实验四 MATLAB数据交换技术

1. 实验目的

**1.** 熟悉文件和数据接口操作的技术和方法

**2.** 掌握MATLAB与Word的相互操作

**3.** 学习MATLAB与Excel的相互操作

**4.** 熟悉自然启发式算法(例如蚁群算法)在MATLAB上的使用

1. 实验平台

Windows7、MATLAB 2022b、Office2010软件

三、演练

**1.**MATLAB与word的交互

在 MATLAB 中与 Word 进行交互通常涉及使用 COM（Component Object Model）对象。这里是一些常用的 MATLAB 函数以及它们的使用方法：

1. **actxserver**：
   * 用法：h = actxserver('ProgramID')
   * 功能：创建一个 COM 服务器对象，可以用于与指定的程序进行交互。在这种情况下，ProgramID 是 Word 应用程序的标识符。
   * 示例：

wordApp = actxserver('Word.Application');

1. **invoke**：
   * 用法：invoke(object, 'MethodName', arg1, arg2, ...) 或 object.MethodName(arg1, arg2, ...)
   * 功能：调用指定 COM 对象的方法。
   * 示例：

wordApp = actxserver('Word.Application');

doc = invoke(wordApp.Documents, 'Add');

1. **get** 和 **set**：
   * 用法：value = get(object, 'PropertyName') 和 set(object, 'PropertyName', value)
   * 功能：用于获取或设置 COM 对象的属性值。
   * 示例：

visible = get(wordApp, 'Visible'); % 获取 Word 应用程序的可见性

set(wordApp, 'Visible', true); % 设置 Word 应用程序为可见

1. **Quit**：
   * 用法：invoke(object, 'Quit')
   * 功能：关闭指定的 COM 对象所代表的应用程序。
   * 示例：

invoke(wordApp, 'Quit');

* 1. 使用MATLAB创立一个word文档，并在Word文档中添加一段文本。

在MATLAB 命令行窗口下键入“edit”或选择“新建脚本”，进入MATLAB 的编辑器，编辑M文件“exampleword.m”。

% 创建 Word 应用程序对象

wordApp = actxserver('Word.Application');

% 显示 Word 应用程序窗口（可选）

wordApp.Visible = true;

% 创建新文档

doc = wordApp.Documents.Add;

% 在文档中添加文本

wordApp.Selection.TypeText('这是来自 MATLAB 的文本。');

% 获取当前工作目录

currentFolder = pwd;

% 拼接保存文件的路径

savePath = fullfile(currentFolder, 'MatlabWordDemo.docx');

% 保存文档

doc.SaveAs2(savePath);

% 关闭 Word 应用程序

wordApp.Quit;

点击“保存”按钮，保存文件名为“exampleword.m”。选择“编辑器”菜单的“运行按选项或直接在MATLAB的命令行窗口下键入文件名exampleword，在MATLAB的命令行窗口下查看运行的结果。

点击当前文件夹下的“MatlabWordDemo.docx”文件,运行结果如图一所示：



图一.MATLAB与Word交互示例一

**1.2** 将 MATLAB 中的数据分析结果自动导出到 Word 文档中，并且包括图表、表格和文本说明。

在MATLAB 命令行窗口下键入“edit”或选择“新建脚本”，进入MATLAB 的编辑器，编辑M文件“exampleword2.m”。

% 连接到 Word 应用程序

wordApp = actxserver('Word.Application');

% 创建一个新的 Word 文档

wordDoc = wordApp.Documents.Add();

% 在 Word 文档中添加标题

wordDoc.Paragraphs.Add();

wordApp.Selection.EndKey(6); % 移动光标到文档末尾

wordApp.Selection.TypeText('MATLAB 数据分析报告');

wordApp.Selection.Font.Bold = 1;

wordApp.Selection.Font.Size = 16;

% 在 MATLAB 中生成示例数据

x = linspace(0, 10, 100);

y = sin(x);

% 在 MATLAB 中绘制图表

figure;

plot(x, y);

xlabel('X');

ylabel('Y');

title('Sine Wave');

% 将图表导出为 PNG 图像

print('-dpng', 'sine\_wave.png');

% 将图像插入到 Word 文档中

wordApp.Selection.EndKey(6); % 移动光标到文档末尾

wordDoc.InlineShapes.AddPicture(fullfile(pwd, 'sine\_wave.png'));

% 创建一个表格

data = rand(5, 3);

tableData = num2cell(data);

% 在 Word 文档中插入表格

wordApp.Selection.TypeParagraph; % 插入新段落

wordApp.Selection.TypeText('示例数据表格:');

wordApp.Selection.Font.Bold = 1;

wordApp.Selection.Font.Size = 12;

wordApp.Selection.TypeParagraph; % 插入新段落

wordDoc.Tables.Add(wordApp.Selection.Range, 5, 3); % 添加表格

for i = 1:5

for j = 1:3

wordDoc.Tables.Item(1).Cell(i,j).Range.Text = num2str(data(i,j));

end

end

% 添加文本说明

wordApp.Selection.TypeParagraph; % 插入新段落

wordApp.Selection.TypeText('这是一个示例的数据分析报告。');

