作业 6-2：ACO 实验

针对附件中 TSP 标准问题 kroA100，设计一种 ACO 算法进行求解。

采用基本蚁群算法进行求解

MATLAB代码如下：

%{

TSP文件信息：

NAME : kroA100

TYPE : TSP

COMMENT : 100 - city problem A(Krolak / Felts / Nelson)

DIMENSION : 100

EDGE\_WEIGHT\_TYPE : EUC\_2D

BEST\_KNOWN : 21282

NODE\_COORD\_SECTION

%}

%% I.读取TSP文件信息

fid = fopen('kroA100.tsp', 'rt'); %以文本只读方式打开文件

if (fid <= 0)

disp('文件打开失败！')

return;

end

location = []; A = [1 2];

tline = fgetl(fid); % 读取文件第一行

while ischar(tline)

if (strcmp(tline, 'NODE\_COORD\_SECTION'))

while ~isempty(A)

A = fscanf(fid, '%f', [3, 1]); % 读取节点坐标数据，每次读取一行之后，文件指针会自动指到下一行

if isempty(A)

break;

end

location = [location; A(2:3)']% 将节点坐标存到location中

end

end

tline = fgetl(fid);

if strcmp(tline, 'EOF') % 判断文件是否结束

break;

end

end

citys = location; % 将城市坐标赋给citys

%% II.计算城市间相互距离

n = size(citys, 1);

D = zeros(n, n);

for i = 1:n

for j = 1 : n

if i ~= j

D(i, j) = sqrt(sum((citys(i, :) - citys(j, :)). ^ 2));

else

D(i, j) = 1e-4; % 给i和i的距离赋予一个较小的值.

end

end

end

%% III.初始化参数

m = 50; % 蚂蚁数量

alpha = 1; % 信息素重要程度因子

beta = 5; % 启发函数重要程度因子

rho = 0.1; % 信息素挥发因子

Q = 1; % 常系数

Eta = 1. / D; % 启发函数

Tau = ones(n, n); % 信息素矩阵

Table = zeros(m, n); % 路径记录表

iter = 1; % 迭代次数初值

iter\_max = 200; % 最大迭代次数

Route\_best = zeros(iter\_max, n); % 各代最佳路径

Length\_best = zeros(iter\_max, 1); % 各代最佳路径的长度

Length\_ave = zeros(iter\_max, 1); % 各代路径的平均长度

%% IV.迭代寻找最佳路径

while iter <= iter\_max

% 随机产生各个蚂蚁的起点城市

start = zeros(m, 1);

for i = 1:m

temp = randperm(n);

start(i) = temp(1);

end

Table(:, 1) = start;

citys\_index = 1:n;

% 逐个蚂蚁路径选择

for i = 1:m

% 逐个城市路径选择

for j = 2 : n

tabu = Table(i, 1:(j - 1)); % 已访问的城市集合(禁忌表)

allow\_index = ~ismember(citys\_index, tabu);

allow = citys\_index(allow\_index); % 待访问的城市集合

P = allow;

% 计算城市间转移概率

for k = 1:length(allow)

P(k) = Tau(tabu(end), allow(k)) ^ alpha \* Eta(tabu(end), allow(k)) ^ beta;

end

P = P / sum(P);

% 轮盘赌法选择下一个访问城市

Pc = cumsum(P);

target\_index = find(Pc >= rand);

target = allow(target\_index(1));

Table(i, j) = target;

end

end

% 计算各个蚂蚁的路径距离

Length = zeros(m, 1);

for i = 1:m

Route = Table(i, :);

for j = 1:(n - 1)

Length(i) = Length(i) + D(Route(j), Route(j + 1));

end

Length(i) = Length(i) + D(Route(n), Route(1));

end

% 计算最短路径距离及平均距离

if iter == 1

[min\_Length, min\_index] = min(Length);

Length\_best(iter) = min\_Length;

Length\_ave(iter) = mean(Length);

Route\_best(iter, :) = Table(min\_index, :);

else

[min\_Length, min\_index] = min(Length);

Length\_best(iter) = min(Length\_best(iter - 1), min\_Length);

Length\_ave(iter) = mean(Length);

if Length\_best(iter) == min\_Length

Route\_best(iter, :) = Table(min\_index, :);

else

Route\_best(iter, :) = Route\_best((iter - 1), :);

end

end

% 更新信息素

Delta\_Tau = zeros(n, n);

