一种带自适应飞行时间因子的粒子群算法

赵茂先,李小丹
(山东科技大学 信息科学与工程学院,山东 青岛 266590)

摘　要:作为一种新型智能算法,粒子群算法具有概念简单、易于实现等特点,但也存在容易陷入局部最优的缺点。为了尽可能找到问题的最优解,提高粒子群算法的收敛速度,提出一种带自适应飞行时间因子的粒子群算法,在算法中引入种群多样性和种群进化度两个参数,并根据这两个参数对算法性能的影响,让飞行时间因子随着这两个参数自适应改变。通过对 4 个基准函数的测试表明,改进后的粒子群算法较其他几种粒子群算法在收敛速度和收敛精度上都有一定提高。

关键词:粒子群算法;飞行时间因子;自适应;智能算法;最优化

中图分类号:TP181　　**文献标志码**:A　　**文章编号**:1672-3767(2014)02-0081-05

Particle Swarm Optimization with Adaptive Flying Time Factor

Zhao Maoxian, Li Xiaodan
(College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong 266590, China)

Abstract: As a new intelligence algorithm, particle swarm optimization with advantages of simple conception and easy implement also suffers the high risk of trapping in a local optimum. To find the optimal solution, a particle swarm optimization algorithm with adaptive flying time factor was presented. Two parameters of species diversity and population evolution degree were introduced, and the flying time factor changed with these two parameters adaptively. The test of the four benchmark functions shows that the modified particle swarm optimization algorithm is better in convergence speed and convergence accuracy compared with other kinds of particle swarm optimization algorithms.

Key words: particle swarm optimization; flying time factor; adaptive; intelligence algorithm; optimization

粒子群算法(particle swarm optimization,PSO)于 1995 年由 Eberhart 和 Kennedy 提出[1],是一种基于群体的、模拟鸟群社会行为的随机搜索过程。在粒子群算法中,种群保持一定数量,群体中每个粒子代表着优化问题的一个潜在解,整个群体代表了优化问题的若干潜在解。粒子群算法的目的是寻找一个粒子的位置,使得目标函数达到最优。

基本粒子群算法存在后期收敛速度慢及容易陷入局部极值的缺点,为了提高算法的收敛速度或收敛精度,许多学者对基本粒子群算法的参数进行不同方式的选择与优化,提出很多改进方法。1998 年,Shi 等[2]提出带惯性权重的标准粒子群算法(standard PSO,SPSO),在此基础上又有很多学者提出惯性权重自适应改变的 PSO 算法,如惯性权重随迭代线性下降[2]、按非线性函数下降[3-5]、随着某种评价指标自适应变化[6-8]。1999 年,Clerc[9]提出带收缩因子的 PSO 算法,该算法可以保证 PSO 算法收敛。2006 年,张建科等[10]提出带飞行时间因子的粒子群算法,给出 3 种飞行时间因子调整方案。2011 年,周喜虎等[11]提出让飞行时间因子随着种群多样性改变的 PSO 算法。上述方法均或多或少提高了算法收敛速度或收敛精度,但仍

收稿日期:2013-12-13
基金项目:国家自然科学基金项目(61370207)
作者简介:赵茂先(1966—),男,江苏江都人,教授,主要从事最优化理论、方法及相关问题的研究. E-mail:sdzmx66@sohu.com

然存在高维问题下收敛速度慢或不稳定等问题。

为提高算法的收敛速度和收敛精度，本研究通过引入飞行时间因子，提出一种带自适应飞行时间因子的粒子群算法(adaptive flying time factor PSO，AFTPSO)，并通过对 4 种基本函数进行测试，比较了较高维数下 AFTPSO 算法与其他几种 PSO 算法的优化性能。

