智能计算第四次实验报告

第五次实验 粒子群算法实验题目

分析求解以下三个问题的粒子群算法原理，并利用粒子群算法实现该问题。

1.北京等六城市的距离表如下。请据此表求解TSP（Travelling Salesman Problem） 问题。即给定N个城市，从某一城市开始走遍所有城市但不许重复，最后回到出发点，要求总路径最短。

表1 六城市间距离表（单位km）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 上海 | 天津 | 石家庄 | 太原 | 呼和浩特 |
| 北京 | 1078 | 119 | 263 | 398 | 401 |
| 上海 | 0 | 963 | 989 | 1096 | 1381 |
| 天津 |  | 0 | 262 | 426 | 504 |
| 石家庄 |  |  | 0 | 171 | 394 |
| 太原 |  |  |  | 0 | 341 |

1. 下面展示了13个城市之间的距离，现要求从城市A出发，找一条最短的旅游顺序，使得游览所有城市后回到城市A。

表2 13个城市间距离表（单位km）

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | G | H | I | J | K | L | M |
| A | 0 | 2.2 | 18.1 | 4.8 | 9.2 | 8.4 | 5.2 | 0.4 | 9.3 | 11.3 | 10.2 | 0.4 | 10.4 |
| B | 2.2 | 0 | 17.5 | 2.8 | 7.9 | 6.6 | 3.3 | 1.8 | 8.5 | 9.8 | 8.7 | 2.4 | 9.6 |
| C | 18.1 | 17.5 | 0 | 14.7 | 9.6 | 10.9 | 14.2 | 18.2 | 9.1 | 7.6 | 8.7 | 18.3 | 8.0 |
| D | 4.8 | 2.8 | 14.7 | 0 | 5.4 | 3.8 | 0.5 | 4.4 | 6.4 | 7.0 | 6.0 | 5.1 | 7.4 |
| E | 9.2 | 7.9 | 9.6 | 5.4 | 0 | 2.4 | 4.9 | 9.0 | 1.7 | 2.3 | 1.2 | 9.5 | 2.2 |
| F | 8.4 | 6.6 | 10.9 | 3.8 | 2.4 | 0 | 3.3 | 8.1 | 4.1 | 3.3 | 2.5 | 8.7 | 4.6 |
| G | 5.2 | 3.3 | 14.2 | 0.5 | 4.9 | 3.3 | 0 | 4.8 | 6.0 | 6.6 | 5.5 | 5.5 | 6.9 |
| H | 0.4 | 1.8 | 18.2 | 4.4 | 9.0 | 8.1 | 4.8 | 0 | 9.2 | 11.1 | 10.0 | 0.6 | 10.3 |
| I | 9.3 | 8.5 | 9.1 | 6.4 | 1.7 | 4.1 | 6.0 | 9.2 | 0 | 3.3 | 2.5 | 9.6 | 1.1 |
| J | 11.3 | 9.8 | 7.6 | 7.0 | 2.3 | 3.3 | 6.6 | 11.1 | 3.3 | 0 | 1.2 | 11.7 | 2.8 |
| K | 10.2 | 8.7 | 8.7 | 6.0 | 1.2 | 2.5 | 5.5 | 10.0 | 2.5 | 1.2 | 0 | 10.5 | 2.5 |
| L | 0.4 | 2.4 | 18.3 | 5.1 | 9.5 | 8.7 | 5.5 | 0.6 | 9.6 | 11.7 | 10.5 | 0 | 10.7 |
| M | 10.4 | 9.6 | 8.0 | 7.4 | 2.2 | 4.6 | 6.9 | 10.3 | 1.1 | 2.8 | 2.5 | 10.7 | 0 |

3.寻找函数y=1-cos(3\*x)\*exp(-x)在[0,4]最大值。

三、实验基本原理

3.1 TSP问题描述

TSP问题是典型的组合优化问题。给定个城市以及各个城市之间的距离，要求找到一条遍历所有城市且每个城市只被访问一次的路线，并使得总路线距离最短。其可用数学式来表达：

