实验6 粒子群算法

姓名：唐川淇 学号：1131190111

# 问题重述

使用粒子群算法求解下面函数的极小值，,其中,已知惯性权重w=1,加速系数c1=c2=2,种群规模N=3。

# 问题分析

使用粒子群算法求解函数的最小值，其中将任意一个解作为一个粒子，设定好初始参数后迭代运算得到最终的最优值。

# 模型的建立与求解

## 初始化

根据问题要求，首先初始化种群数量为3，同时设定最大迭代次数为100。

|  |
| --- |
| 代码 |
| %设置问题相关变量  popsize=3;%个体数量  degree=100;%循环次数 |

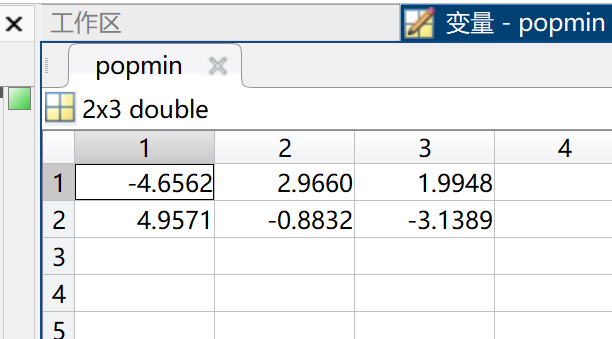
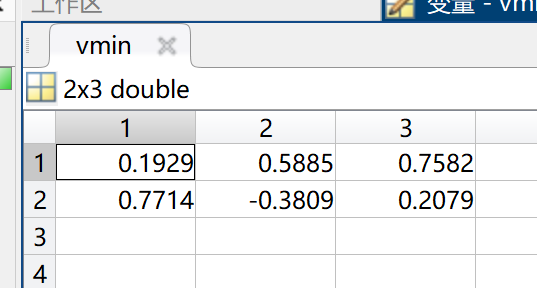
根据题目所给的变量范围，随机生成3个位置和速度，随机初始化函数如下：

|  |
| --- |
| 代码 |
| %对粒子群进行初始化  function [ pop,v ] = Init( size )  for i=1:size  %初始化位置  pop(1,i)=-5+rand(1,1)\*10;  pop(2,i)=-5+rand(1,1)\*10;  %初始化速度  v(1,i)=1\*(rand(1,1)\*2-1);  v(2,i)=1\*(rand(1,1)\*2-1);  end  end |

在主函数中调用随机初始化函数，得到初始速度和位置：

|  |
| --- |
| 代码 |
| [popmin,vmin]=Init(popsize);%初始化求最小值群体的位置和速度 |

生成的数据如下：



## 群体迭代

首先计算群体的初始适应值和群体的最优个体，设定的函数如下：

|  |
| --- |
| 代码 |
| %CALFITVALUE 计算适应度值  function [ fitvalue,gbestvalue,gbest ] = calfitvaluemin( pop )  [px,py]=size(pop);  %计算所有个体的适应度值  for i=1:py  fitvalue(i)=pop(1,i)^2+2\*pop(2,i)+3;  end  %每次循环中的群体最优  [gbestvalue,gbestindex]=min(fitvalue);  gbest=pop(:,gbestindex); |

主函数中调用：

|  |
| --- |
| 代码 |
| %计算初始群体的适应度,群体最优个体  [value\_min,gbestvaluemin,gbestmin]=calfitvaluemin(popmin);  %将个体历史最优先设为初始化值  pbestmin=popmin; |

循环如下操作：根据公式计算出新的位置以及速度，计算新的适应值，更新个体最优以及群体最优。主函数部分如下所示：

|  |
| --- |
| 代码 |
| for i=1:degree  %产生新的位置  [newpopmin,newvmin]=updatepop(popmin,vmin,pbestmin,gbestmin);  %更新速度值  vmin=newvmin;  %更新位置值  popmin=newpopmin;  %计算新的适应度值，该次循环中的群体最优  [newvalue\_min,newgbestvaluemin,newgbestmin]=calfitvaluemin(newpopmin);  %更新个体历史最优  for j=1:popsize  if newvalue\_min(j)<value\_min(j)  pbestmin(:,j)=newpopmin(:,j);  end  end  %更新群体最优  if newgbestvaluemin<gbestvaluemin  gbestvaluemin=newgbestvaluemin;  gbestmin=newgbestmin;  end  end |