1 PSO 基本思想

假设在 D 维空间搜索一个目标函数的最优解，空间有 M 个粒子，每个粒子位置表示成一个 D 维向量 $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$，$i = 1,2,\cdots,M$，每个粒子位置 $\boldsymbol{x}_i$ 代表了一个潜在解。将 $\boldsymbol{x}_i$ 代入目标函数就可以算出其适应值，根据适应值的大小来衡量解的优劣。记第 i 个粒子的飞行速度为 $\boldsymbol{v}_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$，第 i 个粒子目前自身搜索到的最好位置($\boldsymbol{p}_{\text{best}}$)为 $\boldsymbol{p}_i = (p_{i1}, p_{i2}, \cdots, p_{id}, \cdots, p_{iD})$，所有粒子目前搜索到的最优位置($\boldsymbol{g}_{\text{best}}$)记为 $\boldsymbol{p}_g = (p_{g1}, p_{g2}, \cdots, p_{gd}, \cdots, p_{gD})$。最早期的粒子群算法通过下面公式来更新自己的速度和位置：

$$v_{id}^{k+1} = v_{id}^{k} + c_1 r_1 (p_{id} - x_{id}^{k}) + c_2 r_2 (p_{gd} - x_{id}^{k})\text{；} \tag{1}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1}\text{。} \tag{2}$$

其中：k 是迭代次数；r_1 和 r_2 为[0,1]上的随机数，用以反映搜索随机性和种群的多样性；c_1 和 c_2 是学习因子或称为加速因子，代表将每个粒子推向 $\boldsymbol{p}_{\text{best}}$ 和 $\boldsymbol{g}_{\text{best}}$ 位置的统计加速项权值，根据经验，一般取 $c_1 = c_2 = 2$。由于在粒子群算法中没有控制粒子速度的机制，所以有必要对速度的最大值进行限制，当超过这个阈值时粒子速度设为 $\boldsymbol{v}_{\max}$，这个参数非常重要，因为值太大会导致粒子跳过最好解，值太小粒子不能在局部极小点之外进行足够搜索，容易陷入到局部极值区域内。

式(1)由三部分组成：①当前粒子的速度，表示当前状态；②认知部分，表达粒子本身的“想法”；③社会部分，反映群体的信息共享。这三部分共同决定了粒子的空间搜索能力：第一部分在平衡全局同时，具备局部搜索的能力；第二部分使得群体有很强的全局搜索能力并避免陷入局部最优；第三部分反映粒子间的信息共享，使得粒子朝一个已知较好的位置移动。由于这三部分的共同影响，粒子可以到达一个有效的位置。

2 自适应飞行时间因子的粒子群算法

传统的粒子群算法往往无法兼顾收敛速度和收敛精度，一般在保证收敛速度的情况下，算法容易陷入局部最优。本研究提出的 AFTPSO 通过控制粒子飞行速度和飞行时间来达到全局搜索和局部搜索平衡，以保证算法能够在较快速度下找到问题的解。

2.1 飞行时间因子

式(2)是基本粒子群算法中粒子的位置更新公式，从物理角度考虑粒子位置改变应加上一个位移，式(2)中相当于所有粒子飞行时间都为 1，显然，鸟群觅食过程不可能保证每次飞行时间都一样，而且理论上在觅食初期往往需要大范围搜索，飞行时间比较长，而在接近食物的时候，每次飞行时间会缩短，这样更容易找到食物。基于这种考虑，在式(2)中引入了飞行时间因子，得到：

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \times T\text{。} \tag{3}$$

其中，T 表示第 i 个粒子的飞行时间，显然，基本粒子群算法中 $T = 1$。

2.2 自适应的飞行时间因子

前面已经提到，飞行时间随着粒子的进化而变化，粒子进化程度越高飞行时间应该越小，但考虑到种群多样性会随着进化越来越差，导致粒子更易陷入局部最优，此时就需要调整飞行时间来平衡种群的多样性和进化度，使得粒子在不陷入局部最优情况下，更快收敛到全局最优。本研究在参考其他参数选择与优化对粒子群算法性能影响的情况下，对如何选取飞行时间因子提出了如下方法：

$$T = F(\alpha_k, \beta_k) = T_0 + \alpha_k T_\alpha - \beta_k T_\beta\text{。} \tag{4}$$

其中：α_k 和 β_k 分别用来表示种群的多样性和种群进化度；T_0 为 T 的初始值，一般可取 0.8～1.0；T_α 和 T_β 用来调节 α_k 和 β_k 这两个参数对飞行时间因子的影响程度。

