适应度排序改进惯性权重的粒子群算法

陶俊波[1,2],吴彰敦[1],蔡德所[1,3]
TAO Jun-bo[1,2],WU Zhang-dun[1],CAI De-suo[1,3]

1.广西大学 土木建筑工程学院,南宁 530004
2.福建水利电力职业技术学院,福建 永安 366000
3.三峡大学 土木水电学院,湖北 宜昌 443002
1.Department of Civil and Architecture Engineering,Guangxi University,Nanning 530004,China
2.Fujian College of Water Conservancy and Electricity Power,Yong'an,Fujian 366000,China
3.College of Civil and Hydropower Engineering,Three Gorges University,Yichang,Hubei 443002,China

TAO Jun-bo,WU Zhang-dun,CAI De-suo.Use adaptation sequence modified inertia weight particle swarm optimization. Computer Engineering and Applications,2009,45(14):53-55.

Abstract: It has modified PSO inertia weight.The inertia weight not only through its longitudinal linear change by generation;but also think about current and up to now best results of particles adaptation sequence's lateral linear change which rearrange by adaptation good or bad.The lateral linear upper bound un-change and lower bound become small,so the lateral linear range expend gradually.The inertia weight will generate more negative value by generation increasing.That obtain use adaptation sequence modified inertia weight particle swarm optimization(ASMIWPSO).It has been through simulation to explain ASMIWPSO.And make contrastive analysis on the results of PSO and ASMIWPSO by Rastrigrin function.It shows that the ASMIWPSO has better optimization results.

Key words: particle swarm optimization;inertia weight;adaptation sequence

摘 要:改进 PSO 算法的惯性权重。惯性权重不仅随代数纵向线性变化,也根据当前和迄今粒子的适应度重排序横向线性变化。横向线性变化上限不变,下限逐渐减小,使得横向线性变化数值范围随代数逐渐增大。惯性权重数值随着代数逐渐取负,并且适应度差的粒子取负的几率更大。得到基于粒子适应度排序改进惯性权重的粒子群算法(ASMIWPSO 算法)。通过仿真学解释ASMIWPSO 算法。Rastrigrin 函数测试对比 ASMIWPSO 算法、PSO 算法,说明 ASMIWPSO 算法具有更好的优化结果。

关键词:粒子群算法;惯性权重;适应度排序
DOI:10.3778/j.issn.1002-8331.2009.14.014 **文章编号**:1002-8331(2009)14-0053-03 **文献标识码**:A **中图分类号**:TP301.6

1 引言

粒子群优化算法(Particle Swarm Optimization,PSO)是近年来发展较快的一种基于群智能的进化算法,最早由 Eberhart 和 Kennedy 于 1995 年提出[1]。PSO 算法与遗传算法类似,同样基于群体与适应度的概念,从随机解出发,通过迭代寻找最优解。与遗传算法相比较,该算法具有算法简单、实现容易同时又有深刻的智能背景、收敛速度快、需要调整参数少,全局寻优能力强等优点,因此倍受学术界的重视。PSO 算法可用于求解大量非线性、不可微、多峰值及多目标复杂优化问题,它不依赖于所求问题的具体领域,而是直接以决策变量的编码作为运算对象,以适应度函数值为搜索目标,且同时可以使用多个搜索点的信息,已应用于多个学科和工程邻域。

考虑到 PSO 算法在实际应用中存在的问题,各学科学者对粒子群算法进行了大量改进:通过借鉴模拟退火算法(SA)和遗传算法(GA)的思想,在基本 PSO 算法的基础之上,引入了一个称为接受概率的关键参数,改写了原算法中粒子飞翔的速度公式,使粒子以一定的概率随机在解空间的某一方向上产生停滞行为[1]。利用克隆和变异等操作,提高粒子群算法收敛速度和种群的多样性[2]。分析了惯性权重对 PSO 算法优化性能的影响,进而提出选择惯性权重的新策略。在随机选取惯性权重的同时,自适应地调整随机惯性权重的数学期望,有效地调整算法的全局与局部搜索能力[3]。基于极坐标复运算的 PSO 算法,极坐标系和复数运算的采用、重叠空间搜索法的设计和粒子密度径周比(RDRC)的引入,使得 PPSO 算法更易于稳定地、快速地、智能化地实现目标寻优[4]。通过引入梯度信息来影响粒子速度的更新,构造了带有梯度加速的粒子群算法。为减小陷入局优的可能性,当群体最优信息陷入停滞时,对群体进行部分初始化来保持群体的活性[5]。带赌轮盘选择策略双种群粒子群算

