基于粒子进化的多粒子群优化算法

张文爱,刘丽芳,李孝荣
ZHANG Wen-ai,LIU Li-fang,LI Xiao-rong

太原理工大学 信息工程学院,太原 030024
College of Information Engineering,Taiyuan University of Technology,Taiyuan 030024,China
E-mail:happylife21th@163.com

ZHANG Wen-ai,LIU Li-fang,LI Xiao-rong.Particle Swarm Optimization based on particle evolution.Computer Engineering and Applications,2008,44(7):51-53.

Abstract: The Particle Swarms Optimization(PSO) based on particle evolution is proposed.Location best version of PSO is adopted in the algorithm.Particle swarms are employed to search in the solution space independently that enhances the global searching ability.The location of evolutional particles will be reset in order to force it getting out of locally minimum.It makes the particle escaped from the premature convergence and increases the stability of algorithm.Comparative experiments on three testing functions indicate that the algorithm is better than the standard PSO.

Key words: Particle Swarm Optimization(PSO);evolutionary computation;swarm intelligence;location best version PSO

摘 要:提出了一种基于粒子进化的多粒子群优化算法。该算法采用局部版的粒子群优化方法,多个粒子群彼此独立地搜索解空间,从而增强了全局搜索能力;利用重置进化粒子位置的方法使陷入局部值的粒子摆脱局部最小,从而有效地避免了"早熟"问题,提高了算法的稳定性。对3个测试函数进行了对比实验,结果表明该算法优于标准粒子群算法。

关键词:粒子群算法;进化计算;群集智能;局部版粒子群算法
文章编号:1002-8331(2008)07-0051-03 **文献标识码:**A **中图分类号:**TP301.6

1 引言

粒子群算法(Particle Swarm Optimization,PSO)是Eberhart和Kennedy[1]于1995年基于对鸟群捕食行为模拟的基础上提出的一种群集智能的算法。同其它的进化算法相比,PSO算法概念简单、容易实现。因此,PSO算法一提出,立刻引起了进化计算等领域学者们的广泛关注,成为一个研究的热点。随着学者们对粒子群算法研究的不断深入,算法在优化过程中存在问题也逐渐被发现,现将其总结如下:

(1)参数控制问题:在粒子群算法中,参数对算法结果的影响较大。因此,如何选择合适的参数达到最好的优化结果是粒子群算法需要解决的问题。

(2)早熟问题:粒子在优化过程中过早收敛,使算法的寻优停滞在局部最小值,无法继续寻找全局最优解。

(3)稳定性问题:由于算法中粒子的初始位置、速度和一些参数是被随机初始化的,因此每一次算法优化的结果并不相同,有时结果的差别很大,这样就导致了算法优化结果不稳定。

为此,本文提出了一种改进的粒子群优化方法:基于粒子进化的多粒子群优化算法。通过对3个测试函数的优化,将改进的粒子群算法与标准PSO算法和一种非标准的改进方法对比,证明了该算法的有效性。

2 粒子群算法(PSO)

2.1 算法原理

Elberhart和Kennedy[1]提出的基本粒子群算法可简单描述如下:在一个D维的目标搜索空间中,有m个粒子组成一个群落,第i个粒子的位置用向量$\boldsymbol{x}_i=[x_{i1},x_{i2},\cdots,x_{iD}]$($i=1,2,\cdots,m$)表示,飞行速度用$\boldsymbol{v}_i=[v_{i1},v_{i2},\cdots,v_{iD}]$表示,第$i$个粒子搜索到的最优位置为$\boldsymbol{p}_i=[p_{i1},p_{i2},\cdots,p_{iD}]$,整个群体搜索到的最优位置为$\boldsymbol{p}_g=[p_{g1},p_{g2},\cdots,p_{gD}]$,则用下式更新粒子速度和位置:

$$\boldsymbol{v}_i(n+1)=\boldsymbol{v}_i(n)+c_1r_1(\boldsymbol{p}_i-\boldsymbol{x}_i(n))+c_2r_2(\boldsymbol{p}_g-\boldsymbol{x}_i(n)) \tag{1a}$$

$$\boldsymbol{x}_i(n+1)=\boldsymbol{x}_i(n)+\boldsymbol{v}_i(n) \tag{1b}$$

式中,$i=1,2,\cdots,m$,$d=1,2,\cdots,D$,c_1、c_2为大于零的加速因子,r_1、r_2为介于[0,1]之间的随机数,n为迭代次数。算法根据公式(1a)、(1b)进行迭代运算,当粒子的适应度值足够好或达到预设的迭代步数时,算法停止迭代,输出最优解。

式(1a)中的第1部分为粒子前一次迭代的速度;第2部分为"认知(cognition)"部分,表示粒子自身的思考;第3部分为"社会(social)"部分,表示粒子之间的信息共享与相互合作[3],三个部分共同决定了粒子的空间搜索能力。第1部分起到平衡全局和局部搜索的能力;第2部分使粒子有足够强的全局搜索

作者简介:张文爱(1967-),女,副教授,主要研究方向:盲信号处理、可编程逻辑器件;刘丽芳(1981-),女,硕士研究生,主要研究方向:粒子群算法;李孝荣(1982-),男,硕士研究生,主要研究方向:VOIP。

收稿日期:2007-06-25 **修回日期:**2007-09-28

能力,避免陷入局部极小;第3部分体现了粒子间的信息共享;这三部分的共同作用使粒子在搜索空间有较好的飞行速度。

在基本粒子群算法的基础上,1998年shi[3]等人对前面的公式(1a)进行了修正,引入惯性权重因子ω:

$$\boldsymbol{v}_i(n+1)=\omega\boldsymbol{v}_i(n)+c_1r_1(\boldsymbol{p}_i-\boldsymbol{x}_i(n))+c_2r_2(\boldsymbol{p}_g-\boldsymbol{x}_i(n)) \quad (2)$$

$$\omega=\omega_{\max}-i*\frac{\omega_{\max}-\omega_{\min}}{G_{\max}} \quad (3)$$

式(3)中,$\omega_{\max}$、$\omega_{\min}$分别为最大和最小惯性权重,n为当前迭代步数,$n_{\max}$为算法的总迭代次数。从式(3)可以看出,ω随着迭代步数的增加而减小,这样就使得算法在早期有较快的收敛速度,而在后期又有较强的局部搜索能力。ω引入使PSO算法性能有了很大提高,针对不同的搜索问题,ω可以调整全局和局部的搜索能力,这样使得PSO算法能成功的应用于很多实际问题。由此,公式(2)被视为标准PSO算法。

2.2 算法流程

标准PSO的算法流程如下:

步骤1 初始化粒子群,包括群体规模、惯性权重、粒子的初始位置和速度等;

步骤2 计算每个粒子的适应度(fitness);

步骤3 将每个粒子的fitness与其以前经历的最好位置P_{best}时的fitness比较,如果较好,则将其当前的位置作为该粒子的P_{best};

步骤4 将每一个粒子的适应度(fitness)与全体粒子所经历的最好位置g_{best}比较,如果较好,则将更新g_{best}的值;

步骤5 根据式(2)和式(1b)更新每个粒子的速度和位置;

步骤6 算法根据具体问题设定结束条件(通常为足够好的适应值或达到预设的最大迭代步数),如果没有达到预设的结束条件,则返回步骤2;如果满足预设条件,则停止迭代,输出最优解。

2.3 优化算法的测试

在进化计算领域,为了对各种算法的优化结果有一个评判,研究者们均采用一些特定的复杂函数作为算法性能的测试函数。下面将介绍在粒子群算法中经常使用的3个测试函数[5]。

