МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| профессор |  |  |  | Скобцов Ю. А. |
| должность, звание |  | подпись, дата |  | фамилия, инициалы |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6 |
| «Оптимизация путей на графах с помощью муравьиных алгоритмов» |
| по дисциплине: ЭВОЛЮЦИОННЫЕ МЕТОДЫ ПРОЕКТИРОВАНИЯ ПРОГРАММНО-ИНФОРМАЦИОННЫХ СИСТЕМ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4936 |  |  |  | Нестеренко М.Ю. |
|  |  |  | подпись, дата |  | фамилия, инициалы |

Санкт-Петербург 2022

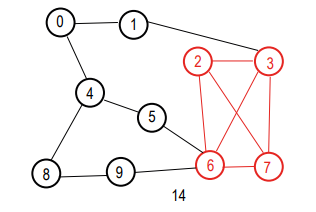
**Цель работы**

Решение задач комбинаторной оптимизации с помощью муравьиных алгоритмов на примере задачи коммивояжера. Графическое отображение результатов оптимизации

**Вариант задания 14**

**Часть 1**

1. Создать программу, использующую МА для решения задачи поиска гамильтонова пути. Индивидуальное заданию выбирается по таблице В.1 в Приложении В согласно номеру варианта;



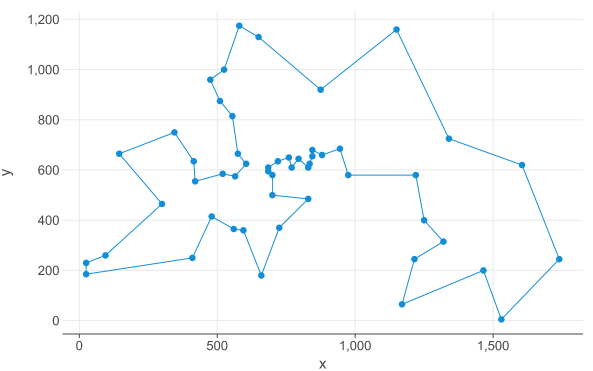
1. Представить графически найденное решение. Предусмотреть возможность пошагового просмотра процесса поиска решения;
2. Сравнить найденное решение с представленным в условии задачи оптимальным решением.

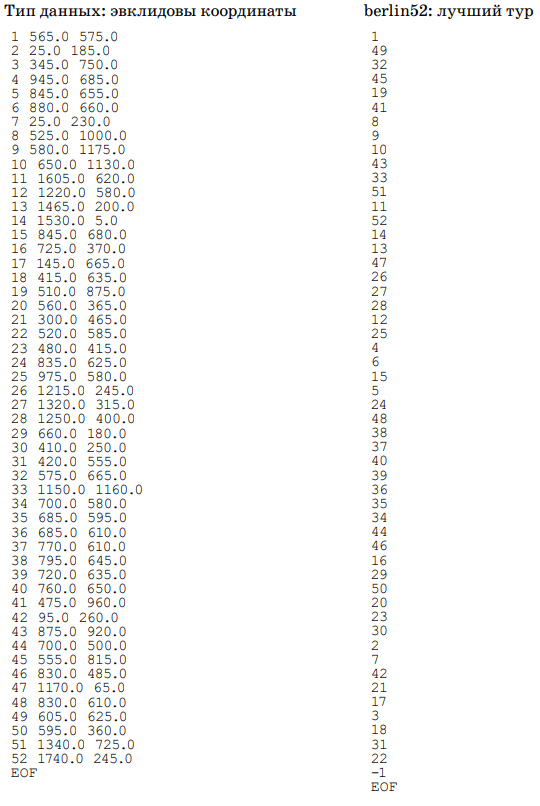
**Часть 2**

1. Реализовать с использованием муравьиных алгоритмов решение задачи коммивояжера по индивидуальному заданию согласно номеру варианта (см. таблицу 3.1. и приложение Б.);
2. Представить графически найденное решение;

Сравнить найденное решение с представленным в условии задачи оптимальным решением и результатами, полученными в лабораторной работе №3.

Маршрут: Berlin52





**Теоретические сведения**

Муравьиные алгоритмы (МА) основаны на использовании популяции потенциальных решений и разработаны для решения задач комбинаторной оптимизации, прежде всего, поиска различных путей на графах. Кооперация между особями (искусственными муравьями) здесь реализуется на основе моделирования. При этом каждый агент, называемый искусственным муравьем, ищет решение поставленной задачи. Искусственные муравьи последовательно строят решение задачи, передвигаясь по графу, откладывают феромон и при выборе дальнейшего участка пути учитывают концентрацию этого фермента. Чем больше концентрация феромона в последующем участке, тем больше вероятность его выбора.