% 保存 Word 文档

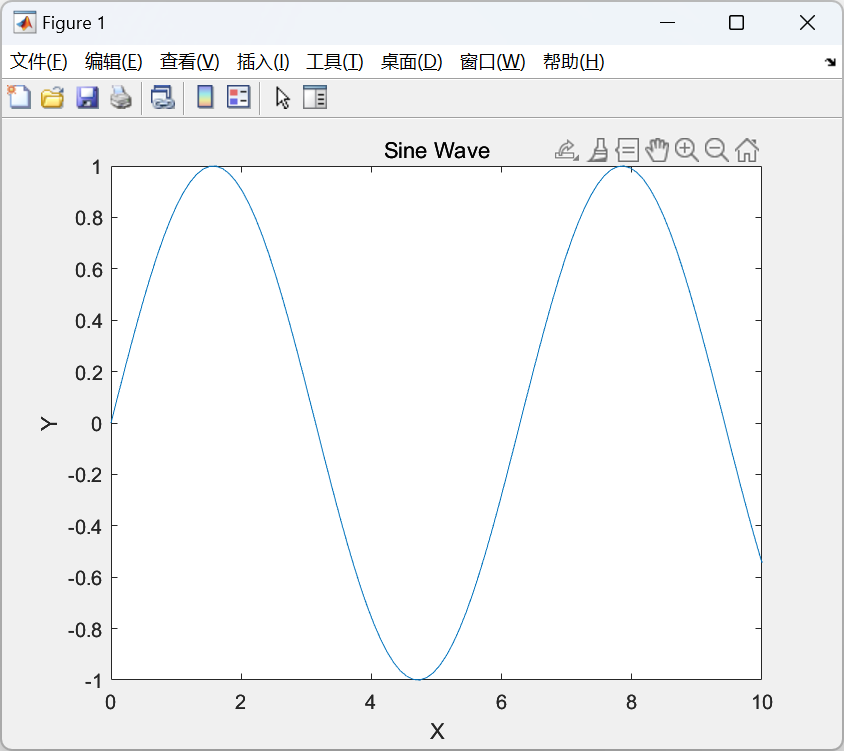
wordDoc.SaveAs2(fullfile(pwd, 'MATLAB\_Report.docx'));

% 关闭 Word 应用程序

wordApp.Quit();

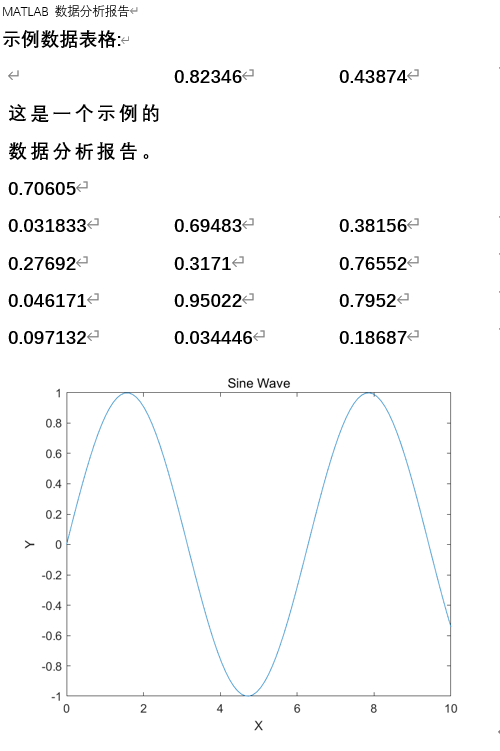
点击“保存”按钮，保存文件名为“exampleword2.m”。选择“编辑器”菜单的“运行按选项或直接在MATLAB的命令行窗口下键入文件名exampleword2，在MATLAB的命令行窗口下查看运行的结果。

运行结果如图二所示：



图二.MATLAB与Word交互示例二(1)

双击MATLAB当前文件夹下的MATLAB\_Report.docx，结果如图三所示



图三.MATLAB与Word交互示例二(2)

**2.** MATLAB与Excel数据交互

MATLAB与Excel数据交互是用**xlsread**函数和**xlswrite**函数来实现的，以下是其语法结构

**xlsread**

data = xlsread(filename)

data = xlsread(filename, sheet)

data = xlsread(filename, range)

data = xlsread(filename, sheet, range)

[num, txt, raw] = xlsread(\_\_\_)

* filename: 要读取数据的 Excel 文件的文件名或者路径。
* sheet (可选): 要读取数据的工作表名称或索引，默认为第一个工作表。
* range (可选): 要读取数据的单元格范围，例如 'A1:B10'，默认为整个工作表。
* data: 读取的数值数据。
* num: 仅读取数值时返回的数值数据。
* txt: 仅读取文本时返回的文本数据。
* raw: 返回原始数据，包括数值和文本。

**xlswrite**

xlswrite(filename, data)

xlswrite(filename, data, sheet)

xlswrite(filename, data, range)

xlswrite(filename, data, sheet, range)

