文章编号：2095-2163(2023)09-0005-04　　中图分类号：TP202+.1　　文献标志码：A

自适应惯性权重优化的粒子群算法

张　豪，王贤琳
(武汉科技大学 机械自动化学院，武汉 430081)

摘　要：惯性权重作为粒子群最重要的参数之一，对全局搜索能力和局部搜索能力有重要的影响。针对传统粒子群算法的局限性，本文对其惯性权重进行改进，提出自适应惯性权重优化的粒子群算法，与原始粒子群算法相比，现在惯性权重和迭代次数与每个粒子适应度有关。仿真结果表明：本文所提出的自适应粒子群算法在迭代次数上优于基本粒子群算法，平均适应度低于基本粒子群算法。

关键词：自适应惯性权重；粒子群算法；迭代次数

Adaptive inertia weight particle swarm optimization algorithm

ZHANG Hao，WANG Xianlin
(School of Machinery and Automation，Wuhan University of Science and Technology，Wuhan 430081，China)

【Abstract】 As one of the most important parameters of particle swarm，inertia weight has an important influence on global search ability and local search ability. Aiming at the limitation of traditional particle swarm optimization algorithm，the inertia weight is improved and an adaptive inertia weight particle swarm optimization algorithm is proposed. Compared with before，the inertia weight is related to the number of iterations and the fitness of each particle. The simulation results show that the proposed adaptive particle swarm optimization algorithm is superior to the basic particle swarm optimization algorithm in the number of iterations，and the average fitness is lower than the basic particle swarm optimization algorithm.

【Key words】 adaptive inertia weight；particle swarm optimization (PSO)；iteration times

0　引　言

粒子群算法(Particle Swarm Optimization，PSO)是由美国学者 Kennedy 和 Eberhart 于 1995 共同提出的，通过对鸟群捕食习惯仿真，利用群体和个体之间信息共享达到捕食的目的，作为智能启发式算法之一，具有操作简单、参数少、易实现等优点[1]。许多学者对粒子群算法进行改进，以加强粒子群寻优性能。文献[2]首次提出粒子群惯性权重，惯性权重取 0.9~1.2 时，粒子群具有较好的性能；文献[3]提出线性递减惯性权重，惯性权重线性下降时，粒子群在运行时可能缺乏全局搜索能力。近年来，为了提高粒子群算法的稳定性，研究人员主要从惯性权重、学习因子和粒子群拓扑关系分析展开研究。

惯性权重是粒子群算法的核心参数之一，影响着算法的收敛性。为了加强算法稳定性，改善收敛能力，文献[4]提出惯性权重一定时，粒子具有较好的收敛性，但是此方法在高维测试函数上求解较弱；文献[5]提出了正态分布衰减惯性权重粒子群优化，使得算法能很好的平衡全局搜索和局部搜索能力；文献[6]对粒子运动状态实施动态监测，并实时调整粒子惯性权重，大大减少粒子无效迭代次数；文献[7]赋予每个粒子每一维度以不同的线性衰减混沌化惯性权重，够较大幅度地增强粒子群算法的搜索能力，提高算法的寻优精度。

本文提出一种自适应惯性权重优化的粒子群算法(Adaptive Particle Swarm Optimization，APSO)，将惯性权重和迭代次数以及每个粒子适应度联系起来，自适应的调整粒子群体中各粒子的惯性权重，改善算法性能。

1　基本粒子群算法(PSO)

粒子群算法在 D 维空间中将每个粒子当作空间中的一个点，在求解过程中粒子不断迭代更新改

基金项目：国家自然科学基金(51975432)。
作者简介：张　豪(1998-)，男，硕士研究生，主要研究方向：绿色制造；王贤琳(1968-)，女，博士，教授，硕士生导师，主要研究方向：再制造。
通讯作者：王贤琳　　Email：wxlwel@ sina.com
收稿日期：2022-09-20

变位置,直到找到最优解,粒子 i 的位置和速度迭代如公式(1)和公式(2),位置与速度皆为向量。

$$v_i^{d+1} = w_i^d v_i^d + c_1 r_1 (p^{best_i^d} - x_i^d) + c_2 r_2 (g^{best^d} - x_i^d) \tag{1}$$

$$x_i^{d+1} = x_i^d + v_i^{d+1} \tag{2}$$

其中,w 为速度的惯性权重;c_1, c_2 为加速因子,一般取值为2;r_1, r_2 为0 ~ 1的随机数;v_i^d 为粒子上一轮迭代的速度;$pbest_i^d - x_i^d$ 为社会学习向量;$gbest^d - x_i^d$ 为个体学习向量。

2 自适应惯性权重粒子群算法(APSO)