作者简介:陶俊波(1982-),男,硕士,从事演化算法、工程结构可靠度计算和水工生态数学模型研究;吴彰敦(1935-),男,教授,从事工程结构可靠度、水工结构分析及高速水流研究;蔡德所(1952-),男,教授,博士,从事水利工程生态和水利工程光纤研究。

收稿日期:2008-08-19 **修回日期**:2008-09-20

法,对两个种群采取不同参数设置,使得粒子在进化过程具有不同飞行轨迹,从而尽可能的搜索空间,增强算法的全局搜索能力;基于赌轮盘的概率选择机制使粒子可以在较好的可行解临近范围内高强度搜索,增加算法的局部搜索能力[6]。根据不同粒子距离全局最优点的距离对基本 PSO 算法的惯性权重进行动态调整的新型粒子群算法(DPSO)[7]。

介绍基于 PSO 算法后,提出基于粒子适应度排序改进惯性权重的 PSO 算法(Use Adaptation Sequence Modified Inertia Weight Particle Swarm Optimization,ASMIWPSO)。

2 粒子群优化算法

算法首先初始化一群随机粒子,然后通过多次迭代找到最优解。在每一次迭代中,粒子通过跟踪两个"极值"来更新自己:一个是粒子本身迄今为止寻找到的最优值,即个体极值 p_{id}^{j};另一个是当前整个粒子群找到的最优解,称之为全局极值 p_{gd}^{j}。

$$\begin{cases} w^{j}=w_{\max}-(w_{\max}-w_{\min})j/j_{\max} \\ c_1=c_2=1.8\sim2 \\ v_{id}^{j+1}=w^{j}v_{id}^{j}+c_1r_1(p_{id}^{j}-x_{id}^{j})+c_2r_2(p_{gd}^{j}-x_{id}^{j}) \\ x_{id}^{j+1}=x_{id}^{j}+v_{id}^{j+1} \end{cases} \tag{1}$$

上标 j 表示当前代,$j+1$ 表示下一代。下标 i 表示第 i 个粒子;下标 d 表示 d 维坐标;v_{id}^{j} 为当前第 i 粒子在第 d 维度上的速度;x_{id}^{j} 为当前第 i 粒子在第 d 维度上的位置;c_1,c_2 分别为个体极值和全局极值的学习因子,通常可取(1.8~2.0)。r_1,r_2 为(0,1)之间的均匀分布随机数;w^{j} 为惯性权重,用来控制速度,较大的 w^{j} 有利于全局搜索,而较小的 w^{j} 有利于局部搜索。粒子群优化算法在解空间内搜索时,在算法后期就会出现粒子在全局最优解的附近"振荡"的现象[8]。

粒子群优化算法在解空间内搜索时,在算法后期就会出现粒子在全局最优解的附近"振荡"的现象,Shi 和 Eberhart 研究发现[9],w 较大时算法具有较强的全局搜索能力,w 较小则算法倾向于局部搜索。LDW 算法将惯性因子线性地减少,其变化公式为了避免这个问题,惯性权重可随着迭代的进行线性减小,最大值 $w_{\max}$ 至最小值 $w_{\min}$。$j_{\max}$ 为最大迭代数。一般 $w_{\max}=0.9$,$w_{\min}=0.4$[9]。

3 ASMIWPSO 算法

对于 GA 个体选择策略,Baker 提出通过个体好坏的排序(并非原始适应值)确定个体选择概率[10-11]。参考个体好坏的排序思想,ASMIWPSO 算法考虑了粒子随适应能力横向线性变化,修正惯性权重的大小,横向惯性权重大小差异随代数逐渐增大。文献[12]将整个粒子种群分成不同的两个小组,当一个小组的粒子飞向目前的最优解时,另外一个小组向反方向飞行,以避免算法陷入局部最优。而 ASMIWPSO 算法方向改变考虑了粒子的适应好坏排序,选取反方向粒子概率逐代增多。