* filename: 要写入数据的 Excel 文件的文件名或者路径。
* data: 要写入的数据，可以是矩阵、单元数组或一维数组。
* sheet (可选): 要写入数据的工作表名称或索引，默认为第一个工作表。
* range (可选): 要写入数据的单元格范围，例如 'A1:B10'，默认为整个工作表。

**2.1** 创建一个 5x5 的魔方矩阵作为示例数据。然后，我们使用 xlswrite 将数据写入名为 example.xlsx 的 Excel 文件的第一个工作表的 A1 单元格。接着，我们使用 xlsread 从同一个文件中读取数据，并将其存储在 read\_data 变量中。

在MATLAB 命令行窗口下键入“edit”或选择“新建脚本”，进入MATLAB 的编辑器，编辑M文件“excinteraction.m”。

% 关闭所有图形窗口并清除内存变量

close all;

clear;

clc;

% 创建一个示例数据

data = magic(5); % 一个 5x5 的魔方矩阵

% 定义要写入的文件名

filename = 'example.xlsx';

% 使用 xlswrite 将数据写入 Excel 文件

xlswrite(filename, data, 'Sheet1', 'A1');

% 提示写入完成

disp('Data has been written to example.xlsx.');

% 读取刚刚写入的 Excel 文件中的数据

read\_data = xlsread(filename);

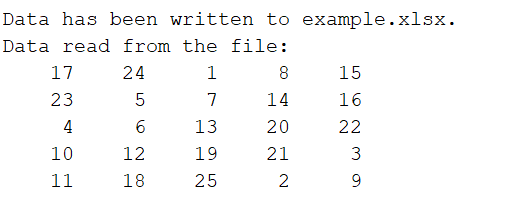
% 显示读取的数据

disp('Data read from the file:');

disp(read\_data);

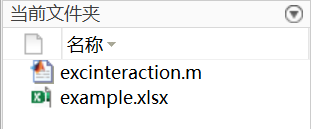
点击“保存”按钮，保存文件名为“excinteraction.m”。选择“编辑器”菜单的“运行按选项或直接在MATLAB的命令行窗口下键入文件名excinteraction，在MATLAB的命令行窗口下查看运行的结果。

结果如图四所示：



图四.魔方矩阵结果

此时，当前文件夹下也会出现如图五所示的example.xlsx。



图五.Excel文件图

**2.2** 对2.1中产生的魔方矩阵，求其相应的逆矩阵，将其写入example.xlsx的Sheet2中，然后求魔方矩阵各元素的倒数矩阵，将其写入example.xlsx的Sheet3中。

在命令控制行内键入

% 对魔方矩阵求逆

inv\_data = inv(data);

% 使用 xlswrite 将逆矩阵写入 Excel 文件到Sheet2中，从A1开始写入，而不清除现有内容

xlswrite(filename, inv\_data, 'Sheet2', 'A1');

% 提示写入完成

disp('Inverse of the magic square has been written to Sheet2.');

% 计算魔方矩阵各元素的倒数

reciprocal\_data = 1 ./ data;

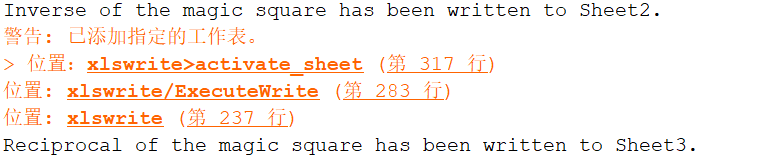
% 使用 xlswrite 将倒数矩阵写入 Excel 文件到Sheet3中，从A1开始写入，而不清除现有内容

xlswrite(filename, reciprocal\_data, 'Sheet3', 'A1');

% 提示写入完成

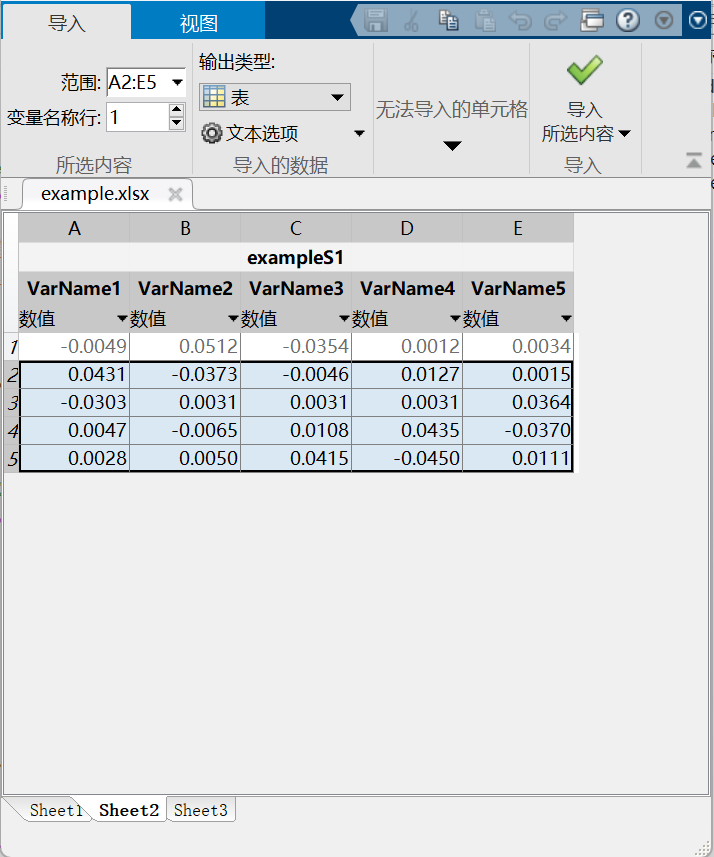
disp('Reciprocal of the magic square has been written to Sheet3.');