第1个测试函数为Rosenbrock函数,也叫香蕉(Banana)函数,是一个经典的优化函数:

$$f_1(x)=\sum_{i=1}^{n-1}(100(x_{i-1}-x_i^2)^2+(x_i-1)^2) \quad (3a)$$

第2个测试函数为Rastrigrin函数:

$$f_2(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10) \quad (3b)$$

第3个测试函数为Griewank函数:

$$f_3(x)=\frac{1}{4\ 000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left[\frac{x_i}{\sqrt{i}}\right]+1 \quad (3c)$$

这3个函数有相同的最小值,它们均为0。其中,f_1为单峰函数,f_2、f_3为多峰函数。在粒子群算法中,它们均作为目标函数,目的是通过算法找到解空间内每一个函数的最小值。在算法的每一次迭代过程中,每个粒子所在空间位置对应的函数值即为粒子的适应度,通过粒子适应度的比较寻找全局最优的粒子。当算法结束时,全局最优粒子的适应度也就是算法最终优化的结果,将它与测试函数的最小值比较,即可确定算法性能的好坏。

3 基于粒子进化的多粒子群优化算法

标准PSO算法及其各种改进算法都是着眼于如何更有效地使粒子群在解空间中搜索到最优解。通过分析式(2)不难发现,粒子群在空间搜索时,若某个粒子发现一个当前最优位置,其它粒子就会迅速向其靠拢,若此最优位置为一局部最优而不是全局最优时,粒子就可能陷入局部极小而无法摆脱,这样就导致了粒子群不能在解空间内重新搜索新的全局极值(g_{best})。算法的这种现象称为"早熟",它限制了粒子的搜索范围。要想扩大搜索范围,就要增加种群的粒子数或者减弱粒子对当前种群搜索到的全局最优位置的追逐。增加粒子数将导致算法复杂度加大,而减弱粒子对全局最优点的追逐使得算法不易收敛。因此,本文提出了基于粒子进化的多粒子群优化算法。

粒子进化思想来源于自然界优胜劣汰、适者生存的法则。即粒子在陷入局部最优值后被淘汰,而其它粒子继续优化。其基本思想是:将种群的全部粒子分为若干个子种群,每个子种群的粒子根据自己的飞行经验和自己所属子种群的"全局最优"来调整飞行的方向。当某个粒子的适应值为最差的次数达到预设次数时,表明该粒子已经不适应目前的搜索环境,需要对其进行进化,即将该粒子的位置重置,使其朝着更好的方向进行优化,而其它的粒子保持原有状态继续优化,直到满足算法结束的条件。

在粒子群算法中,当把整个种群定义为同伴时,算法被称为全局版的粒子群算法;当同伴仅指种群中的一部分粒子时,称为局部版的粒子群算法[2]。本文提出的算法属于局部版的粒子群算法,因此将式(2)中全局最优值P_g用各子种群中的最优值P_{Lg}代替,也就是各子种群彼此独立优化,这样可以减弱粒子对局部最优点的追逐,有效地避免了"早熟"现象,使算法在搜索过程中的稳定性大大提高。下面给出改进后的粒子速度更新公式:

$$\boldsymbol{v}_i(n+1)=\omega\boldsymbol{v}_i(n)+c_1r_1(\boldsymbol{p}_i-\boldsymbol{x}_i(n))+c_2r_2(\boldsymbol{p}_{Lg}-\boldsymbol{x}_i(n)) \quad (4)$$

粒子进化的方法是围绕全局重心重置进化粒子的空间位置,这样不但可以使陷入局部值的粒子跳出继续寻找最优解,而且使进化后的粒子能迅速寻优,加快了算法的收敛速度。粒子进化的程序伪码如下:

```
// after 步骤5 in standard PSO process
If cnt(i)=Ie //判断粒子适应值为最差的次数是否达到预设次数Ie
  X(i,:)=G;  //粒子进化,重置粒子的位置
  cnt(i)=0;  //适应值记录清零
```

