

一种基于距离度量的自适应粒子群优化算法

李太勇^{1,2} 吴 江^{1,2} 朱 波³ 方 冰⁴

(西南财经大学经济信息工程学院 成都 610074)¹ (西南财经大学中国支付体系研究中心 成都 610074)²
(西南财经大学金融学院 成都 610074)³ (贵阳学院计算机科学系 贵阳 550005)⁴

摘 要 惯性权值对粒子群优化(Particle Swarm Optimization, PSO)算法的性能起着重要作用。基本的 PSO 算法未考虑各粒子的差异而在一次迭代中所有粒子采用固定的惯性权值。为了体现各粒子相对于已知最优解的差异,提出了一种基于距离度量的自适应 PSO 算法 DMAPSO(Distance Measurement-based Adaptive PSO)。算法采用欧式距离计算粒子与已知全局最优粒子的差异,然后根据差异自适应调整各粒子的惯性权值。通过基准测试函数对算法进行了实验,结果表明,对于连续函数优化问题,提出的 DMAPSO 算法优于经典 PSO 算法,DMAPSO 收敛到最优解的迭代次数比 PSO 平均减少了约 60%。

关键词 粒子群,优化算法,惯性权值,距离度量

中图法分类号 TP18 **文献标识码** A

Distance Measurement Based Adaptive Particle Swarm Optimization

LI Tai-yong^{1,2} WU Jiang^{1,2} ZHU Bo³ FANG Bing⁴

(School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu 610074, China)¹

(The Research Center for China Payment System, Southwestern University of Finance and Economics, Chengdu 610074, China)²

(School of Finance, Southwestern University of Finance and Economics, Chengdu 610074, China)³

(Department of Computer Science, Guiyang University, Guiyang 550005, China)⁴

Abstract The inertia weight plays an important role in Particle Swarm Optimization(PSO). The classical PSO used a fixed inertia weight for all particles in an iteration and ignored the difference among the particles. To cope with this issue, a Distance Measurement based Adaptive Particle Swarm Optimization(DMAPSO) was proposed. The Euclidean distance was used to calculate the difference between a particle and the known best global particle, and the particle tuned adaptively the value of the inertia weight according to the difference. Several classical benchmark functions were used to evaluate the strategy. The experimental results show that for continuous optimization problems, the DMAPSO outperforms the classical PSO. The iteration times for finding the best solutions in the DMAPSO decrease about 60% averagely compared with that in the classical PSO.

Keywords PSO, Optimization algorithm, Inertia weight, Distance measurement

美国心理学家 Kennedy 和电气工程师 Eberhart 受鸟类觅食行为的启发,于 1995 年提出了粒子群优化算法(Particle Swarm Optimization, PSO)^[1,2]。PSO 由于具有算法简单、容易实现和参数较少等优点而受到了学术界的广泛关注。随着 PSO 研究的深入,它在化工、电力、通信、经济、机械设计、生物信息等领域获得了广泛应用^[3]。

惯性权值是 PSO 中非常重要的一个参数,它直接影响着算法的收敛速度,是影响算法搜索行为和性能的关键所在。文献[4]的研究表明,较大的惯性权值有利于跳出局部极小点,便于全局搜索;而较小的惯性权值则有利于对当前搜索区域进行精确局部搜索,便于算法收敛。因此,在粒子群的迭代

过程中,有必要找到一种策略来调整惯性权值,使算法能在全局搜索和局部搜索性能上达到较好的平衡。基于这种考虑,文献[4]提出了惯性权值线性递减(Linearly Decreasing Inertia Weight, LDIW) PSO 算法,从而提高了算法性能,其缺点是不能很好适应复杂优化问题;文献[5]提出了模糊规则惯性权值(Fuzzy Inertia Weight, FIW) PSO 算法,其缺点是计算复杂,效率低。文献[3]中提出了一种非线性动态适应度调节惯性权值的方法(Adaptive Inertia Weight Factor, AIWF),它使惯性权值随粒子目标值的变化而自动变化,其缺点是未考虑粒子之间的位置关系。更多的相关研究参见文献[6-9]。

LDIW 算法中,惯性权值与迭代次数之间存在线性关系,

到稿日期:2009-11-12 返修日期:2010-01-26 本文受国家自然科学基金(60773169),西南财经大学金融智能与金融工程重点实验室校内公开项目(FIFE2010-P02)资助。

李太勇(1979—),男,博士,讲师,CCF 会员,主要研究方向为数据库与知识工程,E-mail:litaiyong@gmail.com;吴 江(1980—),男,博士,讲师,主要研究方向为数据库与知识工程;朱 波(1977—),男,副教授,主要研究方向为金融资产定价和实证金融;方 冰(1979—),男,硕士,讲师,主要研究方向为智能优化。

