第 36 卷 第 9 期 计 算 机 工 程 2010 年 5 月
Vol.36 No.9 Computer Engineering May 2010
·人工智能及识别技术· 文章编号：1000—3428(2010)09—0192—03 文献标识码：A 中图分类号：TP311.52

一种新的粒子群优化算法

代 军，李 国，徐 晨，陶 艾
(深圳大学数学与计算科学学院，深圳 518060)

摘 要：针对传统粒子群优化算法容易早熟、收敛精度低等缺点，提出一种改进方案，使用随机惯性权重，在每一次迭代中，对可能陷入局部极值的粒子进行有效的随机初始化。通过对 7 个经典测试函数的数值仿真实验证明，该新算法能提高粒子群优化算法的寻优能力，并在维数较高时也能获得较好的优化效果。

关键词：粒子群优化；权重；随机初始化

New Particle Swarm Optimization Algorithm

DAI Jun, LI Guo, XU Chen, TAO Ai
(College of Mathematics and Computational Science, Shenzhen University, Shenzhen 518060)

【Abstract】To overcome the disadvantages of Particle Swarm Optimization(PSO) algorithm such as premature, bad convergence precision, a new improvement scheme is presented. In the new algorithm, random inertia weight is used, and at each iteration, the particles which may fall into the local extremum area are randomly initialized renewedly in an effective method. The numeric experiments of seven classic benchmark functions indicate that the new algorithm greatly improves the particles' ability of searching global optimal value, and has good results in higher dimensions.

【Key words】Particle Swarm Optimization(PSO); weight; random initialization

1 概述

粒子群优化(Particle Swarm Optimization, PSO)算法是一种模拟鸟群飞翔、鱼群游泳的群智能仿生优化算法，其概念明确、易于实现，目前已经得到广泛应用[1-2]。但PSO算法存在的最突出问题是在进化后期容易陷入局部极值区域，难以收敛到全局最优解，从而又限制其进一步应用。因此，提高PSO算法的优化性能显得尤为必要。

本文分析PSO算法陷入局部极值区域时种群的特点，将PSO算法的最新进展[3-4]和粒子自身信息相结合，提出一种新的粒子群优化算法。将其应用于函数优化中，得到较好的优化效果。

2 2 种典型的 PSO 算法

PSO 算法的描述如下：

设在 D 维搜索空间中，某种群的规模为 N，其粒子 i 的第 t 代在该搜索空间中的位置为 $X_i^t=(x_{i1}^t,x_{i2}^t,\cdots,x_{iD}^t)$，$i=1,2,\cdots,N$，速度为 $V_i^t=(v_{i1}^t,v_{i2}^t,\cdots,v_{iD}^t)$，个体历史极值位置为 $P_i^t=(p_{i1}^t,p_{i2}^t,\cdots,p_{iD}^t)$，第 t 代的全局极值位置为 $P_g^t=(p_{g1}^t,p_{g2}^t,\cdots,p_{gD}^t)$，粒子 i 根据如下 2 个方程更新自己：

$$V_i^{t+1}=V_i^t+c_1r_1(P_i^t-X_i^t)+c_2r_1(P_g^t-X_i^t) \quad (1)$$

$$X_i^{t+1}=X_i^t+V_i^{t+1} \quad (2)$$

其中，c_1,c_2 是学习因子，一般取值为 2；r_1 是 [0,1] 之间均匀分布的随机数。在式(1)右项中，V_i^t 称为粒子的先前速度；$c_1r_1(P_i^t-X_i^t)$ 为粒子的自我认知部分；$c_2r_1(P_g^t-X_i^t)$ 为粒子的社会认知部分。

粒子的先前速度和自我认知部分体现了粒子对自身经验的积累；粒子的社会认知部分体现了粒子之间的信息传递。算法在随机初始化一群粒子之后迭代运行，且在每次迭代过程中，粒子通过式(1)、式(2)更新自己，直到满足迭代停止条件为止。

