

改进的微粒群优化算法

夏小翔

(鄂州职业大学 计算机系,湖北 鄂州 436000)

摘要:该文针对基本微粒群算法(Particle Swarm Optimizaton,简称PSO)存在的早熟收敛问题,提出了一种改善粒子活性的改进微粒群(IPSO)算法。当粒子逐步失去活性时,对粒子按一定的概率重新以一定的方式进行赋值,达到激活该粒子的目的,使粒子能够有效地进行全局和局部搜索。通过函数优化测试,结果表明IPSO算法不仅具有较快的收敛速度,而且能够有效、稳定地提高函数优化精度。

关键词:微粒群算法;进化算法;人工生命

中图分类号:TP301.6

文献标识码:A

文章编号:1008-9004(2011)05-0005-03

1 引言

由于受到人工生命(Artificial Life)研究中对鸟群社会行为的模拟所得到的结论的启发,1995年,James Kennedy和Russell Eberhart提出了著名的微粒群(Particle Swarm Optimizaton,PSO)算法^[1-2]。与其他进化算法相比,PSO算法具有收敛速度快、设置参数少、程序实现异常简洁,既适合科学研究,又特别适合工程应用^[3]。因此,PSO算法一经提出就立刻引起了与优化相关领域的学者们的广泛关注。目前微粒群算法在函数优化、神经网络训练、图像处理、模式识别等很多领域都得到了广泛的应用。

但是,由于PSO算法在优化过程中所有粒子都向最优解的方向飞去,所以粒子趋向同一化,群体的多样性逐渐丧失,致使后期算法的收敛速度明显减慢,甚至处于停滞状态,因此就难以获得较好的优化结果^[4]。

本文提出一种改进的PSO算法(IPSO),在算法搜索过程中对失去活性的粒子以一定的概率进行激活,使得进化过程中粒子群体具有较强的活力,有较强的能力逃逸出局部极值点。本文引用了2个著名的Benchmark函数进行数学实验,并与基本PSO算法(SPSO)进行对比分析,结果显示,IPSO在全局搜索能力与收敛速度方面都有明显提高。

2 微粒群优化算法

PSO算法求解优化问题时,问题的解就是搜索空间中的一只鸟的位置,称这些鸟为“粒子”。在优化过程中,PSO算法初始化一群粒子,然后通过迭代找到最优解。在每一次迭代过程中,粒子通过追逐两个极值来更新自己的位置。一个是粒子自身所找到的最优解,这个解称为个体极值pbest;另一个是整个群体当前找到的最优解,这个解称为全局极值gbest。

PSO算法数学描述如下:

种群中每个元素(可行解)称为一个微粒(particle),微粒的个数称为种群规模(size)。我们用n维向量 $X_i=(X_{i1}, X_{i2}, \dots, X_{in})^T$ 来表示第i个微粒的位置,用 $V_i=(V_{i1}, V_{i2}, \dots, V_{in})^T$ 来表示第i个微粒的速度。微粒在搜索空间飞行过程中,它自身所经历的最佳位置记为 $P_i=(P_{i1}, P_{i2}, \dots, P_{in})^T$,也称为pbest。在群体中,所有微粒经历过的最佳位置用索引符号g表示,即Pg,也称为gbest。因此,微粒在每一代中的速度和位置通过如下二个公式计算:

$$V_{id}(t+1)=wV_{id}(t)+c_1r_1(P_{id}(t)-X_{id}(t))+c_2r_2(P_{gd}(t)-X_{id}(t)) \quad (1)$$

$$X_{id}(t+1)=X_{id}(t)+V_{id}(t+1) \quad (2)$$

其中:w为惯性常数(inertia), c_1 和 c_2 为加速常数(acceleration constants), r_1 和 r_2 为两个在[0,1]范围内服从均匀分布的随机变量。公式(1)中的第一

收稿日期:2011-07-01

基金项目:鄂州职业大学青年科研项目(2009qn02)

作者简介:夏小翔(1973-),男,湖北鄂州人,鄂州职业大学计算机系讲师,研究方向:群体智能算法。

部分称为动量部分,表示微粒对当前自身状态的信任;第二部分为认知部分,代表了微粒自身的思考;第三部分为社会部分,表示微粒之间的信息共享与相互合作。

PSO算法的具体实现过程如下:

Step1 设置当前代数 $t=1$,确定粒子数 M ,在特定范围内随机初始化各微粒的位置和速度;

Step2 计算各微粒适应值;

Step3 对每个微粒 i 计算其经历的历史最好位置 P_i ;

Step4 对每个微粒 i 计算其全局最优位置 P_g ;

Step5 根据方程式(1)和(2),更新每个粒子的速度与新的当前位置,并把它们限制在一定范围内;