这与大多数实际问题不符,且粒子群中所有粒子在一次迭代中采用固定的惯性权值,未考虑粒子间的差异。基于此,本文将粒子的距离概念引入到算法中,采用欧式距离公式度量粒子间的差异,然后根据差异的不同自适应调整惯性权值,提出了一种基于距离度量的自适应 PSO 算法 DMAPSO(Distance Measurement based Adaptive PSO),以此来探索惯性权值与 PSO 进化效率之间的联系,并通过实验验证了算法的有效性。

1 粒子群优化算法原理

粒子群优化算法随机初始化一群 d 维粒子,第 i 个粒子通过速度 $V_i=[v_{i1}, v_{i2}, \dots, v_{id}]$ 和位置 $X_i=[x_{i1}, x_{i2}, \dots, x_{id}]$ 来描述。然后群体通过迭代过程找到最优解,在迭代过程中,通过评价 $P_g=[p_{g1}, p_{g2}, \dots, p_{gd}]$,再按照式(1)和式(2)来分别更新粒子的速度和位置^[1-3]。

根据粒子的目标函数,确定第 i 个粒子已经取得的最佳位置 $P_i=[p_{i1}, p_{i2}, \dots, p_{id}]$ 和整个群体已经取得的最佳位置。

$$v_{ij}(t+1)=wv_{ij}(t)+c_1r_1(p_{ij}-x_{ij}(t))+c_2r_2(p_{gj}-x_{ij}(t)) \quad (1)$$

$$x_{ij}(t+1)=x_{ij}(t)+v_{ij}(t+1) \quad (2)$$

式中, t 为迭代的次数, w 为惯性权, c_1 和 c_2 为正的加速常数, r_1 和 r_2 为在 $(0,1)$ 上均匀分布的随机数^[3]。

式(1)和式(2)是粒子群优化算法的主要内容。研究表明, w 值的调整策略对算法的性能有重要影响。较大的 w 值便于全局搜索,而较小的 w 值便于局部搜索。显然,迭代过程中一成不变的 w 值将导致算法效率低下。理想的迭代算法会呈现这样的特点:在迭代初期,群体具有较强的全局搜索能力;而在迭代后期,群体具有较强的局部搜索能力。在迭代初期,应具有较大的 w 值,而在迭代后期,应具有较小的 w 值。基于这样的考虑,文献[4]提出了 LDIW 策略,对 w 值进行动态调整,其策略如式(3)所示。

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

式中, $w_{\max}$ 和 $w_{\min}$ 是凭经验设置的惯性权值的最大值和最小值, t 为当前迭代次数, G 为最大迭代次数。研究表明,对于大多数优化问题,当 $w_{\max}$ 和 $w_{\min}$ 分别取 0.9 和 0.4 时, PSO 算法具有较高性能。从式(3)可见,随着迭代的进行, w 值线性降低。LDIW 策略简单,易于实现,是被广泛采用的惯性权值调整策略。LDIW 是一种典型的线性递减策略,其缺点是与大多数实际问题不符,且粒子群中所有粒子在一次迭代中采用固定的值,未考虑粒子间的差异。本文将根据群体中各粒子的差异来自适应调整粒子的惯性权值,以期提高 PSO 的优化能力。

2 基于距离度量的自适应粒子群优化算法

2.1 距离度量方法

为了度量粒子在自变量上和目标值上的差异,文中采用最常用的欧式距离公式来度量两个粒子之间的距离。

定义 1(粒子间的距离) 粒子 i 和粒子 j 之间的距离定义为:

$$\text{dist}(i,j)=\sqrt{\sum_{k=1}^d(x_{ik}-x_{jk})^2+(f(X_i)-f(X_j))^2} \quad (4)$$

式中,第 i 个粒子表示为 $X_i=[x_{i1}, x_{i2}, \dots, x_{id}]$, d 为粒子的维

数, f 为目标函数的表达式。与一般的距离公式不同,在定义 1 中,既考虑了自变量之间的距离,又考虑了目标函数的距离。

根据定义 1,对于任意粒子 i,j,k ,有如下性质:

性质 1 $\text{dist}(i,j) \geq 0$;

性质 2 $\text{dist}(i,i)=0$;

性质 3 $\text{dist}(i,j)=\text{dist}(j,i)$;