1) 种群多样性指标 α_k

种群多样性指标 α_k 用来描述粒子的聚集程度,粒子越聚集多样性越差,粒子越分散多样性越好,越不容易陷入局部最优。一般 α_k 定义如下:

$$\alpha_k = \min(f_{g_{\text{best}}}^k, \bar{f}_k)/\max(f_{g_{\text{best}}}^k, \bar{f}_k) 。\tag{5}$$

其中:$\bar{f}_k$ 表示第 k 次迭代所有粒子适应值的平均值,$f_{g_{\text{best}}}^k$ 表示第 k 次迭代群体最优位置的适应值。显然 $\alpha_k \in (0,1]$,当 $\alpha_k = 1$ 时,所有粒子都到达同一位置,即粒子聚集在一起,因此种群多样性很差;当 $\alpha_k \leqslant 1$ 时,粒子所在位置差别较大,即粒子较分散,因此粒子的多样性较好。

2)种群进化度指标 β_k

定义粒子群的进化程度指标:

$$\beta_k = \min((f_{g_{\text{best}}}^{k-1} + f_{g_{\text{best}}}^{k-2})/2, f_{g_{\text{best}}}^k)/\max((f_{g_{\text{best}}}^{k-1} + f_{g_{\text{best}}}^{k-2})/2, f_{g_{\text{best}}}^k) 。\tag{6}$$

显然,$\beta_k \in (0,1]$,β_k 值越小进化速度越快,当 $\beta_k = 1$ 时,表明算法没有进化。如果在给定次数下算法都没有进化,可以认为算法找到全局最优解或陷入局部最优。

在进化初期 β_k 值较小,此时粒子的飞行时间应该更长,即飞行时间因子 T 应较大。随着进化程度的减慢,β_k 值逐渐增大,粒子的飞行时间也逐渐缩短,即飞行时间因子 T 逐渐减小。若粒子较为分散,粒子群就不容易陷入局部最优,随着粒子群聚集度的提高,应延长飞行时间,以避免粒子群陷入局部最优。T 应随着粒子群聚集程度的增大而增大,随着粒子群进化程度的降低而减小,即 T 应随着 α_k 的增大而增大,随 β_k 的增大而减小,从而可以将 T 表示为 β_k 和 α_k 的函数,如式(4)所示。由于 $0 < \alpha_k \leqslant 1$,$0 < \beta_k \leqslant 1$,所以 $T_0 - T_\beta < T < T_0 + T_\alpha$ 。

基于上述讨论,提出一种自适应改变飞行时间因子的粒子群算法,该算法在运行过程中根据 α_k 和 β_k 的值来动态改变 T,即 T 随着算法迭代自适应变化,从而改进算法的性能。

2.3 AFTPSO 算法步骤

1)给定固定参数 w, c_1, c_2, $\boldsymbol{v}_{\max}$, T_0, T_α 和 T_β,对规模为 M 的粒子群中每个粒子随机初始化它们的速度 $\boldsymbol{v}_i$ 和位置 $\boldsymbol{x}_i$,每个粒子历史最优位置 $\boldsymbol{p}_i$ 等于各粒子初始位置,群体最优位置 $\boldsymbol{p}_g$ 为适应值最好的粒子所对应的位置;

2)根据目标函数 $f(\boldsymbol{x})$ 计算每个粒子的适应值 $f(\boldsymbol{x}_i)$;

3)对每个粒子将其适应值与历史最优位置 $\boldsymbol{p}_i$ 的适应值进行比较,若较好则更新 $\boldsymbol{p}_i$;

4)对每个粒子将其历史最优位置 $\boldsymbol{p}_i$ 与群体最优位置 $\boldsymbol{p}_g$ 的适应值进行比较,若较好则更新 $\boldsymbol{p}_g$;

5)根据式(1)和式(3)~(6)分别更新粒子的速度和位置;