上述算法在实验中存在收敛速度慢、精度差的缺点。为了加快收敛速度和精度，有学者引入惯性权重的概念，并提出一种线性递减的权重策略，即将式(1)变形为

$$V_i^{t+1}=wV_i^t+c_1r_1(P_i^t-X_i^t)+c_2r_1(P_g^t-X_i^t) \quad (3)$$

其中，w 称为惯性权重，$w=w_{\max}-(w_{\max}-w_{\min})\times iter/iter_{\max}$，$w_{\max}$ 称为最大惯性权重，$w_{\min}$ 称为最小惯性权重，$iter$ 为当前迭代次数，$iter_{\max}$ 为总迭代次数。这样，w 就会随着迭代次数的递增而线性递减，所以，使用式(3)速度更新方程的 PSO 算法被称为 LDWPSO(Linear-Decreased Weight Particle Swarm Optimization)算法。该方法极大提高了 PSO 算法的收敛速度和寻优精度。

为了确保 PSO 算法的收敛，Clerc 引入了压缩因子 α，其速度更新公式如下：

$$V_i^{t+1}=\alpha\times(V_i^t+c_1r_1(P_i^t-X_i^t)+c_2r_1(P_g^t-X_i^t)) \quad (4)$$

其中，$\alpha=2/|2-\varphi-\sqrt{\varphi^2-4\varphi}|\ (\varphi=c_1+c_2>4)$。一般情况下，$c_1=c_2=2.05$，所以，$\alpha=0.729$。使用式(4)速度更新方程的

基金项目：国家“863”计划基金资助项目(2006AA01A116)；广东省自然科学基金资助项目(2008329)；广东省教育部产学研结合基金资助项目(2009B090300355)

作者简介：代 军(1985－)，男，硕士，主研方向：小波分析，智能计算；李 国，副教授、博士；徐 晨，教授、博士；陶 艾，硕士

收稿日期：2009-11-17 **E-mail：**water20007@163.com

PSO 算法一般称为带有压缩因子的 PSO(Particle Swarm Optimization with Constriction factor, CPSO)算法。

3 一种新的 PSO 算法

与 PSO 算法相比，LDWPSO 算法和 CPSO 算法在收敛速度和精度上都得到了很大的改进，但是在进化的后期，粒子容易陷入局部极值区域，从而难以收敛到更好的解，尤其是在高维情况下。

要使粒子在陷入局部极值之后能跳出局部极值区域，让粒子变异是一种最为直接的想法。但有 2 个问题：

(1)以何种方法选取需要变异的粒子；

(2)对选取后的粒子以何种方式变异。

陷入局部极值的粒子之间，差异性很小，但在实验中发现，并不是此时种群中所有粒子之间的差异性都很小，而是仍有少部分的“活跃”粒子(在实验中主要表现为：在最后若干次迭代中，少部分粒子的适应值与全局极值相差很大)，但由于 2 个“极值”已经陷入局部极值，使得这些活跃的粒子得不到正确的“指导”，从而使这些仅剩的、有发展潜力的粒子也只能在远离全局最优区域的地方“徘徊”，因此算法难以找到更优解。所以，若保留这些“活跃”粒子，而只对陷入极值的粒子变异，则可能提高算法的性能。

采取何种方式对粒子进行变异也是提高算法性能的关键。文献[3]提出在粒子陷入局部极值之后，对满足一定条件的粒子进行重新随机初始化的方法，取得了较好的效果。这种方法实现简单，但有一个很大的缺点，即如果随机数的随机性变化不大，则粒子容易陷入新的局部极值区域。所以，若在重新随机初始化的同时，能加入待变异粒子当前的一些信息，则不仅能使算法具有部分动态自适应性，而且能防止算法陷入新的局部最优。

实验证明：较大的惯性权重有利于全局搜索，较小的权重有利于局部搜索。对于是否采取权重线性递减策略，文献[4]经大量的实验发现：w 在 0.4~0.6 之间随机取值，即：

$$w = r_1 \times 0.2 + 0.4 \tag{5}$$

实验效果优于 LDWPSO。为此，结合文献[3]及上述思考，提出一种新型的粒子群优化算法(NPSO)。

首先，对 NPSO 算法中参数做如下规定：设种群第 t 代中各粒子的适应值分别为 $f_1^t, f_2^t, \cdots, f_N^t$，则当代种群的平均适应值 $f_{avg}^t = \sum_{i=1}^{N} f_i^t / N$，粒子 i 与 f_{avg}^t 绝对偏差 $\delta_i^t = | f_i^t - f_{avg}^t |$，检验值 $\lambda_i^t = f_{avg}^t / N$，其中，$N$ 为种群规模。其次，对每个粒子陷入局部极值区域的判断是：$\delta_i^t < \lambda_i^t$；为保持算法的稳定性，对每个待变异且不是全局极值的粒子的操作为