其中更新位置以及速度的函数如下所示：

|  |
| --- |
| 代码 |
| %更新位置和速度  function [ newpop,newv ] = updatepop( pop,v,pbest,gbest )  [px,py]=size(pop);  %将最大速度绝对值设置为2  vmax=2;  for k=1:py  %更新速度  newv(:,k)=1\*v(:,k)+2\*rand(1,1)\*(pbest(:,k)-pop(:,k))+...  2\*rand(1,1)\*(gbest-pop(:,k));  %将超出最大速度的速度值设为最大速度  if abs(newv(:,k))>vmax  if newv(:,k)<0  newv(:,k)=-vmax;  end  if newv(:,k)>0  newv(:,k)=vmax;  end  end  %更新位置  newpop(:,k)=pop(:,k)+newv(:,k);  %对超出定义域的位置取定义域边界  if newpop(1,k)<-5  newpop(1,k)=-5;  end  if newpop(1,k)>5  newpop(1,k)=5;  end  if newpop(2,k)<-5  newpop(2,k)=-5;  end  if newpop(2,k)>5  newpop(2,k)=5;  end  end  end |

计算适应度值的函数刚才已经给出，迭代至最大迭代次数后循环结束。

## 模型结果

|  |  |
| --- | --- |
| 画出适应度值变化图如下：    同时得到最优的个体和最小值，如下图所示： |  |

附录

|  |
| --- |
| 完整代码 |
| %PSO  clc  clear    %设置问题相关变量  popsize=3;%个体数量  degree=100;%循环次数    [popmin,vmin]=Init(popsize);%初始化求最小值群体的位置和速度    %计算初始群体的适应度,群体最优个体  [value\_min,gbestvaluemin,gbestmin]=calfitvaluemin(popmin);    %将个体历史最优先设为初始化值  pbestmin=popmin;    %创建图  set(gcf,'doublebuffer','on')  set(gcf,'Name','粒子群算法PSO演示')  axis([-5 5 -5 5])  grid on    temp=zeros(1,degree);    for i=1:degree  %产生新的位置  [newpopmin,newvmin]=updatepop(popmin,vmin,pbestmin,gbestmin);    %更新速度值  vmin=newvmin;    %更新位置值  popmin=newpopmin;    %计算新的适应度值，该次循环中的群体最优  [newvalue\_min,newgbestvaluemin,newgbestmin]=calfitvaluemin(newpopmin);    %更新个体历史最优  for j=1:popsize  if newvalue\_min(j)<value\_min(j)  pbestmin(:,j)=newpopmin(:,j);  end  end    %更新群体最优  if newgbestvaluemin<gbestvaluemin  gbestvaluemin=newgbestvaluemin;  gbestmin=newgbestmin;  end    %绘制动态运动散点图  plot(newpopmin(1,:),newpopmin(2,:),'go');  drawnow    temp(i)=gbestvaluemin;  end  %适应度曲线  x=1:degree;  plot(x,temp); |

|  |
| --- |
| 完整代码 |
| %对粒子群进行初始化  function [ pop,v ] = Init( size )  for i=1:size  %初始化位置  pop(1,i)=-5+rand(1,1)\*10;  pop(2,i)=-5+rand(1,1)\*10;  %初始化速度  v(1,i)=1\*(rand(1,1)\*2-1);  v(2,i)=1\*(rand(1,1)\*2-1);    end  end |

|  |
| --- |
| 完整代码 |
| %CALFITVALUE 计算适应度值  function [ fitvalue,gbestvalue,gbest ] = calfitvaluemin( pop )    [px,py]=size(pop);    %计算所有个体的适应度值  for i=1:py  fitvalue(i)=pop(1,i)^2+2\*pop(2,i)+3;  end    %每次循环中的群体最优  [gbestvalue,gbestindex]=min(fitvalue);  gbest=pop(:,gbestindex); |

|  |
| --- |
| 完整代码 |
| %更新位置和速度  function [ newpop,newv ] = updatepop( pop,v,pbest,gbest )  [px,py]=size(pop);  %将最大速度绝对值设置为2  vmax=2;  for k=1:py  %更新速度  newv(:,k)=1\*v(:,k)+2\*rand(1,1)\*(pbest(:,k)-pop(:,k))+...  2\*rand(1,1)\*(gbest-pop(:,k));  %将超出最大速度的速度值设为最大速度  if abs(newv(:,k))>vmax  if newv(:,k)<0  newv(:,k)=-vmax;  end  if newv(:,k)>0  newv(:,k)=vmax;  end  end  %更新位置  newpop(:,k)=pop(:,k)+newv(:,k);  %对超出定义域的位置取定义域边界  if newpop(1,k)<-5  newpop(1,k)=-5;  end  if newpop(1,k)>5  newpop(1,k)=5;  end  if newpop(2,k)<-5  newpop(2,k)=-5;  end  if newpop(2,k)>5  newpop(2,k)=5;  end  end  end |