基于杂交变异的动态粒子群优化算法

周利军[1] 彭 卫[2] 曾小强[2] 邹 芳[2]
(四川农业大学资源环境学院 成都 611130)[1] (四川农业大学商学院 成都 611830)[2]

摘 要 粒子群优化算法(PSO)的结构相对简单、运行速度很快,但是算法极易陷入局部最优,出现早熟收敛现象。针对标准粒子群算法存在的问题,引入了一种随迭代次数和粒子间距离大小动态改变的惯性权重,通过设置比例系数控制二者对惯性权重的影响力度。在此基础上为了增加种群多样性,又引入"杂交变异"算子,设计了一种基于杂交变异的动态粒子群优化算法(HV-DPSO)。通过对基准函数的数值试验表明,新算法相对于标准粒子群算法不仅能有效地避免早熟收敛,而且具有更好的收敛效果。

关键词 粒子群优化算法,动态惯性权重,杂交变异,早熟收敛,多样性

中图法分类号 TP18 **文献标识码** A

Dynamic Particle Swarm Optimization Based on Hybrid Variable

ZHOU Li-jun[1] PENG Wei[2] ZENG Xiao-qiang[2] ZOU Fang[2]
(College of Resources Environment, Sichuan Agricultural University, Chengdu 611130, China)[1]
(School of Business, Sichuan Agricultural University, Chengdu 611830, China)[2]

Abstract Particle swarm optimization(PSO) is a relatively simple structure which runs very quickly, but it is easily fall into local optimum and appears the phenomenon of premature convergence. Aiming at the PSO existing problems, by setting the proportional coefficient control of inertia weight between influence strength, this paper introduced a kind of novel way using the iteration number and particle size of the distance between the dynamic change inertia weight. At the same time, in order to increase the diversity of population, using "hybrid variation" operator, designed a kind of dynamic particle swarm optimization based on hybrid variable, (HV-DPSO) based on reference function of numerical experiment. The experimental results show that compared with the traditional PSO, the new algorithm not only can effectively avoid premature convergence but also has better convergence effect.

Keywords Particle swarm optimization, Dynamic inertial weight, Hybrid variation, premature convergence, Diversity

1 引言

粒子群算法(Particle Swarm Optimization, PSO)是一种基于群体的智能优化算法,是通过群体内粒子间的合作与竞争产生的群体智能优化寻优策略,最早由 Kennedy 和 Eberhart 于 1995 年提出[1,2]。该算法由于原理十分简单,所需要的参数也较少,并且易于实现,因此已经被广泛应用于多目标函数优化、系统设计、分类、模式识别、生物系统建模、电力系统等许多领域[3]。PSO 算法一经提出就成为国内外众多学者的研究热点,他们提出了各种改进办法,但粒子群优化算法理论基础薄弱,在后期探测能力不够、收敛速度较慢,对收敛性的讨论很少,证明更没有得到完善解决。针对 PSO 算法的这些问题,研究者们提出了许多通过调整惯性权重来提高粒子群优化算法性能的方法[4,5],例如惯性权重线性递减法[6]、根据不同粒子与全局最优点的距离动态调整惯性权重[7]、自适应惯性权重法[8,9]、动态自适应调整惯性权重策略[10,11]、利用维惯性权重矩阵自适应动态调整惯性权重[12]以及基于云自适应梯度的惯性权重改进策略[13]等等。随着时间的推移,不少研究者又提出了许多其它的 PSO 改进算法和混合算法,这些算法从不同的方面,在一定程度上弥补了原算法的缺陷,大大促进了 PSO 的发展和完善。

针对粒子群算法中较低的种群多样性会导致算法早熟收敛,从而陷入局部最优解的缺陷,提出了一种不同粒子惯性权重 w 的值,其不仅随迭代次数增加而递减,并且随与全局最优点距离的大小而动态变化;为保持粒子群多样性,又引入了杂交变异因子,即基于杂交变异的动态粒子群优化算法(Dynamic particle swarm optimization based on hybrid variable, HV-DPSO)。通过使用 4 个典型的基准函数进行数值实验,证明了该方法不但可以有效地避免早熟收敛,而且具有更好的收敛效果。