运行结果如下

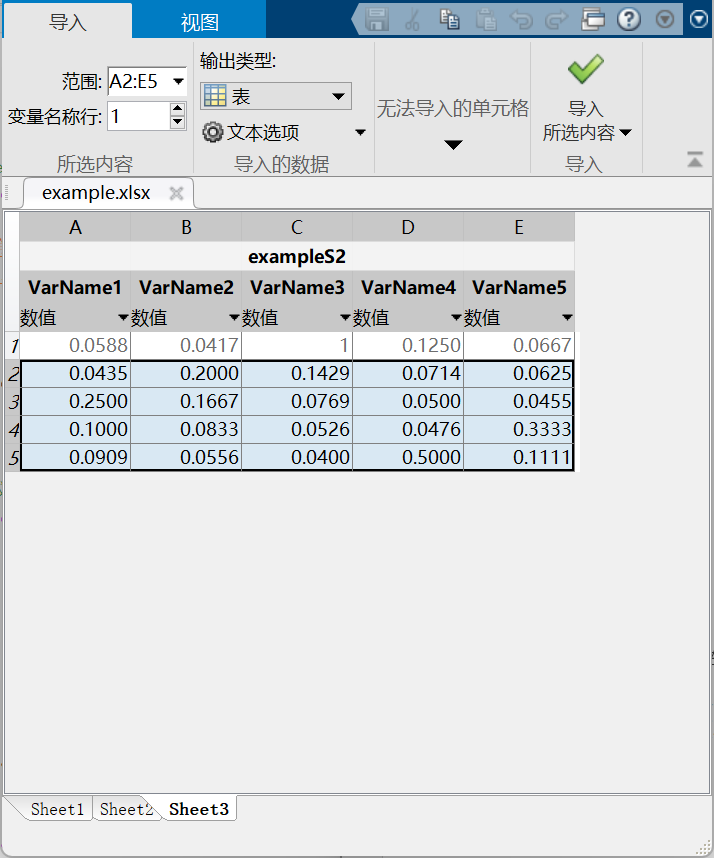


图六.运行结果图

双击当前文件夹下的example.xlsx，，选择Sheet2和Sheet3，可以看到结果如图



图七.Sheet2数据图



图八.Sheet。数据图

**2.3** 将Sheet1和Sheet3中的矩阵做矩阵加法，并将结果放入到Sheet4中。

在命令控制行内键入

% 读取 Sheet1 中的数据

data\_sheet1 = xlsread(filename, 'Sheet1');

% 读取 Sheet3 中的数据

data\_sheet3 = xlsread(filename, 'Sheet3');

% 执行矩阵加法运算

result = data\_sheet1 + data\_sheet3;

% 将结果写入 Sheet4

xlswrite(filename, result, 'Sheet4', 'A1');

% 提示写入完成

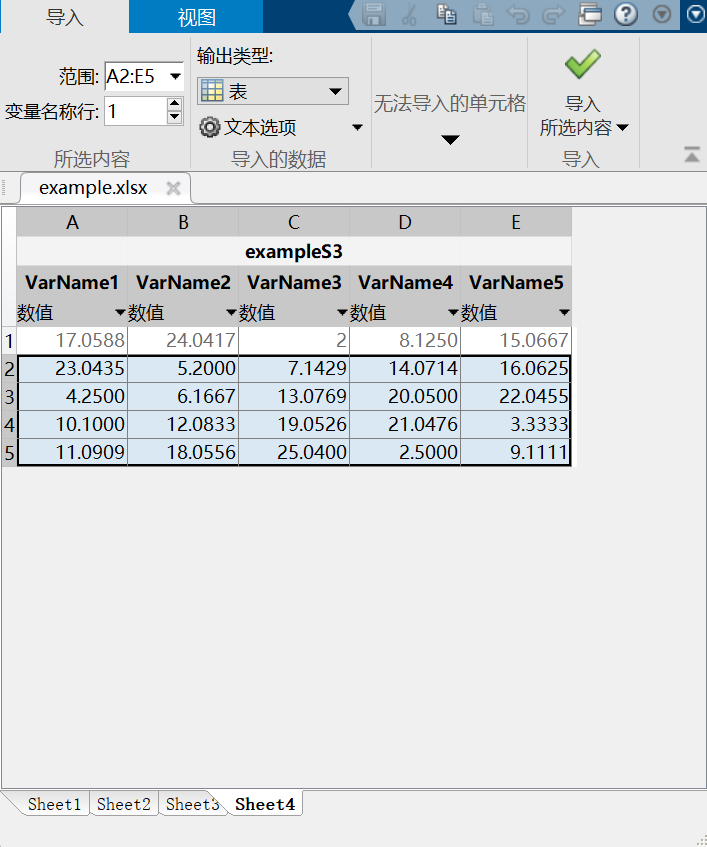
disp('Matrix addition result has been written to Sheet4.');

