粒子群优化算法中惯性权重的研究进展

田雨波,朱人杰,薛权祥
TIAN Yu-bo,ZHU Ren-jie,XUE Quan-xiang

江苏科技大学 电子信息学院,江苏 镇江 212003
School of Electronics and Information,Jiangsu University of Science and Technology,Zhenjiang,Jiangsu 212003,China
E-mail:yubo.tian@163.com

TIAN Yu-bo,ZHU Ren-jie,XUE Quan-xiang.Research advances on inertia weight in particle swarm optimization.Computer Engineering and Applications,2008,44(23):39-41.

Abstract: Particle Swarm Optimization (PSO) is a novel stochastic optimization algorithm based on the simulation of migration and the group model of bird flock in the process of their food-searching,and it can be used to solve optimization problems.Inertia weight is an important parameter in PSO,and it can control the algorithm's exploitation ability and exploration ability.This paper simply introduces the principle of PSO,and overviews the research advances in the inertia weight.
Key words: Particle Swarm Optimization(PSO);inertia weight;optimization algorithm

摘　要:粒子群优化算法是根据鸟群觅食过程中的迁徙和群集模型而提出的用于解决优化问题的一类新兴的随机优化算法。惯性权重是粒子群算法中非常重要的参数,可以用来控制算法的开发和探索能力。简单介绍了标准粒子群优化算法的基本原理,全面综述了现有文献中对惯性权重的研究进展情况。
关键词:粒子群优化;惯性权重;优化算法
DOI:10.3778/j.issn.1002-8331.2008.23.012　　**文章编号:**1002-8331(2008)23-0039-03　　**文献标识码:**A　　**中图分类号:**TP301.6

1　引言

粒子群优化算法(Particle Swarm Optimization,PSO)是一种基于群体智能的进化计算(evolutionary computation)技术,其思想来源于人工生命和进化计算理论,最早是由美国的Kennedy和Eberhart教授受鸟群觅食行为的启发提出的[1]。PSO算法采用实数求解,并且需要调整的参数较少,易于实现,是一种通用的全局搜索算法。因此,算法一提出就得到众多学者的重视,并且已经在神经网络训练、函数优化和模糊系统控制等领域取得了大量的研究成果[2-3]。PSO的优势在于简单容易实现,同时又有深刻的智能背景,既适合科学研究,又特别适合工程应用。

惯性权重是PSO算法中非常重要的一个参数，可以用来控制算法的开发(exploitation)和探索(exploration)能力。惯性权重的大小决定了对粒子当前速度继承的多少:较大的惯性权重将使粒子具有较大的速度,从而有较强的探索能力;较小的惯性权重将使粒子具有较强的开发能力。关于惯性权重的选择一般有常数和时变两种,算法的执行效果很大程度上取决于惯性权重的选取,所以对惯性权重的研究就显得尤为重要。

2　标准PSO算法

PSO算法模拟鸟群的捕食行为。设想这样一个场景:一群鸟在随机搜索食物。在这个区域里只有一块食物,所有的鸟都不知道食物在那里，但是它们知道当前的位置离食物还有多远。那么找到食物的最优策略是什么呢?最简单有效的就是搜寻目前离食物最近的鸟的周围区域。PSO从这种模型中得到启示并用于解决优化问题。PSO中每个优化问题的解都是搜索空间中的一只鸟,称之为“粒子”。所有粒子都有一个由被优化函数决定的适应值(候选解)和一个决定它们飞翔方向与距离的速度。在优化过程中,每个粒子记忆、追随当前的最优粒子,在解空间中进行搜索。PSO算法初始化为一群随机粒子(随机候选解),然后通过迭代找到最优解。在每一次迭代过程中,粒子通过追逐两个极值来更新自己的位置:一个是粒子自身所找到的当前最优解,这个解称为个体极值*pbest*;另一个是整个群体当前找到的最优解,这个解称为全局极值*gbest*。