3.1 惯性权重自适应横向线性变化

对当前代粒子 x_{id}^{j} 进行适应度排序:n 个粒子,适应度矩阵为 1 行 n 列。适应度排序矩阵 $\boldsymbol{A}^{j}$ 取 $(a^{j}+[1,2\cdots n-1,n]^{\mathrm{T}})/(a^{j}+n)$。$a^{j}=a^{1}-\dfrac{j-1}{j_{\max}-1}(a^{1}-a^{j_{\max}})$ 为惯性权重第 j 代修正系数,a^{j} 随代数线性减小。矩阵 $\boldsymbol{A}^{j}$ 元素 a_i^{j} 取值范围 $\left(\dfrac{a^{j}+1}{a^{j}+n},1\right]$。如 $a^{1}=9n$,$a^{1}-a^{j_{\max}}=8.75n$。第一代 $a_i^{1}\in(0.9,1]$,终止代 $a_i^{j_{\max}}\in(0.2,1]$,下限由 0.9 到 0.2 随代数线性减小,适应度排序矩阵元素取值范围逐渐增大,也使得局部搜索逐渐增强。

对粒子迄今最优 p_{id}^{j} 进行适应度排序:适应度排序矩阵 $\boldsymbol{B}^{j}$ 取 $(b^{j}+[1,2\cdots n-1,n]^{\mathrm{T}})/(b^{j}+n)$。$b^{j}=b^{1}-\dfrac{j-1}{j_{\max}-1}(b^{1}-b^{j_{\max}})$ 为惯性权重第 j 代修正系数,b^{j} 随代数线性减小。矩阵 $\boldsymbol{B}^{j}$ 元素 b_i^{j} 取值范围 $\left(\dfrac{b^{j}+1}{b^{j}+n},1\right]$。如 $b^{1}=19n$,$b^{1}-b^{j_{\max}}=10n$。第一代 $b_i^{1}\in(0.95,1]$,终止代 $b_i^{j_{\max}}\in(0.9,1]$。

当前粒子 x_{id}^{j} 和迄今最优粒子 p_{id}^{j} 的适应度重排序用来改进惯性权重,使得适应度好的粒子取较小的惯性权重,适应度差的粒子取较大的惯性权重。

寻找当前最优改为适应度矩阵元素 a_i^{j} 重排序。重排序后最小的 a_i^{j} 对应粒子 i 为当前最优粒子。

伪代码如下处理:

```
for i=1
for j=i+1:n
if f(i)>f(j),
temp=x(i);x(i)=x(j);x(j)=temp;
end
end
end
```

改为

```
for i=1:n
for j=i+1:n
if f(i)>f(j),
temp=a(i);a(i)=a(j);a(j)=temp;
end
end
end
```

迄今最优粒子最优解相比当前粒子随代数变化要稳定,所以 b_i^{j} 与 a_i^{j} 上限同为 1,b_i^{j} 的下限大于 a_i^{j} 的下限,因为 b_i^{j} 相对 a_i^{j} 的数值,变化范围更稳定。

3.2 改变惯性权重方向

当某个粒子陷入局部最优时,可能每次评价总是最好,但每代局部最优的干扰,某个粒子惯性权重修正取 $\dfrac{a^{j}+1}{a^{j}+n}\dfrac{b^{j}+1}{b^{j}+n}$ 会使得迭代过早收敛于局部最优。通过在一定代数后改变惯性权重方向来避免此问题。

$sign(r_0-a_i^{j}b_i^{j}0.5j/j_{\max})$ 更新惯性权重的方向。r_0 生成(0,1)的随机数。这样惯性权重的取值区间从 $w^{j}\left(\dfrac{a^{j}+1}{a^{j}+n},1\right]$ 逐渐变为 $w^{j}\left[-1,-\dfrac{a^{j}+1}{a^{j}+n}\right)\cup w^{j}\left(\dfrac{a^{j}+1}{a^{j}+n},1\right]$。迭代前期,惯性权重主要取正。惯性权重取负可能性增大。迭代后期,$j_{\max}$ 代惯性权重取正向和反向的概率相同。适应度较差的粒子惯性权重反向