惯性权重是粒子群算法很重要的参数,惯性权重一般取值2,对于取定值的粒子群算法,收敛效果并不理想。文献[3]最先加入惯性权重,并分析指出一个较大的惯性权值有利于全局搜索,而一个较小的权值则更利于局部搜索。为了使粒子群算法更稳定,对粒子群算法惯性权重采取自适应变化,与原始粒子群算法相比,现在惯性权重和迭代次数与每个粒子适应度有关。对于最小值问题,惯性权重变化规则如式(3);对于最大值问题,惯性权重变化规则如式(4)。

$$w_i^d = \begin{cases} w_{\min} + (w_{\max} - w_{\min}) \dfrac{f(x_i^d) - f_{\min}^d}{f_{\text{average}}^d - f_{\min}^d}, & f(x_i^d) \leqslant f_{\text{average}}^d \\ w_{\max}, & f(x_i^d) > f_{\text{average}}^d \end{cases} \tag{3}$$

$$w_i^d = \begin{cases} w_{\min} + (w_{\max} - w_{\min}) \dfrac{f_{\max}^d - f(x_i^d)}{f_{\max}^d - f_{\text{average}}^d}, & f(x_i^d) \geqslant f_{\text{average}}^d \\ w_{\max}, & f(x_i^d) < f_{\text{average}}^d \end{cases} \tag{4}$$

其中,$w_{\min}$ 和 $w_{\max}$ 为预先给定的最小惯性系数和最大惯性系数,一般取0.4和0.9。

第 d 次迭代时所有粒子的平均适应度,式(5):

$$f_{\text{average}}^d = \sum_{i=1}^{n} f(x_i^d) / n \tag{5}$$

第 d 次迭代时所有粒子的最小适应度,式(6):

$$f_{\min}^d = \min\{f(x_1^d), f(x_2^d), \cdots, f(x_n^d)\} \tag{6}$$

在每次迭代寻优时,总有部分粒子找到更优的位置,也有部分粒子在较优和较差的位置,在结束此次迭代进行下次迭代时,那些处于越优位置的粒子会进一步达到更优的位置,而在较差位置的粒子会越来越差。经过不断迭代,越优位置的粒子会更接近或达到全局最优位置。每次迭代更新时,依据上次迭代粒子的适应度值,在下次迭代时动态调整惯性权重,对粒子全局寻优和快速收敛有很大帮助。

自适应惯性权重粒子群算法流程:

(1)初始化粒子,设置群体规模 N,最大迭代次数 T,包括粒子的速度和位置,给出个体学习因子和社会学习因子;

(2)计算每个粒子适应度,将单个粒子的最优位置和群体粒子的最优位置分别记为 $pbest_i^d$ 和 $pbest^d$;

(3)算法是否收敛,若是,则直接输出 $pbest^d$,否则进入下一步;

(4)通过式(7)计算粒子 i 在第 d 次迭代后的适应度值变化:

$$\delta f(x_i^d) = f(x_i^d) - f(x_i^{d-1}) \tag{7}$$

其中,$i = 1, 2, \cdots, n, t \geqslant 2$;$f(x_i^d)$ 表示粒子 i 在第 d 次迭代后的适应度值;

(5) 根据式(3) 动态调整惯性权重;

(6)根据式(1)和式(2)更新粒子群体速度和位置;

(7)重新计算粒子适应度,存储 $pbest_i^d$ 和 $pbest^d$,并跳转到步骤(3);

(8)输出群体最优适应度 $pbest^d$,运行结束。

3 仿真试验

3.1 测试函数

为了验证自适应惯性权重粒子群算法的有效性,将固定权重的粒子群算法与自适应惯性权重优化的粒子群算法进行性能比对分析。

Sphere 函数为典型的单峰函数,仅有一个极值点;Rosenbrock 具有一个全局最小值点,但其为病态函数,一般算法难以求得最优解;Rastrigin 和 Griewank 为多峰函数,解空间具有多个局部最小值点。各测试函数的函数表达式、维数、取值范围、理论极值和误差目标见表1。

3.2 参数设置

对于基本 PSO 算法,权值固定 $w = 0.9$,$c_1 = c_2 = 2$;APSO 算法权值 $w_{\max} = 0.9$,$w_{\min} = 0.4$,$c_1 = c_2 = 2.05$;对于这两种算法,粒子数量都设置为1 000,变量个数为30,每次求解过程算法迭代的最大次数为1 000次。