6)判断是否满足终止条件,如果满足,输出最优解,否则,返回 2)。

一般终止条件为达到给定的最大迭代次数,或粒子群的最好适应值在给定的迭代数内几乎不改变。

3 数值实验和结果分析

为验证本研究提出的 AFTPSO 算法性能,测试了常用的 4 个基准函数(表 1)。Sphere 函数为非线性对称单峰函数,此函数寻优比较简单,主要用于测试算法的收敛精度;Rosenbrock 函数的等高线构成了一个"香蕉型山谷",其全局最小值就在该山谷中,由于山谷内的值变化很小,因而找到全局最优点的机会就相当小,尤其是随着维数的增加,找到全局最小值的可能性更是微乎其微,因此该函数常用来测试算法的收敛速度;Rastrigin函数是一个典型的有大量局部最小点的复杂多峰函数,因此该函数常用来测试算法能否跳出局部最优,快速找到全局最小点;Griewank 函数类似于 Rastrigin 函数,也是典型的多峰函数,但该函数的局部最优范围随着维数增加逐渐缩小,算法跳过局部最优区域迅速找到全局最小点的可能性就越大,所以该函数在较高维数时可用来测试算法的收敛精度。Rosenbrock 函数和 Rastrigin 函数的特点决定了较高维数下优化结果的精度会相对较低,所以实验中只要优化结果在 100 以内都认为算法是收敛的。表 1 中"函数最优值"是各函数的理论最小值,"函数优化精度"表示只要实验结果在该数值和函数最优值之间,则对应的算法收敛。

表 1 测试函数表

Tab.1 The test functions

函数名	函数表达式	维数	变量范围	函数最优值	函数优化精度
Sphere	$f_1(x)=\sum_{i=1}^{n}x_i^2$	30	$[-100,100]^n$	0	0.001
Rosenbrock	$f_2(x)=\sum_{i=1}^{n}100(x_{i+1}-x_i^2)^2$	30	$[-30,30]^n$	0	100
Rastrigin	$f_3(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)$	30	$[-5.12,5.12]^n$	0	100
Griewank	$f_4(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	30	$[-600,600]^n$	0	0.1

实验均在 Windows 7,Matlab2010a,E7200 2.53GH CPU,2GB RAM 计算机上实现。与本研究提出的 AFTPSO 进行比较的 4 种 PSO 是标准的 PSO[2]、带收缩因子的 PSO[9]、改进收缩因子 PSO[12] 及 FAA-PSO(2)[10]。实验参数设置如下:大量实验表明,种群规模在 20~40 时,PSO 算法一般都能获得较好的效果,因此种群规模 M 设为 30;加速因子取经验值 $c_1=c_2=2$;文献[2]提出惯性权重 w 从 0.9 线性减至 0.4,虽然该策略实现简单,但能大大提高 PSO 的性能,所以一般的 PSO 算法均采取该策略;为使不同的算法具有可比性,最大速度限制均设为 $\boldsymbol{v}_{\max}=\boldsymbol{x}_{\max}$;由于各算法运行的迭代次数差别较大,因而最大迭代次数 $k_{\max}$ 设为 10 000;其他参数具体细节详见各算法的参考文献。

本研究提出的粒子群算法新增加了飞行时间因子 T,T 的表达式中包括参数 T_0,T_α 和 T_β,在进行多次实验之后,发现 T_0 为 0.8~1.0 时算法效果较好,所以选取 $T_0=0.9$。T_α 和 T_β 可以动态调整,一般较大的 T_α 会使算法容易跳过最优解陷入振荡状态,较大的 T_β 会使算法容易陷入局部最优,因此适当选取 T_α 和 T_β 的值,对算法的性能提升有较大的影响。经过多次实验发现,T_α 在 0.05~0.1 取值,T_β 在 0.4~0.6 取值时,AFTPSO 算法的性能较好,在对比实验中,AFTPSO 算法中 T_α 和 T_β 的取值分别为 0.08 和 0.45。根据上述参数设置,各 PSO 对每个测试函数分别运行 20 次,测试结果见表 2。

表 2 各种粒子群算法对 4 个测试函数优化结果对比表

Tab.2 Result of optimizing four test function using some kind of particle swarm optimization algorithms