的几率更大。

得到式(2)基于粒子适应度排序改进惯性权重的 PSO 算法。

$$\begin{cases} c_{1i}^{j}=c_{2i}^{j}=1.8\sim 2 \\ w_i^j=sign(r_0-a_i^j b_i^j 0.5)j/j_{\max})[w_{\max}-(w_{\max}-w_{\min})j/j_{\max}]a_i^j b_i^j \\ v_{id}^{j+1}=w_i^j v_{id}+c_{1i}^j r_1(p_{id}-x_{id})+c_{2i}^j r_2(p_{gd}-x_{id}) \\ x_{id}^{j+1}=x_{id}^j+v_{id}^{j+1} \end{cases} \quad (2)$$

3.3 仿真学解释

PSO 算法基本原理：用一群粒子代表一组优选方案，每个方案相应的各个设计变量值即一个粒子的各维坐标，模拟一只鸟的位置。粒子群优化模拟鸟群随机搜索食物，找到食物的最优策略就是搜索距离食物最近的鸟所在区域。

$w^j=w_{\max}-(w_{\max}-w_{\min})j/j_{\max}$ 改进惯性权重，鸟群逐渐接近食物时，逐代放慢飞行速度仔细搜索周围区域。

a_i^j、b_i^j动态调整惯性权重数值大小，当前距离食物近和迄今找到食物的最短距离小的鸟认为能先找到食物，所以放慢飞行速度来仔细搜索周围区域。当前距离食物远和迄今找到食物的最短距离大的鸟害怕食物会被抢去，加快飞行速度搜索周围区域。随着距离食物距离的减小，鸟的速度差异逐渐增大。

$sign(r_0-a_i^j b_i^j 0.5j/j_{\max})$更新惯性权重的方向。距离食物较远时，鸟群是能准确判别位置的大概方向，朝着确定方向飞行。距离食物越近时，对位置的精确判别越困难，改变方向来回搜索相关区域，并且越来越频繁。当前距离食物较远和迄今找到食物的最短距离大的鸟反向的可能性更大。

4 行为次数-代数关系

随机性行为次数-代数关系：随代数增大，代数间隔减小或不变，如代数间隔为 n 代，逐渐取间隔 $n-1$、$n-2$、$n-3\cdots$。代数间隔减小或不变时，一般代数间隔内的行为次数和不变或增大。使得一定代数内随机性行为次数和增加。

确定性和指导性行为次数-代数关系：随代数增大，使得一定代数内确定性和指导性行为次数和减少。

随机性行为次数-代数关系不同于 Holland 提出的 GA 个体变异率逐代增加，它是随机性行为次数-代数关系的特例-图 1 的第 1 图。行为次数-代数关系考虑一定代数内的行为次数增加和减少，不仅考虑每代行为次数，还考虑一定代数内的行为次数，行为次数甚至可取 0 和定值。图 1 给出 4 种随机性行为的代数-行为次数关系。

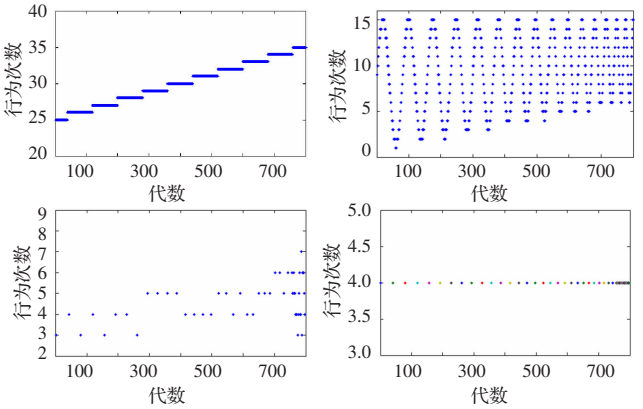

图 1　行为次数-代数关系图

5 仿真算例

仿真算例在 AMD 双核 3800，1 Gddr2 内存，Windows xp Service Pack 2 操作系统，应用 Rastrigrin 函数比较 ASMIWPSO 算法与 PSO 算法，通过 MATLAB 编程运行比较。

$$f(\boldsymbol{x})=\sum_{d=1}^{m}[x_d^2-10\cos(2\pi x_d)+10], x_d\in[-5.12,5.12] \quad (3)$$