设有一个城市集合，其中每对城市之间的距离。求一对经过中每个城市一次的路线。使

其中是的一个置换。

3.2 粒子群算法概述

粒子群(PSO）算法最早是由Eberhart和Kennedy于1995年提出，其是智能计算领域中除了蚁群算法，鱼群算法外的一种群体智能的优化算法，它的基本概念源于对鸟群觅食行为的研究。鸟群中有个体和群体，个体和群体的信息是可以互通的。个体在随机搜寻食物的过程中，只要跟踪离食物最近的群体，就能最有效地找到食物。

四、程序简述

4.1 用粒子群算法求解TSP问题

4.1.1问题的求解

在TSP问题中，粒子的位置可以使用路径来表示。而速度的表示需要进行优化，原本的公式已经不再适用。以下给出基本粒子群算法速度位置更新公式：

基本粒子群算法速度位置更新公式：

4.2用遗传算法求解TSP问题

4.2.1问题的求解

算法的流程与3.5一致，其与4.1属于同一问题，不做过多解释。

4.3寻找函数在特定区间的最大值

4.3.1问题的求解

已知这样的一个函数

Step1：绘图

先画出其在区间[0,4]内的图像，即

图表, 折线图

描述已自动生成

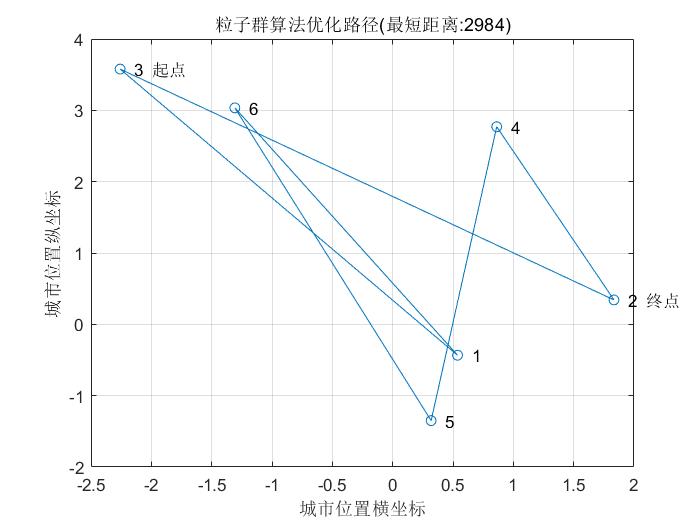
此时需要寻找这个函数在定义域内的最高点（最大值）。这里使用的是粒子群算法解决该问题，

