**实验三 基于粒子群优化算法辨识非线性系统**

**一、实验目的**

1. 掌握粒子群算法的基本原理和步骤。
2. 熟练使用MATLAB软件编写粒子群算法程序。
3. 利用粒子群算法解决非线性系统辨识问题。

**二、实验原理**

粒子群算法是一种群体智能算法，通过追随当前搜索到的[最优值](https://baike.baidu.com/item/%E6%9C%80%E4%BC%98%E5%80%BC" \t "https://baike.baidu.com/item/%E7%B2%92%E5%AD%90%E7%BE%A4%E7%AE%97%E6%B3%95/_blank)来寻找全局最优。该算法实现容易、精度高、收敛快，在解决实际问题中具有很大的优越性。主要步骤可描述如下：

1、初始化粒子群位置和速度。

2、计算每个粒子的适应度，确定全局最优粒子gbest和个体最优粒子pbest。

3、判断算法收敛准则是否满足，若满足，则输出搜索结果；否则执行以下步骤。

4、根据gbest和pbest更新速度分量。

5、根据更新的速度分量更新粒子位置。

6、更新全局最优粒子和个体最优粒子。

7、返回步骤③。

**三、实验内容**

1. 选择辨识对象，以拟合误差（利用辨识参数计算的函数值和原数据值的误差）的平方和作为适应度函数：

1）根据以下数据点对，辨识二阶多项式的系数a，b，c。

x=[1,3,4,5,6,7,8,9,10]; y=[10,5,4,2,1,1,2,3,4];

2）为了分析X射线的杀菌作用，采用X射线来照射细菌，每次照射次数t与存活的细菌数y如表所示，采用函数进行非线性回归，其中a表示细菌开始的数目，b表示细菌死亡的速率，辨识模型中参数a和b，画出原始数据与拟合结果的散点图，并预测t=16时残留的细菌数。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| y | 352 | 211 | 197 | 160 | 142 | 106 | 104 | 60 | 56 | 38 | 36 | 32 | 21 | 19 | 15 |

1. 确定粒子群规模、学习因子、惯性权重系数、速度范围、初始化粒子位置和速度。

3. 实现两种邻域策略：全局最优粒子群算法（星形拓扑）和局部最优粒子群算法（环状拓扑，需确定领域规模）。

1. 计算全局最优粒子gbest和个体最优粒子pbest。
2. 确定算法终止条件（指定迭代次数或者其他条件）。

6. 对每个粒子更新速度和位置，注意范围限制。

7. 更新全局最优粒子gbest和个体最优粒子pbest。

8. 达到迭代要求，输出gbest和最优解。

**四、预习要求**

1. 认真阅读教材中粒子群算法的基本原理与步骤。
2. 根据题目要求编写适应度计算函数文件。
3. 编写速度更新和位置更新程序。

**五、实验要求**

1. 控制参数自选，绘制流程图。

2. 上机编写程序并调试程序。

3. 对比gbest PSO算法和lbest PSO算法性能。

4. 绘制种群每代gbest的适应度的变化曲线。

5. 根据辨识的参数绘制曲线，并与训练数据进行对比。

6. 根据实验结果，撰写实验报告。

**六、实验报告**

1. 实验报告包括实验目的、实验原理、实验内容、实验程序、实验结果与实验分析、实验心得体会。

2. 给出完整的源程序，并对源程序中的关键内容或语句给出必要的文字注释。

3. 给出实验结果（实验结果有图的话可截图打印）。

4. 利用实验数据，分析并解答以下问题，

* 1. 两种邻域策略对算法结果的影响。
  2. 学习因子对算法性能的影响。
  3. lbestPSO邻域规模对算法性能的影响。
  4. 惯性权重对算法性能的影响。

5. 分析粒子群算法的优缺点，并提出改进意见。

6. 本实验的心得体会。

附1：

更新gbest和pbest

N

计算适应度，确定gbest和pbest

算法收敛准则满足？

更新粒子速度

输出

结果

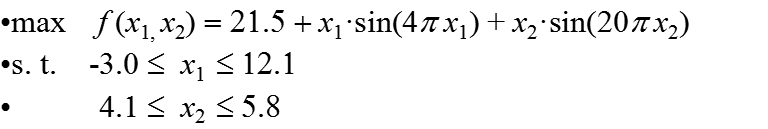
更新粒子位置

初始化粒子位置和速度

Y

图3.1 粒子群算法流程图

附2：gbest PSO例程（求二元函数极大值问题）



clear

clc

M=20;

x1=15.1\*rand(M,1)-3;

x2=1.7\*rand(M,1)+4.1;

X=[x1,x2];

c1=2;c2=2;

wmax=0.9;wmin=0.4;

Tmax=100;

v=ones(M,2);

v1m=1;

v2m=0.1;

pb=X;

%gbest产生

fmax=0;

for i=1:M

fx = f(X(i,:));

if fx>fmax

fmax=fx;

gb=X(i,:);

end

end

for t=1:Tmax

w=wmax-(wmax-wmin)\*t/Tmax;

for i=1:M

v(i,:)=w\*v(i,:)+c1\*rand(1)\*(pb(i,:)-X(i,:))+c2\*rand(1)\*(gb-X(i,:));

if v(i,1)>v1m; v(i,1)=v1m;end

if v(i,2)>v2m; v(i,2)=v2m; end

if v(i,1)<-v1m;v(i,1)=-v1m;end

if v(i,2)<-v2m;v(i,2)=-v2m;end

X(i,:)=X(i,:)+v(i,:);

if X(i,1)<-3; X(i,1)=-3;end

if X(i,1)>12.1; X(i,1)=12.1;end

if X(i,2)<4.1; X(i,2)=4.1;end

if X(i,2)>5.8; X(i,2)=5.8;end

if f(X(i,:))>f(pb(i,:))

end

for i=1:M

if f(X(i,:))>f(pb(i,:))

pb(i,:)=X(i,:);

end

if f(X(i,:))>f(gb)

gb=X(i,:);

end

end

re(t)=f(gb);

end

plot(re)

gb

附3：lbestPSO中gbest产生例程（k为邻域规模）

gb=X;

for i=1:M

for in=i-k:i+k

if in<1;

in=in+M;

elseif in>M

in=in-M;

else

in=in;

end

if f(X(in,:))>f(gb(i,:))

gb(i,:)=X(in,:);

end

end

end