4 实验方法、结果及说明

为了对比改进的效果,本文参数的设定与文献[5,6]基本相同。采用不对称初始化[5],见表1。加速因子设为:$c_1=c_2=2$,每个函数的$v_{\max}$与$X_{\max}$值均相等,$V_{\max}$的设定是为了防止粒子运动速度过快而错过最优解,见表2。线性下降的惯性权重w的变化范围为[0.4,0.9]。

表1 不对称初始化的范围

函数	范围
f_1	(15,30)
f_2	(2.56,5.12)
f_3	(300,600)

表2 $V_{\max}$与$X_{\max}$的值

	$X_{\max}$	$V_{\max}$
f_1	100	100
f_2	10	10
f_3	600	600

由于粒子群算法的优化性能对粒子种群的规模和粒子的维数并不敏感[5],因此在实验中种群规模设为m=20,粒子的维

数为 Dim=10,最大迭代步数相应为 G_{max}=1 000。每一个实验运行 50 次。

表 3 为在相同的实验参数下,不同粒子群算法对 3 个标准测试函数进行优化的平均适应度值的对比结果。在表中,SPSO 代表标准 PSO 的优化结果,其优化结果来自文献[5];HPSO 为谢晓峰[6]等人对标准 PSO 做的改进,其结果来源于文献[6];MPSO 代表本文对标准 PSO 提出的改进算法,表中的结果来自于 Matlab 仿真。其中 $MPSO_1$ 表示仿真时 20 个粒子分为 4 个子种群,每个子种群由 5 个粒子,而 $MPSO_2$ 表示 20 个粒子分为 2 个子种群,每个中有 10 个粒子。

表 3　函数 f_1、f_2、f_3 的平均适应度值

函数	m	Dim	G_{max}	SPSO[6]	$HPSO_1$[7]	$HPSO_2$[7]	$MPSO_1$	$MPSO_2$
f_1	20	10	1 000	96.175 0	70.415 91	45.119 09	26.430 2	40.142 2
f_2	20	10	1 000	5.557 2	2.926 28	6.097 13	5.513 0	5.125 1
f_3	20	10	1 000	0.091 9	0.091 00	0.086 26	0.066 4	0.080 9

通过比较实验结果可以看出:不论是 $MPSO_1$ 还是 $MPSO_2$,对函数 f_1、f_3 的优化结果均优于 SPSO 和 HPSO;但是它们对函数 f_2 的优化效果并不很明显,如 $MPSO_1$ 的结果 5.513 0 与 SPSO 的 5.577 2 很接近,但是逊于 $HPSO_1$ 的 2.926 28 的优化结果。从 $MPSO_1$ 和 $MPSO_2$ 的优化结果不同可以看出,种群的数目对优化结果的有一定的影响。在实验过程中发现,$MPSO_1$ 算法的稳定性很好,在 50 次实验中几乎不会出现失败的现象,而 $MPSO_2$ 可能会在 50 次实验中出现 2~3 次的失败,因此可以认为种群数目的选择对算法搜索的稳定性有一定影响。

图 1~图 3 为函数 f_1、f_2、f_3 在相同粒子数、粒子维数和相同优化迭代数的条件下采用不同优化算法得到的寻优曲线。从图 1 可以看出,SPSO 算法在迭代到 700 步左右时已接近算法所搜索到的最优值,此后算法收敛速度缓慢,陷入了局部最优;HPSO 算法收敛速度很快,在 300 步时已经和 SPSO 迭代 700 次时结果很接近了,但是此后优化结果没有明显变动,出现了“早熟”现象;MPSO 算法的收敛速度要慢于 HPSO,但它不会陷入局部最优,经 750 次迭代后就基本搜索到全局最优值了。通过寻优曲线图 1~图 3 的观察对比可以得出:标准 PSO 算法寻优的速度最慢,且容易陷入局部最优;HPSO 算法由于采用了最小惯性权重,增加了灭绝因子,故收敛速度最好,但易“早熟”;MPSO 采用了多种群竞争机制,且增加了进化因子,故对寻优速度会有些影响,但很大程度上提高了算法收敛的能力,可以跳出局部最优,因此算法的稳定性好。