% 显示结果

disp('Matrix addition result:');

disp(result);

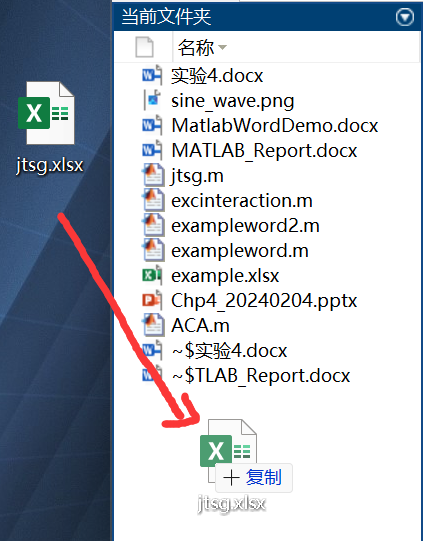
运行结果如下



图九.Sheet4数据图

**2.4** 某市2010～2016年三个路口24小时交通如jtsg.xlsx文件所示，试用MATLAB编程读取Excel文件JTSG.XLSX中的数据并汇出折线图。

步骤一、用鼠标将jtsg.xlsx文件拖动到MATLAB的当前文件夹下，如图十所示：



图十.操作步骤

步骤二、在MATLAB 命令行窗口下键入“edit”或选择“新建脚本”，进入MATLAB 的编辑器，编辑M文件“jtsg.m”。

% 从 Excel 文件中读取数据

[num\_data, text\_data, raw\_data] = xlsread('jtsg.xlsx');

% 提取时间和交通事故次数数据

time = text\_data(3:end, 1); % 从第三行开始提取时间数据

guilin\_road = num\_data(:, 1);

hongqi\_street = num\_data(:, 2);

yugu\_road = num\_data(:, 3);

% 绘制折线图

figure;

plot(1:numel(time), guilin\_road, '-o', 'DisplayName', '桂林路');

hold on;

plot(1:numel(time), hongqi\_street, '-s', 'DisplayName', '红旗街');

plot(1:numel(time), yugu\_road, '-d', 'DisplayName', '玉古路');

hold off;

% 设置横坐标标签

xticks(1:numel(time));

xticklabels(time);

% 添加标题和标签

title('24小时交通事故次数登记表');

xlabel('时间');

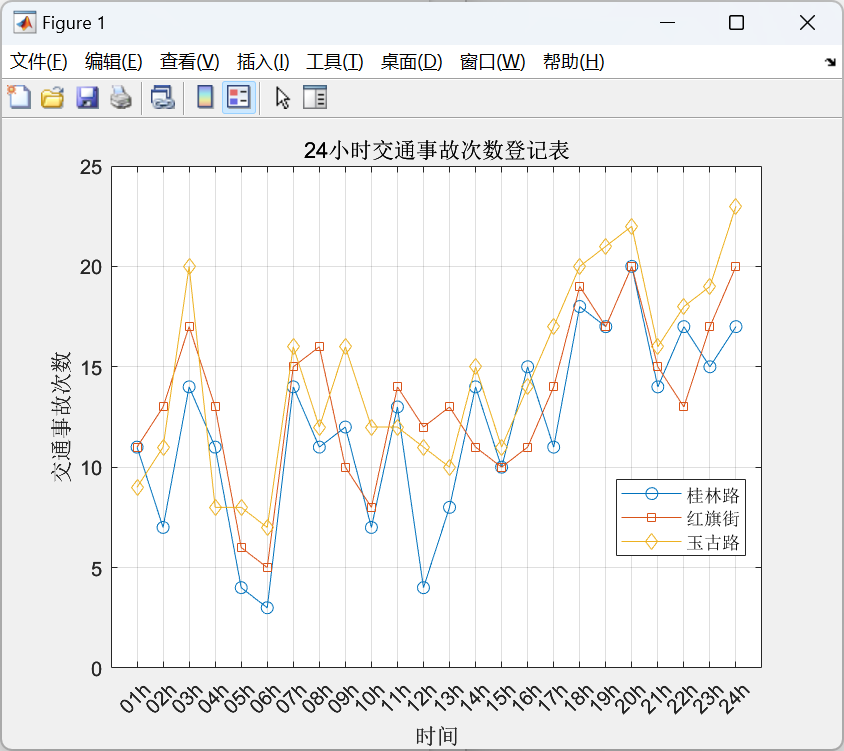
ylabel('交通事故次数');

legend('Location', 'best');

grid on;

步骤三、点击“保存”按钮，保存文件名为“jtsg.m”。选择“编辑器”菜单的“运行按选项或直接在MATLAB的命令行窗口下键入文件名jtsg，在MATLAB的命令行窗口下查看运行的结果。