本文第 1 节介绍了 PSO 算法发展和研究现状;第 2 节介绍了标准粒子群算法;第 3 节针对 PSO 算法存在的问题,提

本文受四川省教育厅青年基金(11ZB058)资助。

周利军(1989—),男,硕士生,主要研究方向为数字图像处理、智能算法,E-mail:stefan_zhou@163.com;**彭 卫**(1969—),男,博士后,副教授,主要研究方向为数字图像处理、智能算法、数据挖掘;**曾小强**(1980—),博士后,讲师,主要研究方向为数字图像处理、智能算法;**邹 芳**(1982—),女,博士,讲师,主要研究方向为智能算法。

出了一种基于杂交变异的动态粒子群优化算法，并给出了该算法的具体改进策略和步骤；最后是数值仿真与结果分析。

2 PSO算法

PSO算法是一种群智能算法，它通过多个粒子的相互合作来进行求解。在该算法中，每个粒子都是一个潜在的解，所有粒子均由一个优化函数决定其适应值。在搜索过程中，每个粒子经历过的最好位置称为个体极值，所有粒子经历过的最好位置称为全局极值。在计算时，首先，在一定范围内初始化一批粒子；然后，所有粒子根据这两个极值和自己的惯性进行更新。

设 $x_i=(x_1,x_2\cdots,x_d,\cdots,x_D)$ 表示第 i 个粒子的位置矢量，其中 $i=(1,2,\cdots,m)$；$v_i=(v_{i1},v_{i2},\cdots,v_{id},\cdots,v_{iD})$ 为粒子 i 的飞行速度矢量；第 i 个粒子的历史最好点用 $P_i=(P_{i1},P_{i2},\cdots,P_{id},\cdots,P_{iD})$ 表示；群体内所有粒子所经过的最好的点表示为 $P_g=(P_{g1},P_{g2},\cdots,P_{gd},\cdots,P_{gD})$。

在每一次迭代中，粒子的位置和速度根据如下方程进行更新：

$$v_{id}^{k+1}=v_{id}^k+c_1r_1(P_{id}-x_{id}^k)+c_2r_2(P_{gd}-x_{id}^k) \tag{1}$$

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

其中，$i=1,2,\cdots,m$，$d=1,2,\cdots,D$；$w_{\max}$ 为初始权重；$w_{\min}$ 为最终权重；$iter_{\max}$ 为最大迭代次数；k 为当前迭代次数；c_1，c_2 称为学习因子，也叫加速因子；r_1，r_2 是在[0,1]区间内均匀分布的随机数。

文献[6]为了更好地控制粒子群算法的开发和探测能力，将惯性权重 w 引入到式(1)中，惯性权重定义如下：

$$w=w_{\max}-\frac{k(w_{\max}-w_{\min})}{iter_{\max}} \tag{3}$$

因此，式(1)更新为：

$$v_{id}^{k+1}=wv_{id}^k+c_1r_1(P_{id}-x_{id}^k)+c_2r_2(P_{gd}-x_{id}^k) \tag{4}$$

w 较大时，v_{id} 靠近全局最优点 P_{gd} 和历史最优点 P_{id} 的机会较小，v_{id} 会偏离当前最优点，从而有可能探索更好的最优点。因此，现在采用的最好方法是 w 随迭代递减的方法，即在早期选择很大的 w 值，晚期则选择较小的 w 值。这符合早期需要粒子的探索能力，而晚期需要保持其收敛性的要求。这也是目前被普遍认可的标准粒子群算法。

标准粒子群优化算法流程如下所述：

1)随机初始化种群中各个粒子速度和位置，并且将个体的历史最优 P_{id} 设为当前位置，而群体中最优的个体作为当前的 P_{gd}；

2)对每个粒子 i 的第 d 维的速度和位置分别按式(4)和式(2)进行更新；

3)在每一代的进化中，计算各个粒子的适应度函数值；

4)如果该粒子当前的适应度函数值比其历史最优值要好，利用当前位置替换个体的历史最优位置；

5)如果该粒子的历史最优比全局最优要好，那么全局最优被该粒子的历史最优所代替；

6)如果还没有达到结束条件，转到2)，否则输出 P_{gd} 并结束。

3 HV-DPSO算法

3.1 惯性权重分析

分析式(4)中的惯性权重 w，可知采用线性递减的策略是非常有利的，这符合早期需要粒子的探索能力，而晚期需要保持其收敛性的要求。在这种取值方法中，也存在一些问题。首先，如果在运行初期探测到较优点，则希望能迅速收敛于全局最优点，而 w 的线性递减减缓了算法的收敛速度；其次，在算法的运行后期，随着 w 的减小，导致全局搜索能力下降，多样性减弱，容易陷入局部最优。