$$X_i^t = r_1(Xup - Xdown) + Xdown \times (-\delta_i^t) \tag{6}$$

其中，$Xup, Xdowm$ 分别为搜索区间的上、下解界；r_1 是 [0,1] 之间均匀分布的随机数。这样，NPSO 即可以保留“活跃”粒子，又对可能陷入局部极值的粒子进行新的重新随机初始化，并保持算法的稳定性，因而能提高 PSO 算法的优化性能。最后，NPSO 算法的步骤如下：

Step1 随机初始化粒子的位置、速度、最大速度，计算粒子的适应值，并初始化种群的全局极值和粒子的个体历史极值。

Step2 根据式(5)计算权重，并根据式(2)、式(3)更新粒子的速度和位置；其中，若粒子当前速度(位置)超过最大速度(区间上界)，则当前速度(位置)为最大速度(区间上界)；若当前速度(位置)小于最大速度的负数(区间下界)，则当前速度(位置)取最大速度的负数(区间下界)。

Step3 计算粒子的适应值，更新粒子的个体极值和全局极值，计算 $f_{avg}^t, \delta_i^t, \lambda_i^t$，并判断粒子是否陷入局部极值，若是，转向 Step4，否则，转向 Step5。

Stpe4 对满足条件的粒子按式(6)随机重新初始化，之后，转向 Step5。

Step5 若算法中止条件满足，执行 Step6，否则，迭代次数加 1，转向 Step2。

Step6 输出全局极值，停机。

4 数值实验

4.1 测试函数

为了验证 NPSO 寻优性能，在实验中，选取 7 个测试函数对 NPSO, LDWPSO, CPSO 及文献[4]方法(记为 FRPSO)的性能进行对比，如表 1 所示。

表 1 测试函数

函数	搜索区间	最优解	最优值	预设精度
$f_1(X) = \sum_{i=1}^{n} x_i^2$	[-100,100]	x_i =0	0	≤e-9
$f_2(X) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,10]	x_i =0	0	≤e-6
$f_3(X) = \sum_{i=1}^{n} \frac{1}{4\,000} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	x_i =0	0	≤e-3
$f_4(X) = \sum_{i=1}^{n-1} 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2$	[-5.12,5.12]	x_i =1	0	≤e-2
$f_5(X) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[-100,100]	x_i =0	0	≤e-9
$f_6(X) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert}) + 418.982\ 9 \times n$	[-500,500]	x_i =420.98	0	≤e-6
$f_7(X) = \sum_{i=1}^{n} (x_i - i)^2$	[-n, n]	$x_i = i$	0	≤e-7

4.2 实验参数设置、方案及结果

4.2.1 实验参数设置

在实验中，算法的公共参数设置为：种群大小固定为 50，粒子的最大速度为搜索区间长度的 10%，函数搜索区间见表 1，其中，f_7 的搜索区间为[-160,160]。LDWPSO 算法中的权重的变化范围为 0.1~0.9。

4.2.2 实验方案

方案 1 在固定迭代次数和预设精度条件下，比较 4 种算法的收敛率该方案中，函数维数固定为 60，迭代 2 000 次，每个函数运行 50 次，预设精度见表 1。若算法在指定迭代次数内，最终优化结果满足预设精度，则称该次算法收敛，否则为不收敛，设算法收敛的次数为 M_s，则算法收敛率为 $M_s/50$，实验结果见表 2。

表 2 方案 1 实验结果

算法	f_1	f_2	f_3	f_4	f_5	f_6	f_7
LDWPSO	0.00	0.00	0.80	0.00	0.00	0.00	0.04
CPSO	0.78	0.00	0.44	0.00	0.00	0.00	0.94
FRPSO	0.34	1.00	0.78	0.00	0.00	0.00	1.00
NPSO	1.00	1.00	0.98	1.00	1.00	1.00	1.00