五、实验结果与分析

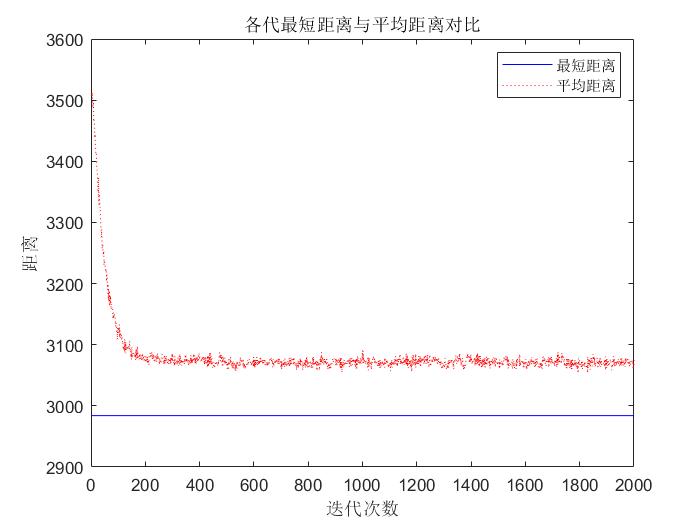
5.1用粒子群算法求解TSP问题

对于city.mat文件中的城市序列，参数在代码注释中写的较为详细，在此不再赘述。

运行代码后得到如下结果：



上图为粒子群算法优化路径示意图，横坐标和纵坐标是随机生成的城市位置。



上图为经过各代最短距离与平均距离的对比图，可以清晰的发现在迭代200次左右趋于平稳

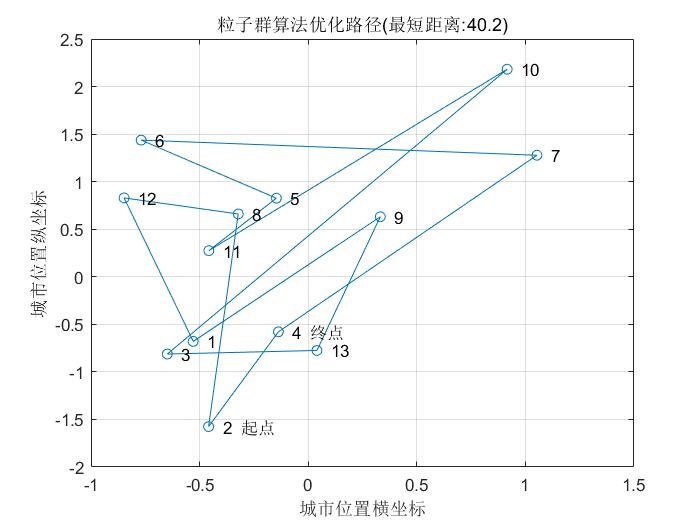
最短距离:2984

最终路径：3,1,6,5,4,2，（3）[3为起点，2为终点]