粒子在找到上述两个极值后,就根据式(1)、式(2)来更新自己的速度与位置:

$$v_{i,d}^{k+1}=v_{i,d}^{k}+c_1\cdot rand()\cdot(pbest_{i,d}^{k}-x_{i,d}^{k})+c_2\cdot rand()\cdot(gbest_{d}^{k}-x_{i,d}^{k}) \quad (1)$$

$$x_{i,d}^{k+1}=x_{i,d}^{k}+v_{i,d}^{k+1} \quad (2)$$

式中c_1和c_2被称为学习因子,$rand()$为介于(0,1)之间的随机数。$v_{i,d}^{k}$和$x_{i,d}^{k}$分别为粒子i在第k次迭代中第d维的速度和位置,两者均被限制在一定的范围内,$pbest_{i,d}^{k}$是粒子i在第d维的个体极值的位置,$gbest_{d}^{k}$是群体在第d维的全局极值的位置。

基金项目:江苏省高校自然科学基础研究项目(No.07KJB510032)。
作者简介:田雨波(1971-),男,博士,副教授,硕士生导师,主要研究方向为计算智能应用于电子学与电磁学问题。
收稿日期:2008-03-03　　**修回日期:**2008-04-24

文献[4]为了更好的控制粒子群算法的开发和探测能力，将惯性权重 ω 引入到式（1）中，形成现在标准的 PSO 算法：

$$v_{i,d}^{k+1}=\omega\cdot v_{i,d}^{k}+c_1\cdot rand()\cdot(pbest_{i,d}^{k}-x_{i,d}^{k})+c_2\cdot rand()\cdot(gbest_{d}^{k}-x_{i,d}^{k}) \quad (3)$$

并通过实验研究了惯性权重 ω 对算法性能的影响，发现较大的 ω 值有利于跳出局部最优，进行全局寻优；而较小的 ω 值有利于局部寻优，加速算法收敛。

3 惯性权重对标准 PSO 算法性能之影响

由式（3）知道，PSO 算法中有三个参数可供调整和设置：惯性权重 ω 与学习因子 c_1 和 c_2。惯性权重 ω 的大小决定了对粒子当前速度继承的多少，合适的选择可以使粒子具有均衡的探索和开发能力；学习因子 c_1 和 c_2 为正常数，它们使粒子具有自我总结和向群体中优秀个体学习的能力，从而向自己的历史最优点以及群体内或邻域内历史最优点靠近。可见，惯性权重作为 PSO 算法中一个重要的参数，为了保持粒子具有均衡的探索和开发能力，必须进行合理的选择。惯性权重的选择主要分为两类：固定权重和时变权重。固定权重的选择就是选择某一常数作为权重值，在优化过程中不变；而时变权重则是选定某一个变化范围，在迭代过程中按照某一规律变化。固定的惯性权重使粒子始终具有相同的探索和开发能力，时变权重使粒子可以在优化的不同时期具有不同的探索和开发能力。

关于惯性权重如何选取，已经有很多文章加以讨论。文献[5-8]分别就固定的惯性权重加以阐述并进行优化，得出一些有益的结论：文献[6]提出当 $\omega=0.6$，$c_1=c_2=1.7$ 时 PSO 算法的性能最佳；文献[7]得出的结论为 $\omega=0.729$，$c_1=c_2=1.494$；文献[8]应用遗传算法对标准 PSO 算法的三个参数进行了优化，并就 10 位的 Rosenbrock 函数和 Griewank 函数进行了仿真，得出结论，对于 Rosenbrock 函数，$\omega=0.659$，$c_1=2.796$，$c_2=1.336$，而对于 Griewank 函数，$\omega=0.613$，$c_1=2.456$，$c_2=2.260$。

关于时变的惯性权重，有更多的文献加以讨论，本文也主要给出现有文献中时变的惯性权重的研究情况。

文献[9]提出了一种递减惯性权重的 PSO 算法，令式（3）中 $c_1r_1=\varphi_1$，$c_2r_2=\varphi_2$，$\varphi_1\in[0,2]$，$\varphi_2\in[0,2]$，与标准算法的主要不同点在于：惯性权重 ω 是迭代次数的函数，沿直线从 0.9 线性递减到 0.4，即 ω 满足：