此外，从式(4)右边三项可以发现，越靠近最优点的粒子，其飞行速度越依赖惯性权重 w。因此，可以让靠近最优点的部分粒子在最优点附近进行搜索。这样就必须让靠近 P_{gd} 的粒子飞行速度很小，只在 P_{gd} 领域范围进行微小探索，而不承担更大范围的探索。所以 P_{gd} 附近粒子的权重 w 应该比其他粒子小。而让其他粒子去承担更大范围的探索任务，进一步去探索可能更优的点，因而其他粒子的惯性权重 w 的值应该要比 P_{gd} 附近粒子的 w 值大。因此，惯性权重 w 的值，不仅要随着迭代次数而变化，而且针对不同粒子，要根据其与最优粒子之间的距离大小而变化。

3.2 惯性权重改进策略

从分析中可知，应设计一种权重 w 随各粒子与最优粒子距离不同而动态变化的粒子群算法。具体的设计思想是：各不同粒子惯性权重 w 的值，不仅随迭代次数增加而递减，并且应该随与全局最优点距离的大小而动态变化，即权重 w 随粒子不同而动态变化。

假设 D_{ki} 表示第 k 次迭代后粒子 i（当前全局最优粒子除外）到当前全局最优 P_{gd} 的欧氏距离，$\max D_k$ 表示第 k 次迭代后 D_{ki} 的最大值，$\min D_k$ 表示第 k 次迭代后 D_{ki} 的最小值，那么权重 w 根据其与最优粒子之间的距离大小变化而动态变化的规律如下：

$$w_d=w_{\min}+\frac{(D_{ki}-\min D_k)(w_{\max}-w_{\min})}{\max D_k-\min D_k} \tag{5}$$

同时惯性权重随迭代次数变化情况由式(3)可得：

$$w_k=w_{\max}-\frac{k(w_{\max}-w_{\min})}{iter_{\max}} \tag{6}$$

结合粒子与最优粒子之间距离而动态变化的规律以及随迭代次数变化的两种情况，由式(5)和式(6)可得：

$$w=aw_d+bw_k \text{ and } a+b=1 \tag{7}$$

式中，w_d 表示粒子与最优粒子之间的距离大小变化而动态变化的惯性权重；w_k 表示粒子随迭代次数变化情况的惯性权重；a 是一个常数，在本文中取 $a=0.45$。

3.3 杂交变异算子

由于式(5)的引入，随着迭代次数增加，最优粒子的控制度会逐渐加强，导致整个粒子种群的多样性呈下降趋势，从而种群更易陷入局部最优。因此，本文引入杂交变异算子，当全局最优连续 N 代（本文中取 $N=5$）没有进化时，说明种群多样性下降可能陷入了局部最优。为了增加种群多样性，选取种群中 M 个较优粒子（包括最优粒子）与新产生的粒子进行随机杂交（本文中取 $M=15$），从而改变最优粒子的前进方向

和速度，使其进入其它区域进行搜索，发现新的个体极值。杂交过程定义如下：

$$P_{new}(x_i)=c*P_{initialize}(x_i)+(1-c)*P_{old}(x_i) \quad (8)$$

式中，c 是一个[0,1]之间的随机变量，这个杂交概率由随机量 c 确定，与粒子的适应值无关。

3.4 HV-DPSO 算法的基本流程

1)随机初始化种群中各粒子速度和位置，并且将个体的历史最优 P_{id} 设为当前位置，而群体中最优的个体作为当前的全局最优 P_{gd}；

2)计算各个粒子与当前全局最优 P_{gd} 的距离，获得 $\min D_k$ 和 $\max D_k$ 的值，然后由式(7)更新下一次迭代过程中每个粒子的惯性权重；

