群算法

文献[4]中的算法是一种具有递增惯性权重的微粒群算法,其主要特点是: $\phi_1, \phi_2 \in [0.5, 2]$, 即令 $\phi_i=0.5+1.5*r_i$, 其中 $i=1, 2$, r_i 为 $(0, 1)$ 之间的随机数。惯性权重 w 沿直线从 0.4 线性递增到 0.9, 即 w 满足:

$$w(k)=0.5 \times \frac{k}{MaxNumber} + 0.4 \quad (7)$$

3.3 具有先增后减惯性权重的微粒群算法

以上两种算法各有引言中所述的优点和不足,为此,作者研究了一种先增后减惯性权重的微粒群算法。以下给出算法分析和相关算法。

因为 PSO 算法中的不同维之间没有交互作用,所以在下面的讨论中,可以不失一般性假设 $D=1$, 这样就可以把下标中的符号 d 去掉。为了描述更简单,现假设只有一个粒子,只分析单个粒子的轨迹,这样可以把下标中的符号 i 去掉,分析粒子轨迹时只考虑离散时间,即用 x_t 表示 $x(t)$ 。这样式(5)就可以简化为:

$$\begin{cases} v_{t+1}=wv_t+\phi_1(y_t-x_t)+\phi_2(\hat{y}_t-x_t) \\ x_{t+1}=x_t+v_{t+1} \end{cases} \quad (8)$$

上式消去 v , 可以得到:

$$x_{t+1}=(1+w-\phi_1-\phi_2)x_t-wx_{t-1}+\phi_1y_t+\phi_2\hat{y}_t \quad (9)$$

可以写成矩阵形式:

$$\begin{bmatrix} x_{t+1} \\ x_t \\ 1 \end{bmatrix} = \begin{bmatrix} 1+w-\phi_1-\phi_2 & -w & \phi_1y_t+\phi_2\hat{y}_t \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ x_{t-1} \\ 1 \end{bmatrix} \quad (10)$$

$$\text{令 } \begin{bmatrix} 1+w-\phi_1-\phi_2 & -w & \phi_1y_t+\phi_2\hat{y}_t \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{A}, \text{ 即 } \begin{bmatrix} x_{t+1} \\ x_t \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_t \\ x_{t-1} \\ 1 \end{bmatrix}, \text{ 求得}$$

\mathbf{A} 的 3 个特征值为:

$$\begin{cases} \lambda_1=1 \\ \lambda_2=\frac{1+w-\phi_1-\phi_2+\Delta}{2} \\ \lambda_3=\frac{1+w-\phi_1-\phi_2-\Delta}{2} \end{cases}, \text{ 其中 } \Delta=\sqrt{(1+w-\phi_1-\phi_2)^2-4w}$$

粒子轨迹序列 $\{x_t\}_{t=0}^{+\infty}$ 的收敛性由 λ_2, λ_3 的范数决定。当 $\max\{\|\lambda_2\|, \|\lambda_3\|\} < 1$ 时, 序列 $\{x_t\}_{t=0}^{+\infty}$ 收敛。Ioan Cristian Trelea^[5] 和 Frans van den Berth^[6] 指出, $\max\{\|\lambda_2\|, \|\lambda_3\|\} < 1$ 等价于 $\phi_1+\phi_2 < 2*(w+1)$, 其中 $0 \leq w \leq 1$, 即在满足下列条件时 PSO 算法是收敛的:

$$\phi_1+\phi_2 < 2*(w+1) \quad (11)$$

收敛区间如图 1 所示。

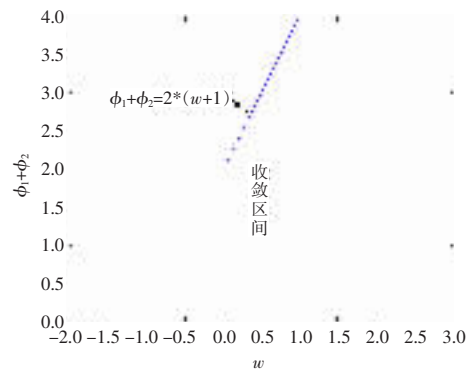


图 1 PSO 算法的收敛区间

依据上述分析和式(11), 令 $0 < \phi_1, \phi_2 < 2*(w+1)$, 即令:

$$\phi_1=(w+1)*r_1 \quad \phi_2=(w+1)*(2-r_1)*r_2$$

其中 r_1, r_2 为 $(0, 1)$ 之间的随机数。惯性权重 w 沿折线从 0.4 线性递增到 0.9, 然后再线性递减到 0.4, 即 w 满足:

$$w(k) = \begin{cases} 1 \times \frac{k}{MaxNumber} + 0.4 & 0 \leq \frac{k}{MaxNumber} \leq 0.5 \\ -1 \times \frac{k}{MaxNumber} + 1.4 & 0.5 < \frac{k}{MaxNumber} \leq 1 \end{cases} \quad (12)$$

$w(k)$ 如图 2 所示。

4 实验

为了有可比性,本文也采用文献[3,4]中使用的如下 4 个非线性标准测试函数进行实验:

(1) $f_1(x)$ 为 Sphere 函数

$$f_1(x) = \sum_{i=1}^n x_i^2, x \in (-100, 100)^n$$

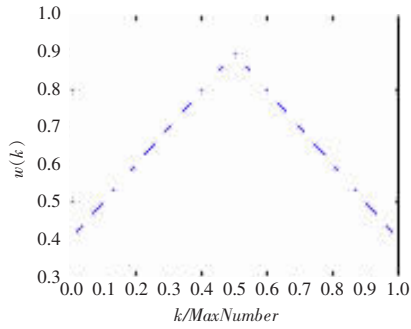


图2 本文提出的新惯性权重

(2) $f_2(x)$ 为 Rosenbrock 函数

$$f_2(x) = \sum_{i=1}^n \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right), x \in (-2.048, 2.048)^n$$