$$\omega(k)=-0.5\times\frac{k}{MaxNumber}+0.9 \quad (4)$$

其中 $MaxNumber$ 为最大迭代次数。

文献[10]提出了一种递增惯性权重的 PSO 算法，令式（1）中 $c_1r_1=\varphi_1$，$c_2r_2=\varphi_2$，$\varphi_1\in[0.5,2]$，$\varphi_2\in[0.5,2]$，$\varphi_i=0.5+1.5\times r_i$，$i=1,2$，$r_i$ 为（0，1）之间的随机数。惯性权重 ω 是迭代次数的函数，沿直线从 0.4 线性递增到 0.9，即 ω 满足：

$$\omega(k)=-0.5\times\frac{k}{MaxNumber}+0.4 \quad (5)$$

文献[11]提出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。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，即惯性权重 ω 沿折线从 0.4 线性递增到 0.9，然后再线性递减到 0.4，即ω 满足：

$$\omega(k)=\begin{cases}1\times\dfrac{k}{MaxNumber}+0.4 & 0\leqslant\dfrac{k}{MaxNumber}\leqslant 0.5\\ -1\times\dfrac{k}{MaxNumber}+1.4 & 0.5\leqslant\dfrac{k}{MaxNumber}\leqslant 1\end{cases} \quad (6)$$

文献[12]详细讨论了递减惯性权重与粒子群规模、种群拓扑结构、求解问题的依赖性和初始权重及终止权重等之间的关系，经过仿真后得到一个相对普遍的结论：当惯性权重的初始值为 0.5、终止值为 0.2 且按照（0.5~0.2）/1 000 时收敛速度和达优率之间达到一种平衡。另外文献[13]也讨论了递减惯性权重对 PSO 性能的影响，在算法实现过程中引入了实验设计的方法。

文献[9-12]给出的都是线性规律变化的惯性权重，也有相当的文献讨论了非线性规律变化的惯性权重。文献[14]引入递减指数和迭代阈值对基本粒子群算法中线性递减权策略进行了改进，在优化迭代过程中，惯性权重随当前迭代次数、指数递减率和迭代阈值非线性变化，即：

$$\omega(k)=\left(\frac{k-1}{K-1}\right)^{\lambda}(\omega_f-\omega_i)+\omega_i \quad (7)$$

式中 λ 为递减指数，K 为迭代阈值，ω_i 为初始惯性权值，ω_f 为达到阈值时的惯性权值。当迭代次数达到 K 时令 $\omega(k)=\omega_f$ 并将保持到搜索结束。在整个迭代过程中因为 λ 的引入，$\omega(k)$ 随着迭代次数的增加而非线性地递减，有利于跳出局部极值点。迭代初期 ω 较大，粒子以较快的飞行速度遍布整个搜索空间从而确定最优值的大致范围，随着迭代的进行 ω 非线性减小，大部分粒子的搜索范围逐渐减小，并且集中在最优值的邻域范围内；在迭代末期当达到迭代阈值时将惯性权重限定为 ω_f，粒子以近乎不变的飞行速度在最优值邻域范围内找到全局最优值，有利于提高算法的收敛速度。文献[15]也讨论了一种非线性动态调整惯性权重的方法，并和标准 PSO 算法及其他算法进行了仿真比较。

文献[16]采用一种基于目标函数信息而变化的惯性权重动态调整方法，使搜索方向的启发性增强。即 ω 按照下式变化：

$$\begin{aligned}&\omega(k)=e^{-\alpha(k)/\alpha(k-1)}\\&\alpha(k)=\frac{1}{m}\sum_{i=1}^{m}|f(X_i(k))-f(X_{\min}(k))| \quad k=0,1,2,\cdots\\&f(X_i(k))=f(x_{i,1}(k),x_{i,2}(k),\cdots,x_{i,D}(k))\\&f(X_{\min}(k))=\min_{i=1,2,\cdots,m}f(X_i(k))\end{aligned} \quad (8)$$