3.3 实验结果

每个算法对每个测试函数独立运行30次,各个函数的适应度及运行时间见表2、表3。

表 1　标准测试函数及其参数

Tab. 1　Standard test function and its parameters

函数名	函数表达式	维数	取值范围	理论极值	误差目标
Sphere	$f(x)=\sum_{i=1}^{n}x_i^2$	30	$[-100,100]^n$	0	10^{-2}
Rosenbrock	$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	$[-30,30]^n$	0	10^2
Rastrigin	$f(x)=\sum_{i=1}^{n}(x^2-10\cos(2\pi x_i)+10)$	30	$[-5.12,5.12]^n$	0	10^2
Griewank	$f(x)=\frac{1}{4\,000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	$[-600,600]^n$	0	10^{-1}

表 2　各个函数适应度结果对比

Tab. 2　Comparison of fitness results of each function

函数	基本 PSO			APSO		
	最大适应度	最小适应度	平均适应度	最大适应度	最小适应度	平均适应度
Sphere	0.409 6	0.030 9	0.142 0	0.003 4	9.39E-06	6.73E-04
Rosenbrock	604.288 0	27.800 0	62.988 1	492.436 0	27.143 4	99.139 1
Rastrigin	66.463 6	22.293 1	42.500 7	103.475 4	27.858 8	56.381 6
Griewank	0.516 8	0.073 2	0.323 2	0.052 0	5.98E-04	0.019 6

表 3　各个函数运行时间(Time/s)结果对比

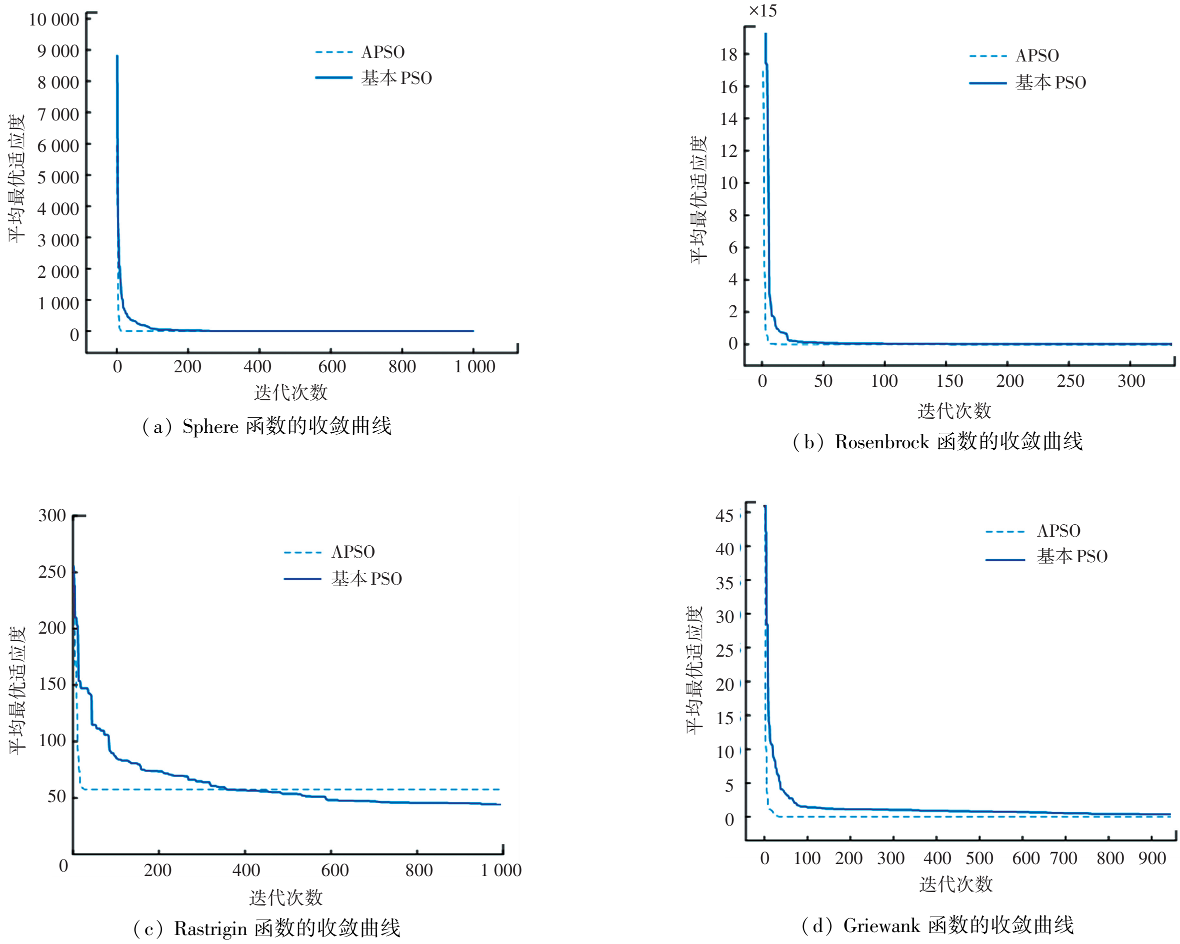
Tab. 3　Comparison of run time (Time/s) results by function