式(3)Rastrigin 函数最优解 $x_d=0$，最优值为 0。维数 d 取 40 和 50。最大迭代 800 次，迭代终止条件：$\|\boldsymbol{x}_{gbest}-\boldsymbol{w}_{gworst}\|<\delta$。

PSO 算法种群规模 60。进行 50 次独立实验。$w_{\max}$ 取 0.8，$w_{\min}$ 取 0.3。$c_1=c_2=1.5$。

ASMIWPSO 算法：$a^1=9n$，$a^1-a^{j_{\max}}=8.75n$。$b^1=19n$，$b^1-b^{j_{\max}}=10n$。

更新权重方向满足随机性行为次数-代数关系，在代数 j 等于[2,52,101,149,196,242,287,331,374,416,457,497,536,574,611,647,682,716,749,781]$^{\mathrm{T}}$ 的元素时更新。

50 次独立实验平均最优值，最好最优值，最差最优值的结果见表 1。在不同维数下，表 1 的 ASMIWPSO 算法平均最优值、最好最优值、最差最优值均优于 PSO 算法，并且寻优结果稳定。未更新惯性权重方向 ASMIWPSO 算法收敛速度快于 PSO 算法，但优化结果不如更新惯性权重方向 ASMIWPSO 算法，只是需要更多的运算时间。

表 1　Rastrigrin 函数结果

维数	PSO 算法		ASMIWPSO 算法（更新惯性权重方向）		ASMIWPSO 算法（未更新惯性权重方向）	
	40	50	40	50	40	50
平均	158.550 4	147.710 1	57.373 9	90.535 4	63.554 7	116.500 8
最差	312.214 5	393.469 0	92.540 6	151.595 9	120.089 8	237.582 0
最好	42.233 5	68.789 3	33.860 8	43.912 4	41.864 7	60.867 6
总时间/s	809	939	938	1 116	706	734
平均终止代	770.28	754.30	800.00	799.80	649.14	608.42

6 结论及展望

ASMIWPSO 算法对 PSO 算法惯性权重数值和方向通过个体适应度排序自适应调整。并通过仿真学解释。更新惯性权重应用了行为次数-代数关系。ASMIWPSO 算法不仅通过纵向随代数的线性变化决定惯性权重，也通过适应度排序横向随粒子适应能力的线性变化决定惯性权重。速度更趋于多样化，多方面逼近于最优解，更易于搜索全局最优解而不至于陷入局部最优解。当前适应度好的粒子更适于局部搜索，适应度差更适于全局搜索。随机改变惯性权重方向，而当前迄今适应性差的粒子改变方向的可能性更大。

算例表明，ASMIWPSO 算法相比较 PSO 算法，优化结果更好。

适应度排序计算的 a_i^j、b_i^j数值控制在稳定的范围内，是否也可用于修正学习因子，使得适应度差的粒子增强学习能力，适应度好的粒子减弱学习能力。

是否可以通过惯性权重交叉的思想来找出最优的惯性权重数值，初始 $w_{\max}$ 不同即 $w_{\max}^i$，不同粒子的 $w_{\min}$ 相同，这样不同粒子具有不同的 w_i^j 通过 $w_i^j=w_{\max}^i-(w_{\max}^i-w_{\min})j/j_{\max}$ 或 $w_i^j=w_{\max}^i$ 更新，再通过当前最优粒子对应的 w_{better}^j 交叉更新 $w_i^j=w_i^j+r(w_i^j-w_{better}^j)$。笔者在另一篇文章中通过算例和工程测试，说明想法是

（下转 57 页）