运行结果如图十一所示：



图十一.运行结果图

**3.**自然启发式算法(蚁群算法)

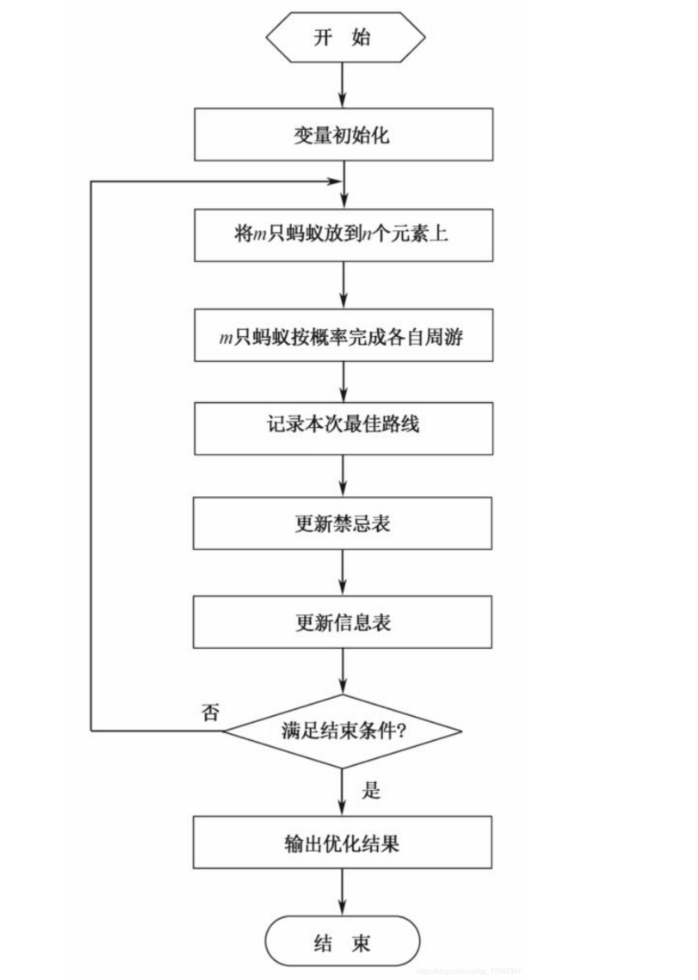
蚁群算法（Ant Colony Optimization, ACO）是一种启发式算法，灵感来自于观察蚂蚁在寻找食物时的行为。

在蚁群算法中，一群虚拟蚂蚁通过在解空间中随机移动和释放信息素，来解决优化问题。其基本思想是模拟蚂蚁在实际环境中的行为，并利用信息素的正反馈机制实现全局搜索和局部搜索的平衡。

蚁群算法的主要步骤包括：

1. **初始化**：初始化虚拟蚂蚁的位置，并初始化信息素的分布。
2. **路径选择**：蚂蚁在解空间中根据一定的策略（通常是概率策略）选择下一步要前往的位置。
3. **更新信息素**：当所有蚂蚁完成路径选择后，根据它们的路径质量更新信息素的分布。
4. **信息素挥发**：为了防止信息素浓度过高导致早熟收敛，需要适时挥发信息素。
5. **迭代**：重复执行步骤2至步骤4，直到达到迭代终止条件（例如达到最大迭代次数或达到问题特定的停止条件）。

蚁群算法主要应用于求解组合优化问题，如旅行商问题（TSP）、调度问题、路径规划等。其优点包括对于高维、非线性、非凸优化问题的较好适用性，以及对局部搜索和全局搜索的平衡能力。



图十二.蚁群算法流程图

各参数含义

1. **m**：这里代表蚂蚁的数量。
2. **Alpha**：Alpha是信息素在路径选择中的重要程度参数，它影响着蚂蚁选择下一步路径时对信息素浓度的重视程度。
3. **Beta**：Beta是启发函数在路径选择中的重要程度参数，它影响着蚂蚁选择下一步路径时对启发信息的重视程度。
4. **Rho**：Rho是信息素挥发率，它控制着信息素在更新过程中的挥发速度。例如Rho被设置为0.1，表示每次更新后信息素的挥发率为10%，即信息素在更新后只有原来的90%。
5. **G**：G表示算法的迭代次数或者迭代终止条件。
6. **Q = 100**：Q是信息素增量参数，用于确定每只蚂蚁在路径上释放的信息素量。在例如Q被设置为100，表示每只蚂蚁在路径上释放100的信息素。

**3.1**旅行商问题（TSP问题）。假设有一个旅行商人要拜访全国 31个省会城市，他需要选择所要走的路径，路径的限制是每个城市只能拜访一次，而且最后要回到原来出发的城市。路径的选择要求是： 所选路径的路程为所有路径之中的最小值。 全国31个省会城市的坐标为 [1304 2312；3639 1315；4177 2244； 3712 1399；3488 1535；3326 1556；3238 1229；4196 1004；4312 790；4386 570；3007 1970；2562 1756；2788 1491；2381 1676；1332 695；3715 1678；3918 2179；4061 2370；3780 2212；3676 2578；4029 2838；4263 2931；3429 1908；3507 2367；3394 2643；3439 3201； 2935 3240；3140 3550；2545 2357；2778 2826；2370 2975]。