式中 $f(X_i(k))$ 为第 i 个粒子在第 k 次迭代是对应的函数值，$f(X_{\min}(k))$ 为最优粒子在第 k 次迭代是对应的函数值。计算 $\alpha(k)$ 指标是用来判断目标函数的平整度，如果 $\alpha(k)$ 较大，则目标函数的平整性较差。每次迭代 $\alpha(k)$ 时指标都根据所得函数值进行变化，这样使原来随着搜索过程线性减小的 ω 变成随搜索位置改变而动态改变的 ω。当 $\alpha(k)/\alpha(k-1)<1$ 时，说明此次迭代总体收敛，$\alpha(k)/\alpha(k-1)$ 比值越小，$\omega(k)$ 越接近 1，搜索步长越大；当 $\alpha(k)/\alpha(k-1)>1$ 时，说明此次迭代总体发散，$\alpha(k)/\alpha(k-1)$ 比值越大，$\omega(k)$ 越接近 0，搜索步长越小。这使得迭代的步长不是一味地减小，而是根据具体目标函数的计算结果来确定。$\alpha(k)$ 的减小幅度越快，说明趋向极值点的速度越快，此时 ω 也就较大，便于保持大步长的搜索；当接近极值点时，$\alpha(k)$ 每次的变化减小，此时 ω 也就较小，便于在极值点附近作小步长搜索。

文献[17]提出了一种根据不同粒子距离全局最优点的距离

对基本PSO算法的惯性权重进行动态调整的新型粒子群算法，惯性权重按照下式动态调整：

$$\omega(k)=\begin{cases}\omega_{\max} & l_{ig}>l_{\max}\\ \omega_{\max}-\dfrac{(l_{ig}-l_{\min})}{(l_{\max}-l_{\min})}\dfrac{k}{MaxNumber}(\omega_{\max}-\omega_{\min}) & l_{\min}<l_{ig}<l_{\max}\\ \omega_{\min} & l_{ig}<l_{\min}\end{cases}\quad(9)$$

式中l_{ig}表示第i个粒子与全局最优粒子的距离，$l_{\max}$和$l_{\min}$表示最大距离和最小距离。由于参数$l_{\max}$和$l_{\min}$随着问题不同取值不一样的，且本文没有给出理论指导，所以增加了算法求解最优的难度。

文献[18]根据群体早熟收敛程度和个体适应值自适应地调整粒子的惯性权重，使群体在进化过程中始终保持惯性权重的多样性，在算法的全局收敛性和收敛速度之间做了一个折衷，即惯性权重按照下式动态调整：

$$\omega(k)=\begin{cases}\omega(k)-(\omega(k)-\omega_{\min})\cdot\left|\dfrac{f_i-f'_{avg}}{f_m-f'_{avg}}\right| & f_i>f'_{avg}\\ \omega(k) & f_{avg}<f_i<f'_{avg}\\ 1.5-\dfrac{1}{1+k_1\exp(-k_2\Delta)} & f_i<f_{avg}\end{cases}\quad(10)$$

式中f_i为第k次迭代中第i个粒子的适应值，f_{avg}为粒子的平均适应值，$f_{avg}=\dfrac{1}{m}\sum_{i=1}^{m}f_i$，$m$为粒子群的大小，$f'_{avg}$为适应值优于$f_{avg}$的适应值求平均得到的值，$\Delta=|f_m-f'_{avg}|$，$f_m$为最优粒子的适应值，$\Delta$用来评价粒子群的早熟收敛程度，且$\Delta$越小说明粒子群越趋于早熟收敛。$k_1$、$k_2$两个参数的选择对算法性能的影响较大，文献[18]中k_1=1.5、k_2=0.3。

文献[19]采用（0，1）均匀分布的随机惯性权重法来取代线性递减的方法，使得微粒能在运行初期有机会获得较小的ω值，也能够在后期有机会得到较大的ω值。但是当全局最好值$gbest$未发生变化时，希望随机取得的ω值较大，加大搜索力度，由于均匀分布的随机ω在（0，1）之间取值的概率是相同的，取的比较小值与比较大值的机会是均等的，并不能有更多的机会取得较大的值，如果取得较小的ω值，则可能陷入局部最优。根据以上对ω的分析，均匀分布的随机ω取值应该根据$gbest$是否发生变化，随机取值的范围不同。分两种情况：当$gbest$不变时，ω随机取一个较大的值，否则ω随机选取的值既可以小，也可以大，表示如下：

$$\text{if}\quad \Delta gbest=0\quad \omega=rand(r_1,r_2)\quad \text{else}\quad \omega=rand(r_3,r_4)\quad(11)$$