3)对每个粒子的速度和位置分别按式(4)和式(2)进行更新；

4)在每一代的进化中，计算各个粒子的适应度函数值；

5)如果该粒子当前的适应度函数值比其历史最优值要好，利用当前位置替换个体的历史最优位置；

6)如果该粒子的历史最优比全局最优要好，那么全局最优将会被该粒子的历史最优所代替；

7)判断种群是否满足杂交变异条件，若满足条件就执行8)，否则就执行9)；

8)产生新粒子并根据式(8)进行杂交变异，替换部分较优的粒子，转到2)；

9)判断是否满足终止条件，如果不满足转到2)，否则输出 P_{gd} 并结束。

4 数值实验及结果分析

为了证明所提方法的有效性，本文通过测试4个典型的基准测试函数[3]优化问题(求解最小值)来加以验证。4个函数分别是：Sphere 函数 f_1、Griewank 函数 f_2、Ackley 函数 f_3 和 Rastrigin 函数 f_4。

$$f_1=\sum_{i=1}^{30}x_i^2,\ x_i\in[-100,100]$$

其中，全局最小值 $\min(f)=f(0,0,\cdots,0)=0$。

$$f_2=\sum_{i=1}^{30}x_i^2/4000-\prod_{i=1}^{30}\cos(|x_i/\sqrt{i}|)+1,\ x_i\in[-600,600]$$

其中，全局最小值 $\min(f)=f(0,0,\cdots,0)=0$。

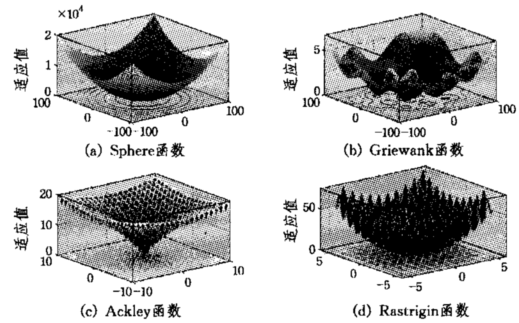
$$f_3=-20e^{-0.2\sqrt{\sum_{i=1}^{n}x_i^2/n}}-e^{\sum_{i=1}^{n}\cos 2\pi x_i/n}+20+e,\ x_i\in[-32,32]$$

其中，全局最小值 $\min(f)=f(0,0,\cdots,0)=0$。

$$f_4=\sum_{i=1}^{30}(x_i^2-10\cos(2\pi x_i)+10),\ x_i\in[-5.12,5.12]$$

其中，全局最小值 $\min(f)=f(0,0,\cdots,0)=0$。

4个测试函数空间特征如图1所示。



(a) Sphere函数 (b) Griewank函数

(c) Ackley函数 (d) Rastrigin函数

图1 4个测试函数维度为2的图像

在实验过程中，本文选取标准粒子群算法(PSO)和引入式(7)的动态粒子群算法(Dynamic Particle Swarm Optimization，DPSO)与引入式(7)、式(8)的基于杂交变异的动态粒子群算法(HV-DPSO)进行对比。所有实验均在 Windows 7 操作系统下并具有双核处理器(Pentium(R) Dual-Core CPU T4400 @2.20GHz)的计算机上运行，使用 MATLAB 7.10.0 编程。算法参数设置如下：种群大小均为100，最大迭代次数1000，学习因子 c_1，c_2 为2；$w_{max}=0.9$，$w_{min}=0.4$；在 DPSO 和 HV-DPSO 中本文取 $a=0.45$；为了消除随机干扰，对每个优化函数独立运行50次。数值实验结果详见表1。