**Код программы**

Код программы написан на языке программирования Matlab.

**main.m**  
clc;

clear;

model = CreateModel();

CostFunction = @(tour) getTourLength(tour, model);

nVar = model.n;

Q = 1;

tau0 = Q / (nVar\*mean(model.D(:))); % Start pheromone value

% Parameters

MaxIt = 100; % количество итераций

nAnt = 50; % размер популяции

alpha = 1; % жадность

beta = 2; % стадность

rho = 0.2; % коэффициент испарения

% Initialization

eta = 1./model.D; % 1/L Matrix

tau = tau0\*ones(nVar, nVar); % Pheromone Matrix

BestCost = zeros(MaxIt, 1); % Array to Hold Best Cost Values

% Empty Ant

empty\_ant.Tour = [];

empty\_ant.Cost = [];

% Ant Colony Matrix

ant = repmat(empty\_ant, nAnt, 1);

% Best Ant

BestSol.Cost = inf;

best = 99999999.0;

% ACO Main Loop

for it = 1:MaxIt

% Move Ants

for k = 1 : nAnt

ant(k).Tour = randi([1 nVar]);

for l = 2:nVar

i = ant(k).Tour(end);

P = tau(i, :).^alpha.\*eta(i, :).^beta;

P(ant(k).Tour) = 0;

P = P/sum(P);

j = randWeight(P);

ant(k).Tour = [ant(k).Tour j];

end

ant(k).Cost = CostFunction(ant(k).Tour);

if ant(k).Cost<BestSol.Cost

BestSol = ant(k);

end

end

% Update pheromone values

for k = 1:nAnt

tour = ant(k).Tour;

tour = [tour tour(1)];

for l = 1:nVar

i = tour(l);

j = tour(l+1);

tau(i, j) = tau(i, j)+Q/ant(k).Cost;

end

end

% calc Evaporation

tau = (1-rho)\*tau;

% Store best cost

BestCost(it) = BestSol.Cost;

disp(['iteration = ' num2str(it) ' | solution = ' num2str(BestCost(it))]);

if BestCost(it) <= best

best = BestCost(it);

end

end

figure(1);

myPlot(BestSol.Tour, model, best);

**createModel.m**

function model = CreateModel()

%x = [1 0 2 0 2];

%y = [2 1 1 0 0];

citiCoordinates = [

1 565 575;

2 25 185;

3 345 750;

4 945 685;

5 845 655;

6 880 660;

7 25 230;

8 525 1000;

9 580 1175;

10 650 1130;

11 1605 620;

12 1220 580;

13 1465 200;

14 1530 5;

15 845 680;

16 725 370;

17 145 665;

18 415 635;

19 510 875;

20 560 365;

21 300 465;

22 520 585;

23 480 415;

24 835 625;

25 975 580;

26 1215 245;

27 1320 315;

28 1250 400;

29 660 180;

30 410 250;

31 420 555;

32 575 665;

33 1150 1160;

34 700 580;

35 685 595;

36 685 610;

37 770 610;

38 795 645;

39 720 635;

40 760 650;

41 475 960;

42 95 260;

43 875 920;

44 700 500;

45 555 815;

46 830 485;

47 1170 65;

48 830 610;

49 605 625;

50 595 360;

51 1340 725;

52 1740 245;

];

x = citiCoordinates(:,2);

y = citiCoordinates(:,3);

n = numel(x);

D = zeros(n, n);

for i = 1:n-1

for j = i+1:n

D(i, j) = sqrt((x(i)-x(j))^2+(y(i)-y(j))^2);

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

end

end

model.n = n;

model.x = x;

model.y = y;

model.D = D;

end

**getTourLength.m**

function L = getTourLength(tour, model)

n = numel(tour);

tour = [tour tour(1)];

L = 0;

for i = 1:n

L = L+model.D(tour(i), tour(i+1));

end

end

**myPlot.m**

function myPlot(tour, model, best)

tour = [tour tour(1)];

plot(model.x(tour), model.y(tour), 'k-o', ...

'MarkerSize', 5, ...

'MarkerFaceColor', 'r', ...