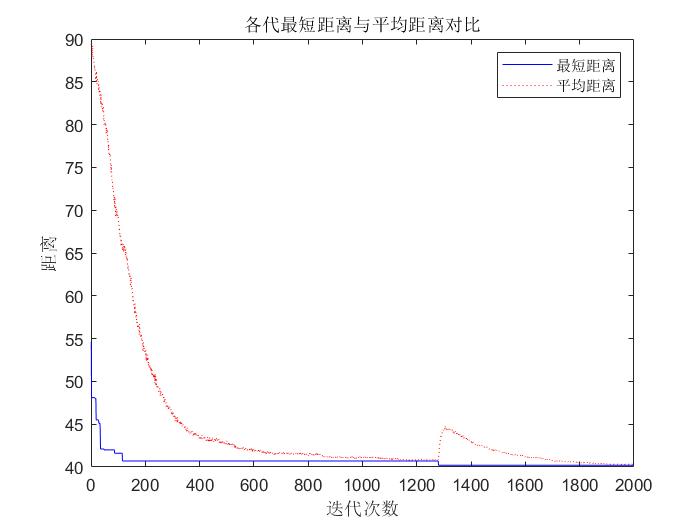
注：1-北京，2-上海，3-天津，4-石家庄，5-太原，6-呼和浩特

5.2用粒子群解决TSP问题（13个城市）

继续沿用上图代码，并加做参数上的修改，



上图为粒子群算法优化路径示意图，横坐标和纵坐标是随机生成的城市位置。

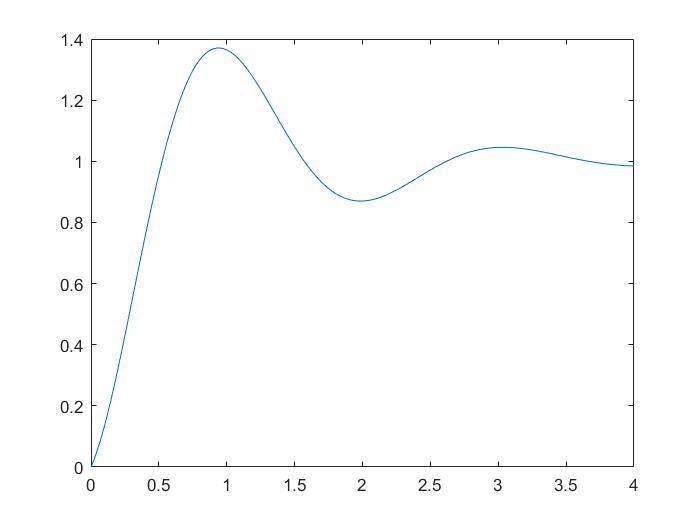


上图为经过各代最短距离与平均距离的对比图，可以清晰的发现在迭代190多次左右趋于平稳

最优路径解2 8 12 1 9 13 3 10 11 5 6 7 4 2，得到的最优路程解的最小路程为40.2

5.3寻求函数在特定区间的最大值

该函数在区间[0,4]之间的图像：



运行6.3的实验代码后，得到如下结果

图表, 折线图

描述已自动生成 图表

描述已自动生成

可以得出当x=1时得最大值1.3642

六、实验代码

liziqunsuanfa\_text\_N=6;

liziqunsuanfa\_text\_citys=randn(liziqunsuanfa\_text\_N,2);

save citys.mat

function change=liziqunsuanfa\_text\_position\_minus\_position(liziqunsuanfa\_text\_best,liziqunsuanfa\_text\_pop)

for liziqunsuanfa\_text\_i=1:size(liziqunsuanfa\_text\_best,1)

for liziqunsuanfa\_text\_j=1:size(liziqunsuanfa\_text\_best,2)

change(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j)=find(liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,:)==liziqunsuanfa\_text\_best(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j));

temp=liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j);

liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j)=liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,change(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j));

liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,change(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j))=temp;

end

end

end

function liziqunsuanfa\_text\_change = liziqunsuanfa\_text\_constant\_times\_velocity(liziqunsuanfa\_text\_constant,liziqunsuanfa\_text\_change)

for liziqunsuanfa\_text\_i=1:size(liziqunsuanfa\_text\_change,1)

for liziqunsuanfa\_text\_j=1:size(liziqunsuanfa\_text\_change,2)

if rand>liziqunsuanfa\_text\_constant

liziqunsuanfa\_text\_change(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j)=0;

end

end

end

end

function liziqunsuanfa\_text\_pop = position\_plus\_velocity(liziqunsuanfa\_text\_pop,liziqunsuanfa\_text\_v)

for liziqunsuanfa\_text\_i=1:size(liziqunsuanfa\_text\_pop,1)

for liziqunsuanfa\_text\_j=1:size(liziqunsuanfa\_text\_pop,2)

if liziqunsuanfa\_text\_v(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j)~=0

liziqunsuanfa\_text\_temp=liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j);

liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j)=liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_v(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j));

liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_v(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j))=liziqunsuanfa\_text\_temp;

end

end

end

end

clear all;

clc;

load citys.mat;

liziqunsuanfa\_text\_D=[ 0,1078,119,263,398,401;

1078,0,963,989,1096,1381;

119,963,0,262,426,504;

263,989,262,0,171,394;

398,1096,426,171,0,341;

401,1381,504,394,341,0];

liziqunsuanfa\_text\_n=6;

liziqunsuanfa\_text\_c1=0.1;

liziqunsuanfa\_text\_c2=0.075;

liziqunsuanfa\_text\_w=1;

liziqunsuanfa\_text\_m=500;

liziqunsuanfa\_text\_pop=zeros(liziqunsuanfa\_text\_m,liziqunsuanfa\_text\_n);

liziqunsuanfa\_text\_v=zeros(liziqunsuanfa\_text\_m,liziqunsuanfa\_text\_n);

liziqunsuanfa\_text\_gen=1;

liziqunsuanfa\_text\_genmax=2000;

liziqunsuanfa\_text\_fitness=zeros(liziqunsuanfa\_text\_m,1);