表1 数值实验结果

函数	维数	收敛次数			平均收敛代数			收敛适应值					
								PSO		DPSO		HV-DPSO	
		PSO	DPSO	HV-DPSO	PSO	DPSO	HV-DPSO	MEAN	MIN	MEAN	MEAN	MIN	MEAN
f1	2	48	50	45	568	333	378	1.55E-26	1.23E-37	3.81E−28	3.03E−62	2.29E−27	8.96E−86
	5	50	50	50	576	335	369	4.78E−21	2.98E−27	6.07E−20	4.09E−48	4.85E−21	3.99E−63
	10	50	50	50	583	390	391	2.97E−13	2.92E−17	6.92E−13	1.37E−26	1.48E−14	1.47E−27
f2	2	44	49	47	556	326	309	0	0	0	0	0	0
	5	50	50	50	562	338	331	0	0	0	0	0	0
	10	50	50	50	574	348	393	2.11E−13	1.11E−16	1.86E−11	0	9.76E−13	1.11E−16
f3	2	50	50	50	553	346	332	1.64E−13	8.88E−16	1.62E−13	8.88E−16	2.54E−13	8.88E−16
	5	50	48	50	559	368	352	1.01E−10	6.84E−14	3.78E−11	8.88E−16	3.75E−11	8.88E−16
	10	9	17	17	577	369	369	1.31E−06	2.20E−07	5.15E−07	7.20E−08	5.15E−07	7.20E−08
f4	2	50	50	50	544	391	358	0	0	0	0	0	0
	3	49	37	37	556	464	405	0	0	0	0	0	0

从表1可知，对于所有测试函数，本文算法的优化结果都要好于 PSO，全局最优解的最小和平均都要好于 PSO；在50次的试验中 DPSO 和 HV-DPSO 保持了 PSO 的优点，同时在收敛进度和迭代次数都要明显好于 PSO。其中对于 f_2 和 f_4，本文算法都达到理论最优值，f_1 和 f_3 的收敛精度 DPSO 和 HV-DPSO 都较高。DPSO 与 HV-DPSO 相比，在迭代次数方面，对于 f_1，DPSO 好于 HV-DPSO，而对于 f_2、f_3 和 f_4，HV-DPSO 都表现出了较好的优良性。

为了更直观地了解3种算法的性能差异，图2至图5给出了4个测试函数对于3种算法的平均最优函数值进化曲线图。从曲线图中可以看出对于 f_1，f_2，f_3 和 f_4，HV-DPSO 和 DPSO 相对于 PSO 来说，克服了早熟收敛，并且收敛速度加

快，其中对于 f_1 和 f_2，HV-DPSO 在收敛速度上明显高于 PSO；对于 f_1，f_2，f_3 和 f_4，收敛次数和收敛速度 DPSO 和 HV-DPSO 有较大优势，随着测试函数的复杂度增加，DPSO 的优势逐渐下降，而 HV-DPSO 的优势基本保持不变；对于 f_3 和 f_4，杂交变异算子的引入，体现了在多维复杂函数方面，进一步提高了算法的性能。实验表明，在综合性能方面，改进后的 DPSO 算法和 HV-DPSO 算法都好于 PSO 算法，HV-DPSO 算法要好于 DPSO 算法。

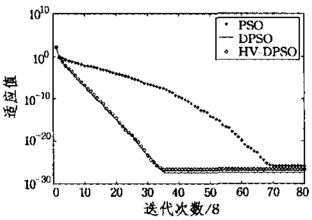


图 2 Sphere 函数收敛对比分析

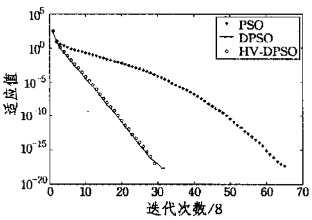


图 3 Griewan 函数收敛对比分析

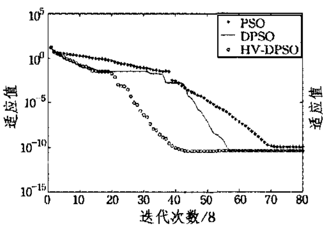


图 4 Ackley 函数收敛对比分析

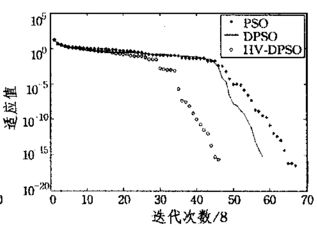


图 5 Rastrigin 函数收敛对比分析

结束语 粒子群算法中，较低的种群多样性会导致算法早熟收敛，陷入局部最优解。本文引入动态变化的惯性权重以及杂交变异规则来增加其多样性，控制种群多样性处于高效的范围内，通过设置比例系数，控制迭代次数与惯性权重的关系、其他粒子与最优粒子之间距离的关系。在保持种群多样性的基础上，实现算法全局与局部搜索的平衡。采用 4 个常用基准函数进行了试验，并与标准 PSO 做了比较，结果表明改进算法相对于标准 PSO 算法，克服了早熟收敛而且具有较好的收敛精度。