方案 2 固定迭代次数，比较 4 种算法在不同维数下的平均值、最小值、方差，在此方案中，函数迭代次数固定为 2 000，每个函数运行 30 次，测试函数维数分别为 90, 120。

以 30 次运行结果的平均值(avg)和最小值作为算法的优化结果和最优值(opt)，并计算相应的方差(var)。实验结果见

表 3~表 4，f_1~ f_7 在 d =120 的进化曲线见图 1~图 8。

表 3　方案 2(d=90)实验结果

算法		f_1	f_2	f_3	f_4	f_5	f_6	f_7
LDWPSO	avg	7.280e-2	1.337e-1	4.936e-2	1.288e+2	3.443e+2	1.824e+4	1.056e+3
	opt	8.999e-4	2.525e-2	6.471e-4	8.503e+1	2.497e+2	1.528e+4	7.519e-3
	var	1.111e-2	1.345e-2	2.697e-3	1.448e+3	2.374e+3	2.806e+6	5.841e+6
CPSO	avg	1.261e-1	1.887	4.192e-1	1.400e+2	1.110e+3	1.896e+4	1.121e+3
	opt	1.052e-4	5.138e-2	1.475e-3	7.131e+1	6.049e+2	1.493e+4	5.831e-5
	var	1.526e-1	1.371e+1	5.720e-1	2.069e+3	1.157e+5	3.393e+6	6.495e+6
FRPSO	avg	9.770e-4	1.379e-3	9.129e-3	1.319e+2	2.870e+2	1.878e+4	2.911e-3
	opt	1.797e-4	2.385e-4	6.167e-5	8.004e+1	1.972e+2	1.589e+4	3.096e-4
	var	5.481e-7	2.019e-6	1.538e-4	1.718e+3	1.806e+3	1.993e+6	5.576e-6
NPSO	avg	1.593e-6	3.663e-4	7.663e-3	9.647e-2	6.879e-2	1.688e-3	2.609e-3
	opt	0	1.301e-5	6.831e-6	1.758e-2	0	2.869e-4	2.558e-4
	var	1.225e-11	5.188e-8	7.638e-5	2.825e-2	6.145e-2	2.274e-6	3.279e-6

表 4　方案 2(d=120)实验结果

算法		f_1	f_2	f_3	f_4	f_5	f_6	f_7
LDWPSO	avg	2.519	2.951	3.564e-1	1.843e+2	6.011e+2	2.531e+4	1.404e+4
	opt	1.502e-1	2.392e-1	6.583e-2	1.184e+2	4.341e+02	2.153e+4	6.716e-1
	var	1.323e+1	2.307e+1	4.099e-2	2.895e+3	7.714e+3	3.972e+6	7.756+7
CPSO	avg	9.927	5.204	1.016	2.840e+2	1.831e+3	2.649e+4	1.200e+4
	opt	2.054e-1	5.433e-1	1.155e-1	1.598e+2	1.046e+3	2.279e+4	2.154e+1
	var	3.247e+2	1.177e+1	1.652	5.357e+3	2.393e+5	4.669e+6	7.987e+7
FRPSO	avg	2.084e-1	1.131e-1	9.959e-2	2.155e+2	5.334e+2	2.549e+4	2.842e+3
	opt	5.678e-2	2.882e-2	2.912e-2	1.195e+2	4.183e+2	1.932e+4	1.198e-1
	var	1.098e-2	3.149e-2	6.932e-2	1.998e+3	6.808e+3	5.926e+6	9.318e+6
NPSO	avg	1.112 e-4	1.394e-2	1.981e-2	5.278e-1	4.992e-1	4.037e-1	6.144e-1
	opt	0	1.599e-3	2.703e-4	2.029e-1	0	1.295e-1	2.079e-1
	var	1.734e-8	7.129e-5	4.563e-4	4.766e-2	9.083e-1	4.521e-2	1.232e-1