(3) $f_3(x)$ 为 Rastrigin 函数

$$f_3(x) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10), x \in (-5.12, 5.12)^n$$

(4) $f_4(x)$ 为 Griewank 函数

$$f_4(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, x \in (-600, 600)^n$$

同时,本文也采用文献[3]中提出的随机初始化时采用的不对称初始化区间方法,如表1所示。

实验中采用40个粒子,函数自变量的维数为20,最大迭代次数为1500次,对每一个测试函数, v_{\max} 的取值设为自变量 x 的上界,如表2所示。因为随机数的存在,本文所有实验都是在相同条件下执行50次,然后取平均值作为实验结果,如图3所示。

表1 不对称初始化区间

函数	不对称初始化区间
Sphere	(50, 100) ⁿ
Rosenbrock	(1.024, 2.048) ⁿ
Rastrigin	(2.56, 5.12) ⁿ
Griewank	(300, 600) ⁿ

表2 每个测试函数的 v_{\max}

函数	v_{\max}
Sphere	100
Rosenbrock	2.048
Rastrigin	5.12
Griewank	600

图3中,横轴表示迭代次数,纵轴表示50次的平均函数值,虚线是用递减惯性权重的PSO算法得到的曲线,实线是用惯性权重递增算法得到的曲线,点划线是本文算法得到的曲线。

从图3可以看出,在三种不同惯性权重的粒子群算法中,先增后减惯性权重的算法在前期阶段具有较快的收敛速度,尤其以Rastrigin函数 $f_3(x)$ 最为明显,在后期也具有较好的局部搜索能力,函数的最优值以Rosenbrock函数 $f_2(x)$ 和Griewank函数 $f_4(x)$ 较为明显。如表3所示。结果表明了本文提出的新粒子群算法的有效性和优越性,表3给出了4个标准测试函数在不同惯性权重下的平均最优值。

表3 不同函数在不同惯性权重下的平均最优值

函数	惯性权重		
	线性递减	线性递增	先增后减
Sphere 函数 $f_1(x)$ 最优值	9.96E-10	8.94E-14	5.663 3E-07
Rosenbrock 函数 $f_2(x)$ 最优值	16.798 4	6.248 1	0.932 9
Rastrigin 函数 $f_3(x)$ 最优值	15.267 7	21.267 7	19.663 9
Griewank 函数 $f_4(x)$ 最优值	0.035 4	0.032 8	0.019 3

从表3和图3综合来看,本文算法对 $f_2(x)$ 、 $f_4(x)$ 两个函数的最优值有了明显改善,对函数 $f_1(x)$ 的优化结果看起来似乎略差一些,但几种优化算法的结果都基本近似为0,其结果仅

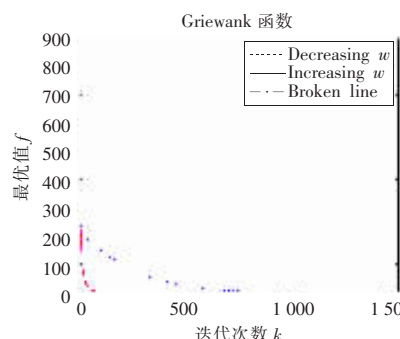
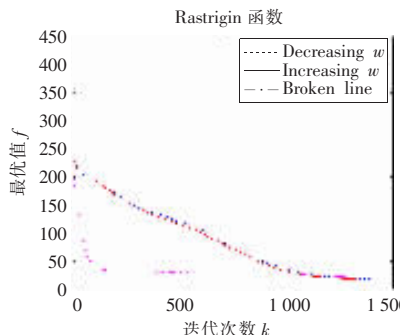
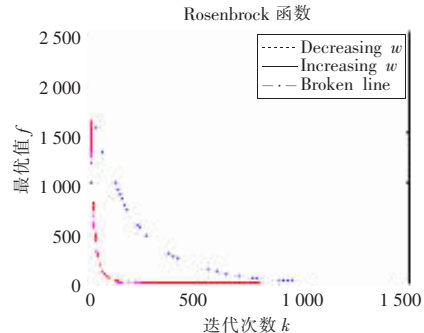
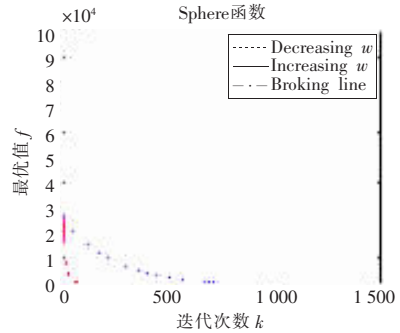


图3 各个测试函数在三种不同惯性权重条件下的测试结果

仅是7位小数以后的差异,因而基本没有影响;本文算法获得的 $f_3(x)$ 的最优值介于原来两种算法之间,但收敛性有了明显改善。说明了本算法具有一定的优越性。

5 结论

本文用实验方法分析了不同惯性权重 w 条件下微粒群算法粒子轨迹的收敛性。具有递减惯性权重的算法在前期阶段具有较好的全局搜索能力,后期收敛性也较好,但收敛速度比较慢;而具有递增惯性权重的算法虽然前期收敛速度很快,但后期的局部搜索能力较差。结合算法的收敛区间,本文提出了具有沿折线先增后减的动态惯性权重的微粒群算法,这样在算法