liziqunsuanfa\_text\_Pbest=zeros(liziqunsuanfa\_text\_m,liziqunsuanfa\_text\_n);

liziqunsuanfa\_text\_Pbest\_fitness=zeros(liziqunsuanfa\_text\_m,1);

liziqunsuanfa\_text\_Gbest=zeros(liziqunsuanfa\_text\_genmax,liziqunsuanfa\_text\_n);

liziqunsuanfa\_text\_Gbest\_fitness=zeros(liziqunsuanfa\_text\_genmax,1);

liziqunsuanfa\_text\_Length\_ave=zeros(liziqunsuanfa\_text\_genmax,1);

liziqunsuanfa\_text\_ws=1;

liziqunsuanfa\_text\_we=0.8;

for liziqunsuanfa\_text\_i=1:liziqunsuanfa\_text\_m

liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,:)=randperm(liziqunsuanfa\_text\_n);

liziqunsuanfa\_text\_v(liziqunsuanfa\_text\_i,:)=randperm(liziqunsuanfa\_text\_n);

end

for liziqunsuanfa\_text\_i=1:liziqunsuanfa\_text\_m

for liziqunsuanfa\_text\_j=1:liziqunsuanfa\_text\_n-1

liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_i)=liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_i) + liziqunsuanfa\_text\_D(liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j),liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j+1));

end

liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_i)=liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_i) + liziqunsuanfa\_text\_D(liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,end),liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,1));

end

liziqunsuanfa\_text\_Pbest\_fitness=liziqunsuanfa\_text\_fitness;

liziqunsuanfa\_text\_Pbest=liziqunsuanfa\_text\_pop;

[liziqunsuanfa\_text\_Gbest\_fitness(1),liziqunsuanfa\_text\_min\_index]=min(liziqunsuanfa\_text\_fitness);

liziqunsuanfa\_text\_Gbest(1,:)=liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_min\_index,:);

liziqunsuanfa\_text\_Length\_ave(1)=mean(liziqunsuanfa\_text\_fitness);

while liziqunsuanfa\_text\_gen<liziqunsuanfa\_text\_genmax

liziqunsuanfa\_text\_gen=liziqunsuanfa\_text\_gen+1;

liziqunsuanfa\_text\_w = liziqunsuanfa\_text\_ws - (liziqunsuanfa\_text\_ws-liziqunsuanfa\_text\_we)\*(liziqunsuanfa\_text\_gen/liziqunsuanfa\_text\_genmax)^2;

liziqunsuanfa\_text\_change1=liziqunsuanfa\_text\_position\_minus\_position(liziqunsuanfa\_text\_Pbest,liziqunsuanfa\_text\_pop);

liziqunsuanfa\_text\_change1=liziqunsuanfa\_text\_constant\_times\_velocity(liziqunsuanfa\_text\_c1,liziqunsuanfa\_text\_change1);

change2=liziqunsuanfa\_text\_position\_minus\_position(repmat(liziqunsuanfa\_text\_Gbest(liziqunsuanfa\_text\_gen-1,:),liziqunsuanfa\_text\_m,1),liziqunsuanfa\_text\_pop);

change2=liziqunsuanfa\_text\_constant\_times\_velocity(liziqunsuanfa\_text\_c2,change2);

liziqunsuanfa\_text\_v=liziqunsuanfa\_text\_constant\_times\_velocity(liziqunsuanfa\_text\_w,liziqunsuanfa\_text\_v);

for liziqunsuanfa\_text\_i=1:liziqunsuanfa\_text\_m

for liziqunsuanfa\_text\_j=1:liziqunsuanfa\_text\_n

if liziqunsuanfa\_text\_change1(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j)~=0

liziqunsuanfa\_text\_v(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j)=liziqunsuanfa\_text\_change1(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j);

end

if change2(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j)~=0

liziqunsuanfa\_text\_v(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j)=change2(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j);