性质 4 $\text{dist}(i,j) \leq \text{dist}(i,k)+\text{dist}(k,j)$ 。

由于性质 1—性质 4 相对简单,为节省篇幅,证明从略。

定义 2(粒子间平均距离) 群体中粒子间的平均距离定义为:

$$\text{Mdist}=\sum_{i=1}^N\sum_{j=1}^N\text{dist}(i,j)/N^2 \quad (5)$$

式中, N 为群体规模,即群体中粒子数目。根据性质 2 和性质 3,实际在计算平均距离时,只需考虑 $i < j$ 时的距离情况即可,如式(6)所示。

$$\text{Mdist}=2\sum_{i=1}^{N-1}\sum_{j=i+1}^N\text{dist}(i,j)/N^2 \quad (6)$$

2.2 自适应粒子群优化算法

较大的惯性权值有利于跳出局部极小点,便于全局搜索;而较小的惯性权值则有利于对当前搜索区域进行精确局部搜索,便于算法收敛。因此,在自适应粒子群优化算法中,将根据粒子与最优粒子之间的距离来动态调整惯性权值:如果粒子与最优粒子之间的距离大于某个阈值 T ,则该粒子分配较大的惯性权值,便于全局搜索;否则,该粒子分配较小的惯性权值,便于粒子进行局部求精,提高优化精度。

本文将粒子间平均距离 Mdist 作为上述阈值,粒子 X 的自适应惯性权值的计算表达式如式(7)所示。

$$w=\begin{cases} w_{\max} & \text{dist}(X,p_g) > \text{Mdist} \\ w_{\min}+(w_{\max}-w_{\min})\text{dist}(X,p_g)/\text{Mdist} & \text{otherwise} \end{cases} \quad (7)$$

式中, p_g 是群体中已计算出的全局最优粒子, $w_{\max}$ 和 $w_{\min}$ 分别代表 w 的最大值和最小值。

采用上述自适应惯性权值调整策略后,粒子群优化算法的算法框架如下:

算法 1 基于距离度量的自适应 PSO 算法 DMAPSO

输入:最大迭代次数 G ,种群规模 N , $w_{\max}$, $w_{\min}$, c_1 , c_2

输出:全局最优解 P_g

Begin

1. $t=1$;

2. 建立初始化种群,计算各粒子最优个体 P_i 和群体最优粒子 P_g ;

3. while($t \leq G$) {

4. 按式(6)计算群体中粒子间的平均距离;

5. 按式(7)计算群体中各粒子的惯性权值;

6. 按式(1)一式(2)更新各粒子的速度和位置;

7. 更新 P_i 和 P_g ;

8. $t=t+1$;

9. 输出 P_g ;

End

算法 1 与基本粒子群算法的区别在于更新惯性权值的方式不同:基本粒子群优化算法根据当前的迭代次数来线性调整惯性权值,而算法 1 根据粒子与最优粒子之间的距离自适

应地调整惯性权值。因此,算法1中考虑了粒子间的差异性,而基本粒子群算法忽略了该差异。

3 仿真实验

3.1 实验设置

采用4个著名的测试函数来测试本文算法^[3],它们在文献中被广泛用于 PSO 算法的性能测试,其具体函数形式如下:

(1) Branin 函数

$$F_1(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$$

$$-5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15 \quad (8)$$

(2) Rastrigin 函数

$$F_2(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2) \quad -1 \leq x_1, x_2 \leq 1 \quad (9)$$

(3) Shubert 函数

$$F_3(x) = \{\sum_{i=1}^5 i \cos[(i+1)x_1 + i]\} \{\sum_{i=1}^5 i \cos[(i+1)x_2 + i]\}$$

$$-10 \leq x_1, x_2 \leq 10 \quad (10)$$

(4) Six-hump Camel Back 函数

$$F_4(x) = (4 - 2.1x_1^2 + x_1^4/3)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$

$$-1.9 \leq x_1 \leq 1.9, -1.1 \leq x_2 \leq 1.1 \quad (11)$$

为了与基本粒子群优化算法 PSO 的性能进行对比,所有测试函数均分别采用 PSO 和 DMAPSO 进行实验。实验参数设置为:种群大小设置为 20, c_1 和 c_2 设置为 2, w_{max} 和 w_{min} 分别设置为 0.9 和 0.4,迭代次数为 200。

采用 Matlab 7 进行仿真,系统软硬件环境为:P4 2.0G, 512M 内存, Windows XP。

3.2 实验结果

所有实验均重复执行 20 次,假设 20 次执行得到的最优目标值的算术平均值为 T' ,理想目标值为 T 。对全局最小化问题,采用 $\lg(T'-T)$ 来评价算法的性能, $\lg(T'-T)$ 反映了 T' 和 T 的逼近程度。该值越小,表明 T' 越逼近 T 。除此之外, $\lg(T'-T)$ 还能直观反映各算法的收敛精度的数量级关系。在求解精度上, T' 和 T 的值保留到小数点后 8 位,当 $\lg(T'-T)$ 值为 -8 时,表明算法找到了理想目标值。