在MATLAB 命令行窗口下键入“edit”或选择“新建脚本”，进入MATLAB 的编辑器，编辑M文件“ACA.m”。

%%%%%%%%%%%%%%%%%%%%%%%%%%Alpha = 1;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

m = 50;

Alpha = 1;

Beta = 5;

Rho = 0.1;

G = 200;

Q = 100;

C=[1304 2312;3639 1315;4177 2244;3712 1399;3488 1535;3326 1556;3238 1229;4196 1004;4312 790;4386 570;3007 1970;2562 1756;2788 1491;2381 1676;1332 695;3715 1678;3918 2179;4061 2370;3780 2212;3676 2578;4029 2838;4263 2931;3429 1908;3507 2367;3394 2643;3439 3201;2935 3240;3140 3550;2545 2357;2778 2826;2370 2975];

%%%%%%%%%%%%%%%%%%%%%%%第一步：变量初始化%%%%%%%%%%%%%%%%%%%%%%%%%%

n = size(C,1);

D = zeros(n,n);

for i = 1:n

for j = 1:n

if i ~= j

D(i,j) = ((C(i,1)-C(j,1))^2+(C(i,2)-C(j,2))^2)^0.5;

else

D(i,j) = eps;

end

D(j,i) = D(i,j);

end

end

Eta = 1./D; %Eta为启发因子，这里设为距离的倒数

Tau = ones(n,n); %Tau为信息素矩阵

Tabu = zeros(m,n); %存储并记录路径的生成

NC = 1; %迭代计数器

R\_best = zeros(G,n); %各代最佳路线

L\_best = inf.\*ones(G,1); %各代最佳路线的长度

figure(1); %优化解

while NC <= G

%%%%%%%%%%%%%%%%%%%%第二步：将m只蚂蚁放到n个城市上%%%%%%%%%%%%%%%%%%%

Randpos = [];

for i = 1:(ceil(m/n))

Randpos = [Randpos,randperm(n)];

end

Tabu (:,1) = (Randpos(1,1:m))';

%%%%第三步：m只蚂蚁按概率选择下一座城市，完成各自的周游%%%%%%%%%%%%%%%%

for j = 2:n

for i = 1:m

visited = Tabu(i,1:(j-1)); %已访问的城市

J = zeros(1,(n-j+1)); %待访问的城市

P=J; %待访问城市的选择概率分布

Jc=1;

for k = 1:n

if length(find(visited==k))==0

J(Jc) = k;

Jc = Jc+1;

end

end

%%%%%%%%%%%%%计算待选城市的概率分布%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k = 1:length(J)

P(k) = (Tau(visited(end),J(k))^Alpha)...

\*(Eta(visited(end),J(k))^Beta);

end

P = P/(sum(P));

%%%%%%%%%%%%%按概率原则选取下一个城市%%%%%%%%%%%%%%%%%%%%%%%%%%

Pcum = cumsum(P);

Select = find(Pcum >= rand);

to\_visit = J(Select(1));

Tabu(i,j) = to\_visit;

end

end

if NC >= 2

Tabu(1,:) = R\_best(NC-1,:);

end

%%%%%%%%%%%%%%%%%第四步：记录本次迭代最佳路线%%%%%%%%%%%%%%%%%%%%%%%%

L = zeros(m,1);

for i = 1:m

R = Tabu(i,:);

for j = 1:n-1

L(i) = L(i)+D(R(j),R(j+1));

end

L(i) = L(i)+D(R(1),R(n));

end

L\_best(NC) = min(L);

pos = find(L==L\_best(NC));

R\_best(NC,:) = Tabu(pos(1),:);

%%%%%%%%%%%%%%%%%%第五步：更新信息素%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Delta\_Tau = zeros(n,n);

for i = 1:m

for j = 1:(n-1)

Delta\_Tau(Tabu(i,j),Tabu(i,j+1)) = ...

Delta\_Tau(Tabu(i,j),Tabu(i,j+1))+Q/L(i);

end

Delta\_Tau(Tabu(i,n),Tabu(i,1)) = ...

Delta\_Tau(Tabu(i,n),Tabu(i,1))+Q/L(i);

end

Tau = (1-Rho).\*Tau+Delta\_Tau;

%%%%%%%%%%%%%%%%%%%%%%%%%第六步：禁忌表清零%%%%%%%%%%%%%%%%%%%%%%%%%%

Tabu = zeros(m,n);

%%%%%%%%%%%%%%%%%%%%%%%%%历代最优路线%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i = 1:n-1

plot([C(R\_best(NC,i),1),C(R\_best(NC,i+1),1)],...

[C(R\_best(NC,i),2),C(R\_best(NC,i+1),2)],'bo-');

hold on;

end

plot([C(R\_best(NC,n),1),C(R\_best(NC,1),1)],...