end

end

end

liziqunsuanfa\_text\_pop=position\_plus\_velocity(liziqunsuanfa\_text\_pop,liziqunsuanfa\_text\_v);

liziqunsuanfa\_text\_fitness=zeros(liziqunsuanfa\_text\_m,1);

for liziqunsuanfa\_text\_i=1:liziqunsuanfa\_text\_m

for liziqunsuanfa\_text\_j=1:liziqunsuanfa\_text\_n-1

liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_i)=liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_i) + liziqunsuanfa\_text\_D(liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j),liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,liziqunsuanfa\_text\_j+1));

end

liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_i)=liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_i) + liziqunsuanfa\_text\_D(liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,end),liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,1));

end

for liziqunsuanfa\_text\_i=1:liziqunsuanfa\_text\_m

if liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_i)<liziqunsuanfa\_text\_Pbest\_fitness(liziqunsuanfa\_text\_i)

liziqunsuanfa\_text\_Pbest\_fitness(liziqunsuanfa\_text\_i)=liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_i);

liziqunsuanfa\_text\_Pbest(liziqunsuanfa\_text\_i,:)=liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,:);

end

end

[liziqunsuanfa\_text\_minvalue,liziqunsuanfa\_text\_min\_index]=min(liziqunsuanfa\_text\_fitness);

if liziqunsuanfa\_text\_minvalue<liziqunsuanfa\_text\_Gbest\_fitness(liziqunsuanfa\_text\_gen-1)

liziqunsuanfa\_text\_Gbest\_fitness(liziqunsuanfa\_text\_gen)=liziqunsuanfa\_text\_minvalue;

liziqunsuanfa\_text\_Gbest(liziqunsuanfa\_text\_gen,:)=liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_min\_index,:);

else

liziqunsuanfa\_text\_Gbest\_fitness(liziqunsuanfa\_text\_gen)=liziqunsuanfa\_text\_Gbest\_fitness(liziqunsuanfa\_text\_gen-1);

liziqunsuanfa\_text\_Gbest(liziqunsuanfa\_text\_gen,:)=liziqunsuanfa\_text\_Gbest(liziqunsuanfa\_text\_gen-1,:);

end

liziqunsuanfa\_text\_Length\_ave(liziqunsuanfa\_text\_gen)=mean(liziqunsuanfa\_text\_fitness);

end

[liziqunsuanfa\_text\_Shortest\_Length,liziqunsuanfa\_text\_index] = min(liziqunsuanfa\_text\_Gbest\_fitness);

liziqunsuanfa\_text\_Shortest\_Route = liziqunsuanfa\_text\_Gbest(liziqunsuanfa\_text\_index,:);

disp(['最短距离:' num2str(liziqunsuanfa\_text\_Shortest\_Length)]);

disp(['最短路径:' num2str([liziqunsuanfa\_text\_Shortest\_Route liziqunsuanfa\_text\_Shortest\_Route(1)])]);

figure(1)

plot([liziqunsuanfa\_text\_citys(liziqunsuanfa\_text\_Shortest\_Route,1);liziqunsuanfa\_text\_citys(liziqunsuanfa\_text\_Shortest\_Route(1),1)],...

[liziqunsuanfa\_text\_citys(liziqunsuanfa\_text\_Shortest\_Route,2);liziqunsuanfa\_text\_citys(liziqunsuanfa\_text\_Shortest\_Route(1),2)],'o-');

grid on

for liziqunsuanfa\_text\_i = 1:size(liziqunsuanfa\_text\_citys,1)

text(liziqunsuanfa\_text\_citys(liziqunsuanfa\_text\_i,1),liziqunsuanfa\_text\_citys(liziqunsuanfa\_text\_i,2),[' ' num2str(liziqunsuanfa\_text\_i)]);