函数	基本 PSO			APSO		
	最大 Time	最小 Time	平均 Time	最大 Time	最小 Time	平均 Time
Sphere	4.957 841	4.702 290	4.825 312	6.061 252	5.508 042	5.699 254
Rosenbrock	6.980 216	6.288 116	6.444 974	7.434 860	6.982 172	7.252 117
Rastrigin	7.207 524	6.497 882	6.812 241	7.314 094	6.742 256	7.019 888
Griewank	7.909 218	7.478 285	7.607 106	7.685 933	7.147 286	7.327 177

从表 2 可以看出粒子群算法在对 Sphere 函数和 Griewank 函数寻找最低值时明显优于 Rosenbrock 函数和 Rastrigin 函数，无论是基本 PSO 还是 APSO 算法，对于 Sphere 函数和 Griewank 函数，其平均适应度小于 1，而对于 Rosenbrock 函数和 Rastrigin 函数，其平均适应度在 40~100 之间，表明在测试函数 Rosenbrock 和 Rastrigin 上，具有不稳定性。对于基本 PSO 和 APSO 两种算法，在测试函数 Sphere 和 Griewank 上也可以看出 APSO 明显优于基本 PSO 算法，例如 Sphere 函数中，基本 PSO 算法的平均适应度为 0.142 0，APSO 算法的平均适应度为 6.73E-04。至于 Rosenbrock 函数和 Rastrigin 函数，APSO 的平均适应度稍大于基本 PSO，也进一步说明粒子群算法优化的不稳定性。

见表 3，Sphere 函数较为简单，平均运行时间最短，基本 PSO 为 4.825 3，APSO 为 5.699 2，均小于其他函数平均运行时间。对于所有的测试函数，APSO 算法的运行时间全部大于基本 PSO 算法，说明 APSO 算法的惯性权重为自适应变化，优化性能更好，优化时间也较长。

为了更加清楚的看到两种算法的收敛性，对测试函数进行收敛性分析，采用基本 PSO 和 APSO 算法分别求解 4 种测试函数成功收敛时的平均最优适应度下降曲线如图 1 所示，可以看出两种算法在探索阶段均可实现有效搜索，其中 APSO 算法的平均适应度相比于基本 PSO 算法下降较快，迭代次数也明显少于基本 PSO 算法。



（a）Sphere 函数的收敛曲线

（b）Rosenbrock 函数的收敛曲线

（c）Rastrigin 函数的收敛曲线

（d）Griewank 函数的收敛曲线

图 1 测试函数收敛曲线对比图

Fig. 1 Test function convergence curve comparison diagram

4 结束语

为了改善传统 PSO 算法的收敛性能，本文提出一种自适应惯性权重优化的粒子群算法 APSO，惯性权重采取自适应变化，与每个粒子的适应度有关，该算法简单，推广性强。对 Sphere、Rosenbrock、Rastrigin 和 Griewank 4 个函数进行验证，结果表明 APSO 算法在 Sphere 和 Griewank 函数上有较好的效果，其最小值分别为 6.73E-04 和0.019 6，精度大幅提高，APSO 明显优于基本 PSO 算法。但 APSO 具有一定的不稳定性，后续也可与其他方法融合以提高算法稳定性。

参考文献

[1] KENNEDY J, EBERHART R. Particle swarm optimization[C]// Proceedings of ICNN´95 - international conference on neural networks. IEEE, 1995: 1942-1948.

[2] SHI Y, EBERHART R. A modified particle swarm optimizer [C]// 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360). IEEE, 1998: 69-73.

[3] SHI Y, EBERHART R C. Empirical study of particle swarm optimization [C]// Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406). IEEE, 1999: 1945-1950.

[4] CLERC M, KENNEDY J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space [J]. IEEE transactions on Evolutionary Computation, 2002, 6 (1): 58-73.

[5] 徐浩天，季伟东，孙小晴，等. 基于正态分布衰减惯性权重的粒子群优化算法[J]. 深圳大学学报(理工版)，2020，37(2)：208-213.

[6] 敖永才，师奕兵，张伟，等. 自适应惯性权重的改进粒子群算法[J]. 电子科技大学学报，2014，43(6)：874-880.

[7] 蒋晓岫，任佳，顾敏明. 多维度惯性权重衰减混沌化粒子群算法及应用[J]. 仪器仪表学报，2015，36(6)：1333-1341.