参考文献

[1] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[C]//Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Japan: Na-goya, 1995: 39-43
[2] Kennedy J, Eberhart R C. Particle swarm optimization[C]//Proceedings of the IEEE International Conference on Neural Networks. Piscataway: IEEE, 1995: 1942-1948
[3] 纪震，廖慧连，吴青华. 粒子群算法及应用[M]. 北京：科学出版社，2010
[4] 田雨波，朱人杰，薛权祥. 粒子群优化算法中惯性权重的研究进展[J]. 计算机工程与应用，2008，44(23)：39-41
[5] 唐忠. 粒子群算法惯性权重的研究[J]. 广西大学学报：自然科学版，2009，34(5)：640-644
[6] Shi Y, Eberhart R C. A modified particle swarm optimizer[C]//IEEE World Congress on Computational Intelligence. Piscataway: IEEE, 1998: 69-73
[7] 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
[8] 徐玉杰，仇雷，刘清. 自适应惯性权重的混沌粒子群算法研究[J]. 南京师范大学学报：工程技术版，2012，12(1)：64-69
[9] 王克华，牛慧，张亚南，等. 一种参数自适应调整和边界约束的粒子群算法[J]. 电子设计工程，2011，19(21)：46-49
[10] 盛桂敏，薛玉翠，张博阳. 动态自适应粒子群优化算法[J]. 绥化学院学报，2011，31(6)：190-192
[11] 张顶学，关治洪，刘新. 一种动态改变惯性权重的自适应粒子群算法[J]. 控制与决策，2008，23(11)：1253-1257
[12] 龙文，梁昔明，董淑华，等. 动态调整惯性权重的粒子群优化算法[J]. 计算机应用，2009，29(8)：2240-2242
[13] 祝洪博，徐刚刚，海冉冉，等. 基于云自适应梯度粒子群算法的无功优化[J]. 电网技术，2012，36(3)：162-167

(上接第 142 页)

[3] Zhang S C. Decision tree classifiers sensitive to heterogeneous costs[J]. The Journal of Systems and Software, 2012, 85: 771-779
[4] Elkan C. The foundations of cost-sensitive learning[C]//Proceeding of the Seventeenth International Joint Conference of Artificial Intelligence. Morgan Kaufmann, Seattle, August 2001: 973-978
[5] Nunez M. The use of background knowledge in decision tree induction[J]. Machine Learning, 1991, 6: 231-250
[6] Tan M, Schimmer J. Cost-sensitive concept learning of sensor use in approach and recognition[C]//Proceedings of the 6th International Workshop on Machine Learning. Ithaca, New York, 1989: 392-395
[7] Freitas A, Costa-Pereira A, Brazdil P. Cost - sensitive decision trees applied to medical data[C]//Proceedings of DaWak-2007, LNCS 4654. 2007: 303-312
[8] Davis J V, Ha J, Rossbach C J, et al. Cost-sensitive decision tree learning for forensic classification[C]//Proceedings of the 17th European Conference on Machine Learning (ECML). 2006: 622-629
[9] Zhang S C, Jin Z, Zhu X F. Missing Data Imputation by Utilizing Information within Incomplete Instances[J]. Journal of Systems & Software, 2011, 84: 452-459
[10] Wang W D, Miao S, Yang J Y. Classifier Algorithm on Orthogonal Projection on[J]. Computer Scienc, 2011, 38(5): 1-4
[11] Wu H R, Qin J, Zheng B J. Anti-attack Ability Based on Costs in Complex Network[J]. Computer Scienc, 2012, 39(8): 1-4
[12] Jan T K, Lin C H, Wang D W, et al. A Simple Methodology for Soft Cost-sensitive Classification KDD[Z]. Beijing, 2012, 8