end

text(liziqunsuanfa\_text\_citys(liziqunsuanfa\_text\_Shortest\_Route(1),1),liziqunsuanfa\_text\_citys(liziqunsuanfa\_text\_Shortest\_Route(1),2),' 起点');

text(liziqunsuanfa\_text\_citys(liziqunsuanfa\_text\_Shortest\_Route(end),1),liziqunsuanfa\_text\_citys(liziqunsuanfa\_text\_Shortest\_Route(end),2),' 终点');

xlabel('城市位置横坐标')

ylabel('城市位置纵坐标')

title(['粒子群算法优化路径(最短距离:' num2str(liziqunsuanfa\_text\_Shortest\_Length) ')'])

figure(2)

plot(1:liziqunsuanfa\_text\_genmax,liziqunsuanfa\_text\_Gbest\_fitness,'b',1:liziqunsuanfa\_text\_genmax,liziqunsuanfa\_text\_Length\_ave,'r:')

legend('最短距离','平均距离')

xlabel('迭代次数')

ylabel('距离')

title('各代最短距离与平均距离对比')

liziqunsuanfa\_text\_D=[ 0 2.2 18.1 4.8 9.2 8.4 5.2 0.4 9.3 11.3 10.2 0.4 10.4;

2.2 0 17.5 2.8 7.9 6.6 3.3 1.8 8.5 9.8 8.7 2.4 9.6;

18.1 17.5 0 14.7 9.6 10.9 14.2 18.2 9.1 7.6 8.7 18.3 8.0;

4.8 2.8 14.7 0 5.4 3.8 0.5 4.4 6.4 7.0 6.0 5.1 7.4;

9.2 7.9 9.6 5.4 0 2.4 4.9 9.0 1.7 2.3 1.2 9.5 2.2;

8.4 6.6 10.9 3.8 2.4 0 3.3 8.1 4.1 3.3 2.5 8.7 4.6;

5.2 3.3 14.2 0.5 4.9 3.3 0 4.8 6.0 6.6 5.5 5.5 6.9;

0.4 1.8 18.2 4.4 9.0 8.1 4.8 0 9.2 11.1 10.0 0.6 10.3;

9.3 8.5 9.1 6.4 1.7 4.1 6.0 9.2 0 3.3 2.5 9.6 1.1;

11.3 9.8 7.6 7.0 2.3 3.3 6.6 11.1 3.3 0 1.2 11.7 2.8;

10.2 8.7 8.7 6.0 1.2 2.5 5.5 10.0 2.5 1.2 0 10.5 2.5;

0.4 2.4 18.3 5.1 9.5 8.7 5.5 0.6 9.6 11.7 10.5 0 10.7;

10.4 9.6 8.0 7.4 2.2 4.6 6.9 10.3 1.1 2.8 2.5 10.7 0];

clc

clear all

liziqunsuanfa\_text\_x=0:0.01:4;

liziqunsuanfa\_text\_y=1-cos(3\*liziqunsuanfa\_text\_x).\*exp(-liziqunsuanfa\_text\_x);

figure

plot(liziqunsuanfa\_text\_x, liziqunsuanfa\_text\_y)

hold on

liziqunsuanfa\_text\_c1 = 1.49445;

liziqunsuanfa\_text\_c2 = 1.49445;

liziqunsuanfa\_text\_maxgen = 50;

liziqunsuanfa\_text\_sizepop = 10;

liziqunsuanfa\_text\_Vmax = 0.5;

liziqunsuanfa\_text\_Vmin = -0.5;

liziqunsuanfa\_text\_popmax = 2;

liziqunsuanfa\_text\_popmin = 1;

fun=@(x)1-cos(3\*x).\*exp(-x);

for liziqunsuanfa\_text\_i = 1:liziqunsuanfa\_text\_sizepop

liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,:) = (rands(1) + 1) / 2 + 1;

liziqunsuanfa\_text\_V(liziqunsuanfa\_text\_i,:) = 0.5 \* rands(1);

liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_i) = fun(liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_i,:));