其中$r_3<r_2$。

文献[20]将混沌优化机制引入到PSO算法中，提出混沌递减惯性权重和混沌随机惯性权重，从而使PSO算法的收敛精度、收敛速度以及全局收敛性达到一种很好的平衡。

4 结论

惯性权重是粒子群算法中一个非常重要的参数，可以用来控制算法的开发能力和探索能力，本文对粒子群算法中的惯性权重的研究进展作了全方位的讨论。综合现有文献可以看出，粒子群算法中的惯性权重可以从不同的角度分成不同的类型，如固定权重和时变权重、线性变化权重和非线性变化权重、随机权重和非随机权重等。现有文献对惯性权重的研究大都引入了一些参数，且往往这些参数具有问题依赖性，缺乏理论指导。可见，如何对粒子群算法中的惯性权重加以理论分析并对其选择加以理论指导必将成为今后研究的一个重点。

参考文献：

[1] Kennedy J，Eberhart R C.Particle swarm optimization[C]//IEEE International Conference on Neural Networks.Piscataway，NJ：IEEE Press，1995：1942-1948.

[2] Clerc M.Particle swarm optimization[M].London：ISTE Publishing Company，2006.

[3] 曾建潮，介婧，崔志华.微粒群算法[M].北京：科学出版社，2004.

[4] Shi Y，Eberhart R C.A modified particle swarm optimizer[C]//Proceedings of the IEEE Conference on Evolutionary Computation. Piscataway，NJ：IEEE Press，1998：69-73.

[5] Clerc M，Kennedy J.The particle swarm：explosion，stability，and convergence in a multi-dimensional complex space[J].IEEE Transactions on Evolutionary Computation，2002，6（1）：58-73.

[6] Trelea I.The particle swarm optimization algorithm：convergence analysis and parameter selection [J].Information Processing Letters，2003，85（6）：317-325.

[7] Eberhart R，Shi Y.Comparing inertia weights and constriction factors in particle swarm optimization[C]//IEEE Congress on Evolutionary Computation.Piscataway：IEEE Service Center，2000：84-88.

[8] 张丽平.微粒群算法的理论与实践[D].杭州：浙江大学，2005.

[9] Shi Y，Eberhart R.Empirical study of particle swarm optimization[C]//Proceedings of the 1999 Congress on Evolutionary Computation，1999：1945-1950.

[10] Zheng Yong-ling，Ma Long-hua，Zhang Li-yan，et al.On the convergence analysis and parameter selection in particle swarm optimization[C]//Proceedings of the Second International Conference on Machine Learning and Cybernetics，Xi'an，2003：1802-1807.

[11] 崔红梅，朱庆保.微粒群算法的参数选择及收敛性分析[J].计算机工程与应用，2007，43（23）：89-91.

[12] 王俊伟，汪定伟.粒子群算法中惯性权重的实验与分析[J].系统工程学报，2005，20（2）：194-198.

[13] Fan Shu-kai，Chiu Yi-yin.A decreasing inertia weight particle swarm optimizer[J].Engineering Optimization，2007，39（2）：203-228.

[14] 王丽，王晓凯.一种非线性改变惯性权重的粒子群算法[J].计算机工程与应用，2007，43（4）：47-48.

[15] Chatterjee A，Siarry P.Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization[J].Computers and Operations Research，2006，33（3）：859-871.

[16] 王启付，王战江，王书亭.一种动态改变惯性权重的粒子群优化算法[J].中国机械工程，2005，16（11）：945-948.

[17] 刘建华，樊晓平，瞿志华.一种惯性权重动态调整的新型粒子群算法[J].计算机工程与应用，2007，43（7）：68-70.

[18] 韩江洪，李正荣，魏振春.一种自适应粒子群优化算法及其仿真研究[J].系统仿真学报，2006，18（10）：2969-2971.

[19] 延丽平，曾建潮.具有自适应随机惯性权重的PSO算法[J].计算机工程与设计，2006，27（4）：4677-4679.

[20] Feng Yong，Yao Yong-mei，Wang Ai-xin，Comparing with chaotic inertia weights in particle swarm optimization[C]//2007 International Conference on Machine Learning and Cybernetics，Hong Kong，China，2007：329-333.