[C(R\_best(NC,n),2),C(R\_best(NC,1),2)],'ro-');

title(['优化最短路径：',num2str(L\_best(NC))]);

hold off;

pause(0.005);

NC=NC+1;

end

%%%%%%%%%%%%%%%%%%%%%%%%第七步：输出结果%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Pos = find(L\_best==min(L\_best));

Shortest\_Route = R\_best(Pos(1),:);

Shortest\_Length = L\_best(Pos(1));

figure(2);

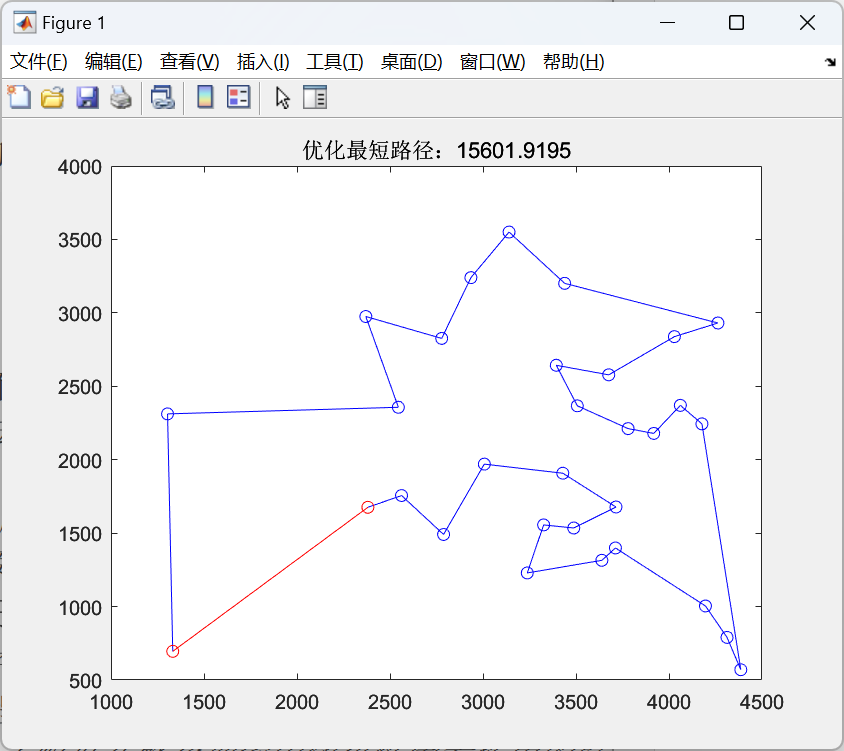
plot(L\_best)

xlabel('迭代次数')

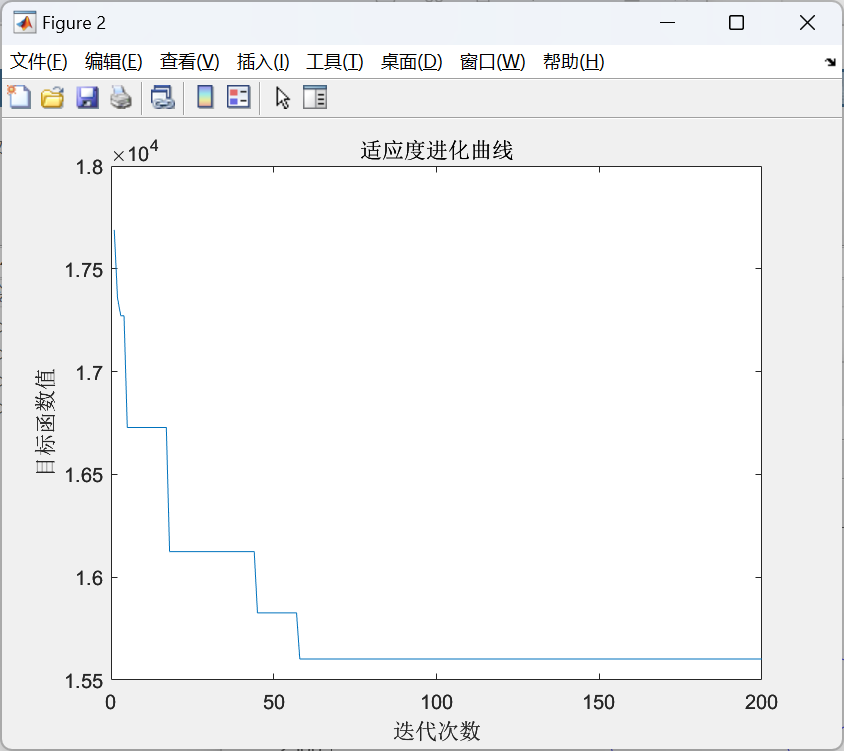
ylabel('目标函数值')

title('适应度进化曲线')

点击“保存”按钮，保存文件名为“ACA.m”。选择“编辑器”菜单的“运行按选项或直接在MATLAB的命令行窗口下键入文件名ACA，在MATLAB的命令行窗口下查看运行的结果。



图十三.蚁群算法解决TSP问题路径图



图十四.适应度进化曲线图