end

[liziqunsuanfa\_text\_bestfitness liziqunsuanfa\_text\_bestindex] = max(liziqunsuanfa\_text\_fitness);

liziqunsuanfa\_text\_zbest = liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_bestindex,:);

liziqunsuanfa\_text\_gbest = liziqunsuanfa\_text\_pop;

liziqunsuanfa\_text\_fitnessgbest = liziqunsuanfa\_text\_fitness;

liziqunsuanfa\_text\_fitnesszbest = liziqunsuanfa\_text\_bestfitness;

for liziqunsuanfa\_text\_i = 1:liziqunsuanfa\_text\_maxgen

for liziqunsuanfa\_text\_j = 1:liziqunsuanfa\_text\_sizepop

liziqunsuanfa\_text\_V(liziqunsuanfa\_text\_j,:) = liziqunsuanfa\_text\_V(liziqunsuanfa\_text\_j,:) + liziqunsuanfa\_text\_c1\*rand\*(liziqunsuanfa\_text\_gbest(liziqunsuanfa\_text\_j,:) - liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_j,:)) + liziqunsuanfa\_text\_c2\*rand\*(liziqunsuanfa\_text\_zbest - liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_j,:));

liziqunsuanfa\_text\_V(liziqunsuanfa\_text\_j,find(liziqunsuanfa\_text\_V(liziqunsuanfa\_text\_j,:)>liziqunsuanfa\_text\_Vmax)) = liziqunsuanfa\_text\_Vmax;

liziqunsuanfa\_text\_V(liziqunsuanfa\_text\_j,find(liziqunsuanfa\_text\_V(liziqunsuanfa\_text\_j,:)<liziqunsuanfa\_text\_Vmin)) = liziqunsuanfa\_text\_Vmin;

liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_j,:) = liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_j,:) + liziqunsuanfa\_text\_V(liziqunsuanfa\_text\_j,:);

liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_j,find(liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_j,:)>liziqunsuanfa\_text\_popmax)) = liziqunsuanfa\_text\_popmax;

liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_j,find(liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_j,:)<liziqunsuanfa\_text\_popmin)) = liziqunsuanfa\_text\_popmin;

liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_j) = fun(liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_j,:));

end

for liziqunsuanfa\_text\_j = 1:liziqunsuanfa\_text\_sizepop

if liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_j) > liziqunsuanfa\_text\_fitnessgbest(liziqunsuanfa\_text\_j)

liziqunsuanfa\_text\_gbest(liziqunsuanfa\_text\_j,:) = liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_j,:);

liziqunsuanfa\_text\_fitnessgbest(liziqunsuanfa\_text\_j) = liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_j);

end

if liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_j) > liziqunsuanfa\_text\_fitnesszbest

liziqunsuanfa\_text\_zbest = liziqunsuanfa\_text\_pop(liziqunsuanfa\_text\_j,:);

liziqunsuanfa\_text\_fitnesszbest = liziqunsuanfa\_text\_fitness(liziqunsuanfa\_text\_j);

end

end

yy(liziqunsuanfa\_text\_i) = liziqunsuanfa\_text\_fitnesszbest;

end

[liziqunsuanfa\_text\_fitnesszbest liziqunsuanfa\_text\_zbest]

plot(liziqunsuanfa\_text\_zbest, liziqunsuanfa\_text\_fitnesszbest,'r\*')

figure

plot(yy)

title('最优个体适应度','fontsize',12);

xlabel('进化代数','fontsize',12);ylabel('适应度','fontsize',12);

liziqunsuanfa\_text\_x=0:0.0001:4;

liziqunsuanfa\_text\_y=1-cos(3.\*liziqunsuanfa\_text\_x).\*exp(-liziqunsuanfa\_text\_x);

[liziqunsuanfa\_text\_ymax,liziqunsuanfa\_text\_xmax]=max(liziqunsuanfa\_text\_y)