% 逐个蚂蚁计算

for i = 1:m

% 逐个城市计算

for j = 1 : (n - 1)

Delta\_Tau(Table(i, j), Table(i, j + 1)) = Delta\_Tau(Table(i, j), Table(i, j + 1)) + Q / Length(i);

end

Delta\_Tau(Table(i, n), Table(i, 1)) = Delta\_Tau(Table(i, n), Table(i, 1)) + Q / Length(i);

end

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

% 迭代次数加1，清空路径记录表

iter = iter + 1;

Table = zeros(m, n);

end

%% V.结果显示

[Shortest\_Length, index] = min(Length\_best);

Shortest\_Route = Route\_best(index, :);

disp(['最短距离:' num2str(Shortest\_Length)]);

disp(['最短路径:' num2str([Shortest\_Route Shortest\_Route(1)])]);

%% VI.绘图

figure(1)

plot([citys(Shortest\_Route, 1); citys(Shortest\_Route(1), 1)], ...

[citys(Shortest\_Route, 2); citys(Shortest\_Route(1), 2)], 'o-');

grid on

for i = 1:size(citys, 1)

text(citys(i, 1), citys(i, 2), [' ' num2str(i)]);

end

text(citys(Shortest\_Route(1), 1), citys(Shortest\_Route(1), 2), ' 起点');

text(citys(Shortest\_Route(end), 1), citys(Shortest\_Route(end), 2), ' 终点');

xlabel('城市位置横坐标')

ylabel('城市位置纵坐标')

title(['蚁群算法优化路径(最短距离:' num2str(Shortest\_Length) ')'])

figure(2)

plot(1:iter\_max, Length\_best, 'b', 1 : iter\_max, 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')

实验参数如下：

m = 50; % 蚂蚁数量

alpha = 1; % 信息素重要程度因子

beta = 5; % 启发函数重要程度因子

rho = 0.1; % 信息素挥发因子

Q = 1; % 常系数

Eta = 1. / D; % 启发函数

Tau = ones(n, n); % 信息素矩阵

Table = zeros(m, n); % 路径记录表

iter = 1; % 迭代次数初值

iter\_max = 200; % 最大迭代次数

Route\_best = zeros(iter\_max, n); % 各代最佳路径

Length\_best = zeros(iter\_max, 1); % 各代最佳路径的长度

Length\_ave = zeros(iter\_max, 1); % 各代路径的平均长度

信息素更新策略：蚁量模型

初始位置：随机确定

运行结果：

最短距离稳定在22600左右

蚁群算法优化路径图和最短距离与平均距离对比图如下：

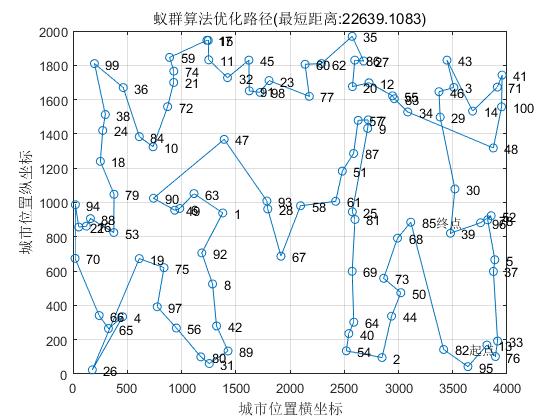


图1 优化路径图

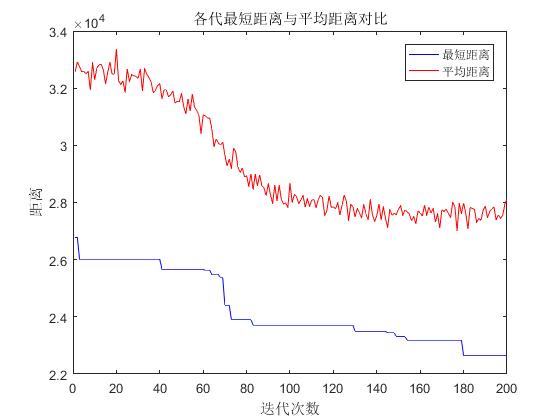


图2 最短距离与平均距离对比图

结果分析：

随着迭代次数的增加，路径逐渐优化，最短距离与平均距离都逐渐缩短，迭代180次左右达到比较好的结果。