Step7 $t=t+1$,重复Step2至Step5,直到获得一个预期的适应值或 t 达到设定的最大迭代次数。

3 保持粒子活性的改进的微粒群算法

基本微粒群算法在早期具有较好的收敛速度,但在算法后期,由于所有粒子都向局部极小值或全局极小值收敛,此时,每个粒子的 p_{best} 、所有粒子的 g_{best} ,以及每个粒子的当前位置都会趋向于同一点,根据公式(1)很容易知道粒子的速度也趋向于0,粒子群的多样性也就逐渐丧失,致使算法的收敛速度明显变慢,甚至处于停滞状态,因而也就难以获得理想的优化结果。

为了使粒子群在进化过程中既能够快速收敛又不至于丧失群体的多样性,本文从每个粒子着手,如果能够使连续一定代数都没有取得进展的粒子重新活跃起来,那么由众多单个粒子组成的粒子群就会有较强的活力,从而有效地保持了粒子群的多样性。

为了保持粒子的活性,本文通过在迭代过程比较单个粒子的当前适应值与粒子的历史最优 p_{best} 来判断粒子是否失去活性,如果粒子在一定的演变代数 num 内都没有取得好于粒子的历史最优的适应值,则认为粒子失去活性。然后,为了激活该粒子,使粒子以较小的变异概率在其迭代过程中获得的维空间内按照一定的方式重新初始化,以此来增强全局搜索能力,克服粒子群陷入局部解而难以逃脱出来,同时又可以加快收敛速度、提高搜索精度。

失去活性粒子激活算法如下:

```
For(int i=1;i<=size;i++)
{
    If((particle_count[i]>=num)
```

```
{
    If(rand()<P)
    {
        For(int j=1;j<=wei_num;j++)
        {
            X(i,j)=(max_d(i,j)-min_d(i,j))*rand()+
            min_d(i,j);
            V(i,j)=0;
        }
    }
}
Else
{
    V_id(t+1)=wV_id(t)+c1r1(P_id(t)-X_id(t))+c2r2(P_gd(t)-X_id(t))
    X_id(t+1)=X_id(t)+V_id(t+1)
}
}
```

其中, $size$ 是种群空间大小; $particle_count[i]$ 是粒子 i 失去活性的次数, num 为设定次数的值; $rand()$ 是一个0至1之间的随机数; P 表示粒子的变异概率; wei_num 为粒子的维数; $max_d(i,j)$, $min_d(i,j)$ 分别表示迭代过程中第 i 个粒子在第 j 维的上限与下限。

4 实验仿真

4.1 测试函数

为了比较IPSO与SPSO的性能,本文通过二个典型的函数来进行测试并分析比较。二个测试函数分别如下:

(1)Rosenbrock函数:非凸、病态,在 $x_i=1$ 时达到极小值。

$$\text{MinF1}=f_1(x)=\sum_{i=1}^n(100(x_{i+1}^2-x_{i+1})^2+(1-x_i)^2)$$

其中 $x_i \in [-30,30]$

(2)Rastrigin函数:多峰,有很多正弦凸起的局部极小点,在 $x_i=0$ 时达到全局极小值。

$$\text{MinF2}=f_2(x)=\sum_{i=1}^n[x_i^2-10\cos(2\pi x_i)+10]$$

其中 $x_i \in [-5.12,5.12]$

4.2 实验设置

在二个测试函数中:函数F1最大运行代数2000, $V_{\max}=100$;函数F2最大运行代数2000, $V_{\max}=10$; $C_1=1.8$, $C_2=1.8$; $\omega=1.0-(\text{MAXGEN} \times 0.6)/(\text{MAXGEN}-1)$,其中 MAXGEN 表示迭代的最大代数; $\pi=3.1415926$; V_{\max} 与 X_{\max} 取值相同。

对于IPSO,设置失活代数 $num=5$,变异概率 $P=$

0.005;函数的值即为适应值,每种实验情况均运行50次。

4.3 结果与讨论

表1和表2显示了基本微粒群算法 (SPSO)与IPSO在不同情况下的平均最好适应值 (Mean Best Fitness, 简称MBF)和最好适应值 (Best Fitness,简称BF)。“0”后括号中的数值表示实验运行50次其中结果出现“0”的次数,size表示种群的大小,wei_num表示函数的维数,T_{max}表示算法的最大迭代次数。

从表1和表2可以看出,与SPSO算法相比,IPSO算法的收敛性不论在低维还是在高维情况下均有明显提高,表现在IPSO算法的平均最好适应值和最好适应值均优于SPSO算法。特别是对Rastrigin函数,IPSO算法能够在有限代数内较多次地收敛

到近似理论值0。而且从平均最好适应值和最好适应值的对比上,还可以看出无论对于什么函数、对于多少维数的同一函数,IPSO算法优化结果均比SPSO算法的精度要高,这也充分体现出IPSO算法的稳定性和健壮性。