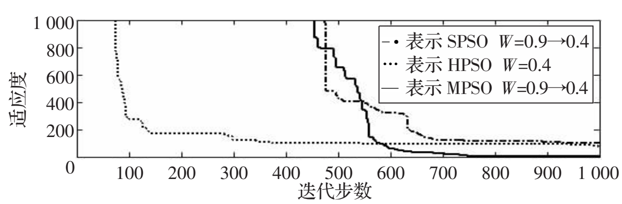

图 1　$f_1(x)$ 函数的寻优曲线图

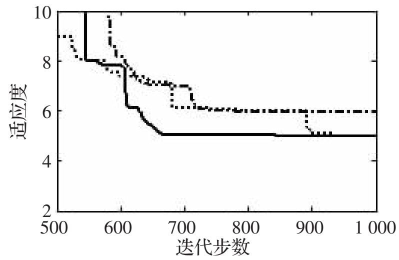

图 2　$f_2(x)$ 函数的寻优曲线图

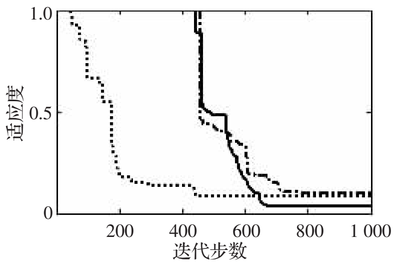

图 3　$f_3(x)$ 函数的寻优曲线图

5　小结

本文提出了一种基于粒子进化的多粒子群优化算法。该算法通过多种群并行搜索,提高了算法的寻优能力,而且适当的进化可以使在处理高维复杂问题时陷入局部最优的粒子迅速跳出,增强了算法全局搜索的能力,有效的避免了“早熟”问题。经过常用标准函数的测试证明,改进的 PSO 算法在一定程度上提高了算法的性能,为一些复杂的优化问题提供了方法。但是在算法仿真过程中发现种群的数目对算法的稳定性有一定的影响,而且算法的收敛速度相对较慢,因此,如何设定种群数目以及加快算法的收敛速度,这都是算法进一步的研究方向。

参考文献:

[1] Kernnedy J,Eberhart R C.Particle swarm optimization[C]//Proceeding of IEEE International Conference on Neutral Networks,Perth,Australia,1995:1942-1948.

[2] Eberhart R C,Kennedy J.A new optimizer using particle swarm theory[C]//Proc of the 6th International Symposium on Micro Machine and Human Science,Nagoya,Japan,1995:39-43.

[3] Shi Y,Eberhart R C.A modified particle swarm optimizer[C]//Proceedings of the IEEE Congress on Evolutionary Computation,1998:69-73.

[4] Kennedy J.The particle swarm:social adaptation of knowledge[C]//Proc IEEE Int Conf on Evolutionary Computation,1997:303-308.

[5] Shi Y,Eberhart R C.Empirical study of particle swarm optimization[C]//Proceedings of the IEEE Congress on Evolutionary Computation.Piscataway,NJ:IEEE,1999:1945-1950.

[6] Xie Xiao-feng,Zhang Wen-jun,Yang Zhi-lian.Hybrid particle swarm optimizer with mass extinction[C]//Int Conf on Communication,Circuits and Systems(ICCCAS),2002:1170-1731.

[7] Clerc M.The swarm and the queen:towards a deterministic and adaptive particle swarm optimization[C]//Proc 1999 Congress Evolutionary Computation.Piscataway,NJ:IEEE Press,1999:1951-1957.