F1—F4 函数的优化结果如图 1—图 4 所示。

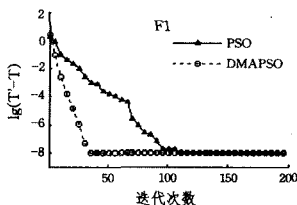


图 1 优化 F1 函数的算法比较

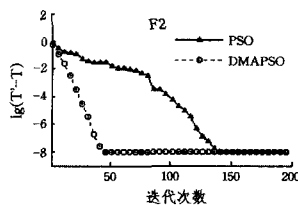


图 2 优化 F2 函数的算法比较

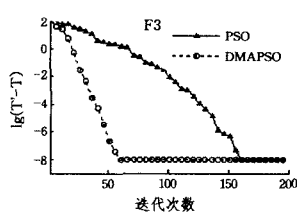


图 3 优化 F3 函数的算法比较

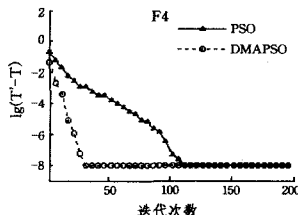


图 4 优化 F4 函数的算法比较

从图 1—图 4 可以看出,对于给定的 4 个测试函数,各方法都能在指定的迭代次数内找到最优解,但各方法的收敛效果有所不同。

在给定的 4 个基准测试函数上, DMAPSO 均早于基本 PSO 收敛。平均地,相对于 PSO, DMAPSO 收敛到最优解的迭代次数减少了 60%。

实验验证了 DMAPSO 算法的有效性。这得益于 DMAPSO 的如下特点:(1)惯性权值与迭代次数无关,仅与粒子本身特性有关,因此,在一次迭代中,不同的粒子采用不同的惯性权值;(2)根据粒子与最优粒子的距离自适应调整粒子的惯性权值。当距离远时,粒子具有较大的惯性权值,因此具有较强的全局优化能力;当距离近时,粒子具有较小的惯性权值,因此具有较强的局部优化能力。

结束语 粒子群优化算法是当前的研究热点,其中惯性权值的调整直接影响到算法的性能。本文引入了粒子间的距离计算方法,通过计算粒子与最优粒子之间的距离自适应地调整惯性权值,提出了 DMAPSO 算法,使远离最优粒子的粒子具有较强的全局优化能力,而靠近最优粒子的粒子具有较强的局部优化能力,并通过实验验证了算法的性能。实验表明,在连续函数的优化过程中, DMAPSO 的性能远优于基本 PSO。下一步将研究 DMAPSO 在函数优化以外的应用以及惯性权值的其他自适应调整方法。

参考文献

- [1] Kennedy J, Eberhart R. Particle swarm optimization[C]// Proceedings of International Conference on Neural Networks, Piscataway, NJ: IEEE Service Center, 1995: 1942-1948
- [2] Kennedy J, Eberhart R, Shi Y. Swarm Intelligence [M]. San Francisco: Morgan Kaufmann Publishers, 2001
- [3] 王凌, 刘波. 微粒群优化与调度算法[M]. 北京: 清华大学出版社, 2008
- [4] Shi Y, Eberhart R. A modified particle swarm optimizer[C]// Proceedings of the IEEE International Conference on Evolutionary Computation, Piscataway, NJ: IEEE Press, 1998: 69-73
- [5] Shi Y, Eberhart R. Particle Swarm Optimization with Fuzzy Adaptive Inertia Weight[C]// Proceedings of the Workshop on Particle Swarm Optimization, Indianapolis, IN: Purdue School of Engineering and Technology, 2001: 101-106
- [6] Eberhart R, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization[C]// Proceedings of Congress on Evolutionary Computation 2000, San Diego, CA, 2000: 84-88
- [7] 张选平, 杜玉平, 秦国强, 等. 一种动态改变惯性权的自适应粒子群算法[J]. 西安交通大学学报, 2005, 39(10): 1039-1042
- [8] 陈贵敏, 贾建援, 韩琪. 粒子群优化算法的惯性权值递减策略研究[J]. 西安交通大学学报, 2006, 40(1): 53-56, 61
- [9] Shi Y, Eberhart R. Parameter selection in particle swarm optimization[C]// Proceedings of the 1998 Annual Conference on Evolutionary Computation, 1998: 591-600