从表2可以看出,IPSO算法并不能保证每次都能在有限代数内找到极小值点,这是由算法随机搜索的性质决定的。

5 结论

本文针对基本微粒群算法 (SPSO)存在的早熟问题,提出了一种保持粒子活性的改进微粒群算法 (IPSO)。通过函数优化的对比实验,可以看出IPSO算法不仅在收敛速度上,而且在优化精度上都明显优于SPSO算法,同时还显示了改进算法的稳定性。

表1 Rosenbrock函数的测试结果

ize	wei_num	T _{max}	SPSO		IPSO	
			BF	MBF	BF	MBF
0	10	1000	3. 4485	79. 4452	0. 2533	10. 2595
	20	1500	1. 4578	182. 5362	0. 9851	18. 4586
	30	2000	9. 5628	398. 8565	2. 3546	39. 1884
0	10	1000	0. 0582	48. 6518	0. 4215	12. 1546
	20	1500	0. 0651	134. 8642	1. 1020	18. 6546
	30	2000	6. 5642	225. 5302	2. 3564	21. 8025
0	10	1000	0. 9253	45. 9546	0. 3564	7. 8574
	20	1500	2. 2145	115. 1056	1. 2546	20. 1542
	30	2000	4. 5875	146. 4568	2. 1315	28. 8543

表2 Rastrigin函数的测试结果

Size	wei_num	T _{max}	SPSO		IPSO	
			BF	MBF	BF	MBF
20	10	1000	0. 8564	5. 6593	0 (25)	1. 2564
	20	1500	7. 8574	21. 5625	0 (7)	7. 6501
	30	2000	30. 2145	52. 2437	0 (1)	15. 2252
40	10	1000	0. 0005	3. 2432	0 (39)	0. 7488
	20	1500	10. 4845	18. 2815	0 (11)	4. 9827
	30	2000	22. 8750	43. 7546	0 (1)	5. 4275
60	10	1000	0. 0001	2. 4589	0 (45)	0. 2136
	20	1500	5. 4521	13. 5874	0 (27)	1. 8712
	30	2000	17. 8845	32. 2557	0 (8)	5. 9832

(下转第21页)

电子证书中应包括诸如版本号、序列号、数字签名参数、签发者名称、有效日期、持有者名称、持有者公钥等信息,既可用于身份验证又可用于发送公钥。除此之外,还可采取如下措施来加强安全保护设置:目录访问权,通过地址控制访问权。

4.5 入侵检测和网络监控技术

入侵检测系统 (Intrusion Detection System 简称:IDS)是近年来发展起来的一种防范技术,其方法是从多种计算机系统及网络系统中收集信息,再通过这此信息分析入侵特征的网络安全系统。入侵检测的作用包括威慑、检测、响应、损失情况评估、攻击预测和起诉支持。入侵检测技术是为保证计算机系统的安全而设计与配置的一种能够及

时发现并报告系统中未授权或异常现象的技术,是一种用于检测计算机网络中违反安全策略行为的技术^{[4]12-13}。

5 结束语

计算机网络在全球范围内得到了飞速发展,计算机网络安全是一个综合性的课题,涉及到技术、管理、使用和维护等多方面。网络安全和数据保护这些防范措施都有一定的限度,并不能完全保证安全和可靠。因此,在看一个内部网是否安全时不仅要考察其手段,更重要的是该网络所采取的各种措施,其中不仅是物理防范,还要对人员的素质等其他“软”的因素进行综合评估,从而得出是否安全的结论。

参考文献:

- [1]肖军模,刘军,周海刚.网络信息安全[M].北京:机械工业出版社,2003:14-15.
- [2]林建平.计算机网络安全防控策略的若干分析[J].山西广播电视大学学报,2006(6):24-25.
- [3]周海刚,邱正伦.网络主动防御安全模型及体系结构[J].解放军理工大学学报,2005(1):40-43.
- [4]苏杰.主动防御技术及其在网络安全中的应用[M].北京:中国科技信息杂志社,2005:12-13.

(上接第7页)

参考文献:

- [1]J.Kennedy,R.Eberhart. Particle swarm optimization[C]//IEEE International Conference on Neural Networks, 1995: 1942-1948.
- [2]曾建潮,介婧,崔志华.微粒群算法[M].北京:科学出版社,2004:28-30.
- [3]Eberhart R,Shi Y.Particle swarm optimization:developments,applications and resource[C]//Proc Congress on Evolutionary Computation, 2001: 39-43.
- [4]Angeline P J.Evolutionary optimization versus particle swarm optimization:Philosophy and performance differences[J].Evolutionary Programming,1998,48(17):1956-1959.