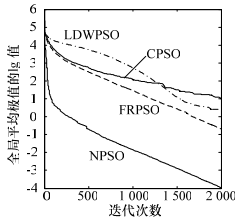

图 1　f_1进化过程

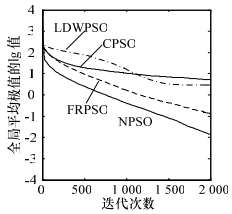

图 2　f_2进化过程

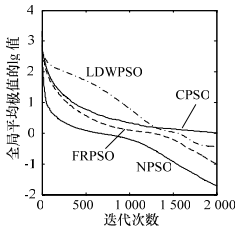

图 3　f_3进化过程

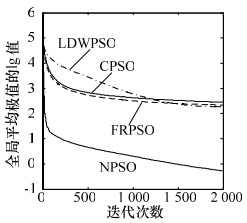

图 4　f_4进化过程

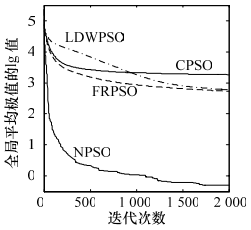

图 5　f_5进化过程

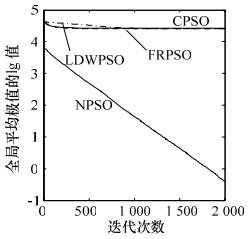

图 6　f_6进化过程(整体)

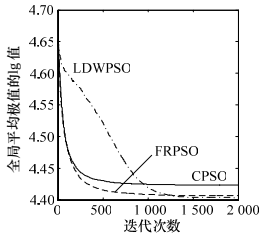

图 7　f_6进化过程(整体的上部)

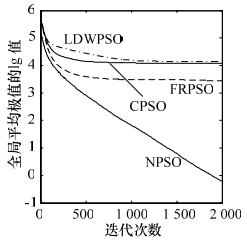

图 8　f_7进化过程

5　结束语

本文针对粒子群优化算法在进化后期容易陷入局部极值的缺点，对PSO算法进行 2 个方面改进：(1)使用随机惯性权重；(2)对满足一定条件的粒子进行一种新的重新随机初始化。前者有效平衡了全局搜索能力和局部开发能力[4]，后者有效预防了粒子重新陷入局部极值。新算法明显地改善了PSO算法，且实现简单，在较高维也有很好的效果，是一种应用前景较好的智能优化算法。

参考文献

[1] 葛洪伟，刘林炬. 基于改进粒子群优化算法的矩形 Packing 问题[J]. 计算机工程, 2009, 35(7): 186-188.

[2] 高洪元，刁　鸣，贾宗圣，等. 基于克隆粒子群优化算法的多用户检测器[J]. 计算机工程, 2008, 34(3): 228-230.

[3] 李玉毛. 粒子群优化算法的研究及改进[D]. 西安：西北大学, 2009.

[4] 黄　轩，张　军，詹志辉. 基于随机惯量权重的快速粒子群优化算法[J]. 计算机工程与设计, 2009, 30(3): 647-650.

编辑　陈　文

（上接第 191 页）

5　结束语

本文的创新点在于：将蚁群算法和聚类算法相结合，将大型网络分割成子网络进行处理后进行合成以及在过程中直接简化网络的方式来获取最优解，同时在求解的步骤中，透明化某些无意义的点，从而加快收敛速度，提高算法效率。

参考文献

[1] 吴　斌，史忠植. 一种基于蚁群算法的 TSP 问题分段求解算法[J]. 计算机学报, 2001, 24(12): 1328-1333.

[2] 王合义，丁建立，唐万生. 基于蚁群优化的路由算法[J]. 计算机应用, 2008, 28(1): 7-13.

[3] Dortgo M, Maniezzo V. Ant System: Optimization by A Colony Cooperating Agents[J]. IEEE Trans. on System, Man, and Cybernetics, 2006, 26(1): 29-41.

[4] Zhou Xiawei, Chen Changjia. A Genetic Algorithm for Multicasting Routing Problem[C]//Proc. of WCC-ICCT'00. Beijing, China: [s. n.], 2000.

[5] Eberhart R C, Keennedy J. A New Optimizer Using Particle Swarm Theory[C]//Proc. of the 6th International Symposium of Micro Machine and Human Science. Nagoya, Japan: [s. n.], 2005: 39-43.

[6] 叶小勇，雷　勇. 蚁群算法在全局最优路径寻优中的应用[J]. 系统仿真学报, 2007, 19(24): 5643-5647.

编辑　金胡考

.