函数	算法性能评价指标	标准 PSO	带收缩因子 PSO	改进收缩因子 PSO	FAA-PSO(2)	AFTPSO
	迭代均值	1 537.80	552.05	529.65	857.25	602.15
$f_1(x)$	最优均值	3.4265×10^{-5}	2.4023×10^{-14}	9.0614×10^{-17}	1.6811×10^{-29}	4.3573×10^{-43}
	收敛/次	20	20	20	18	20
	迭代均值	3 517.35	1 424.10	992.00	618.50	583.45
$f_2(x)$	最优均值	73.345	92.538	76.813	47.976	7.1276
	收敛/次	20	20	20	19	20
	迭代均值	1 320.90	6 823.00	213.45	219.42	323.20
$f_3(x)$	最优均值	82.386	79.260	62.439	70.704	59.697
	收敛/次	20	19	19	19	20
	迭代均值	2 900.50	437.00	312.60	1 311.10	480.95
$f_4(x)$	最优均值	1.7753×10^{-2}	1.4779×10^{-2}	4.5502×10^{-5}	7.3960×10^{-3}	1.3724×10^{-16}
	收敛/次	20	17	19	20	19

表 2 是各粒子群算法对 4 个测试函数优化结果的对比，其中“最优均值”表示 20 次运行中收敛情况下优化结果的平均值；“迭代均值”表示 20 次运行中收敛情况下总的迭代次数的平均值。从表 2 可以看出：对 $f_1(x)$，$f_4(x)$，AFTPSO 算法比另 4 种 PSO 算法在收敛精度上有较大提高；对 $f_2(x)$，$f_3(x)$，AFTPSO 算法比另外 4 种 PSO 算法在收敛速度上有较明显改进；从收敛次数看，AFTPSO 的算法性能也有一定改善。

4 结束语

在改进粒子群算法中引入飞行时间因子，并通过考虑种群的多样性与进化度，动态调整飞行时间因子，以此来平衡算法的全局搜索和局部搜索能力，提高了算法的收敛速度和收敛精度。数值实验表明，该算法的收敛性能明显优于标准粒子群算法，也优于带收缩因子的粒子群算法、改进收缩因子的粒子群算法和文献[10]中提出的改进粒子群算法。本算法中飞行时间因子只是关于种群多样性与进化度的简单线性函数，还可以考虑其他非线性函数。另外，随着粒子群算法研究的深入，可以将一些改进的粒子群算法结合传统的聚类分析方法进行高维数据聚类分析，进行进一步研究。

参考文献：

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

[2] Shi Y，Eberhart R. A modified particle swarm optimizer[C]//Proceedings of IEEE International Conference on Evolutionary Computation. Anchorage：IEEE Service Center，1998：69-73.

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

[4] 陈国初，俞金寿. 增强型微粒群优化算法及其在软测量中的应用[J]. 控制与决策，2005，20(4)：377-381.
Chen Guochu，Yu Jinshou. Enhanced particle swarm optimization and its application in soft sensor[J]. Control and Decision，2005，20(4)：377-381.

[5] Lei K，Wang F，Qiu Y，et al. An adaptive inertia weight strategy for particle swarm optimizer[C]//International Conference on Mechatronics and Information Technology. Chongqing，Dec. 22，2005：51-55.

[6] Fourie P C，Groenwold A A. The particle swarm optimization algorithm in size and shape optimization[J]. Structural and Multidisciplinary Optimization，2002，23(4)：259-267.

[7] Jie J，Zeng J，Han C. Adaptive particle swarm optimization with feedback control of diversity[M]//Computational Intelligence and Bioinformatics. Heidelberg：Springer，2006：81-92.

[8] 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.
Zhang Xuanping，Du Yuping，Qin Guoqiang，et al. Aadaptive particle swarm algorithm with dynamically changing inertia weight[J]. Journal of Xi'an Jiaotong University，2005，39(10)：1039-1042.

[9] Clerc M. The swarm and the queen：Towards a deterministic and adaptive particle swarm optimization[C]//Congress of Evolutionary Computation. Washington：IEEE Service Center，1999：1951-1957.

[10] 张建科，刘三阳，张晓清. 飞行时间自适应调整的粒子群算法[J]. 计算机应用，2006，26(10)：2513-2515.
Zhang Jianke，Liu Sanyang，Zhang Xiaoqing. Particle swarm optimization with flying time adaptively adjusted[J]. Computer Applications，2006，26(10)：2513-2515.

[11] 周喜虎，高兴宝. 具有时间因子的粒子群优化算法[J]. 纺织高校基础科学学报，2011，24(2)：303-308.
Zhou Xihu，Gao Xingbao. Particle swarm optimization with time factor[J]. Basic Sciences Journal of Textile Universities，2011，24(2)：303-308.

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

（责任编辑：吕文红）