'LineWidth', 1);

xlabel('x');

ylabel('y');

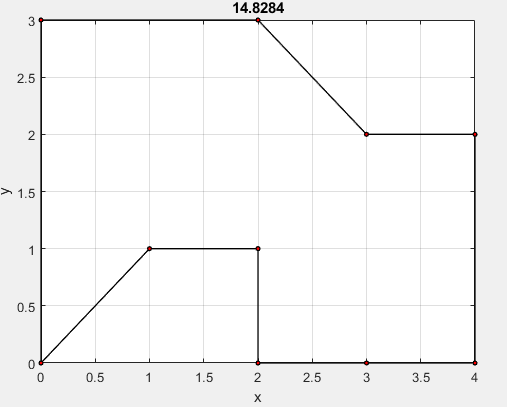
title(num2str(best));

grid on;

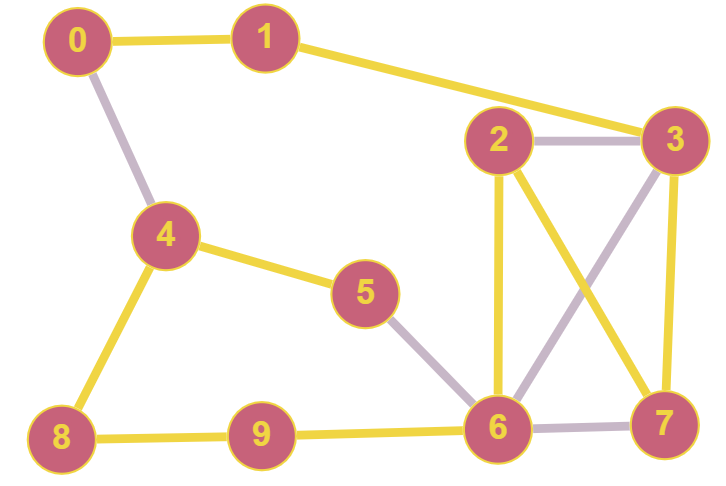
end

**Результат выполнения**

Найденный гамильтонов путь:



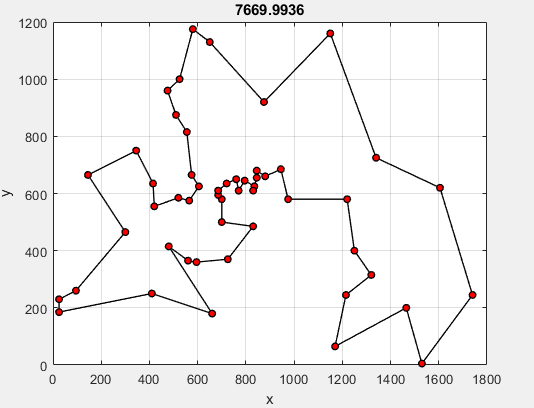
Решение, найденное с помощью Graph Online:



Найденный путь является оптимизированным.

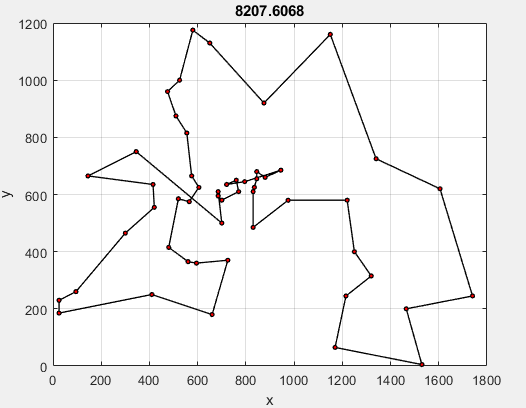
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| α | β | ρ | Nпок | N |
| 1 | 2 | 0.2 | 100 | 50 |

Найденное решение: 7669.9936

****

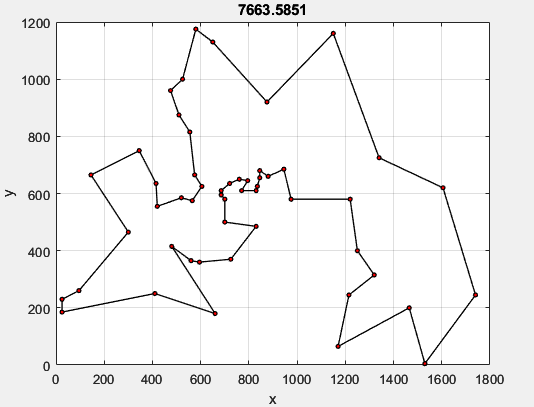
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| α | β | ρ | Nпок | N |
| 1 | 1 | 0.5 | 200 | 50 |

Найденное решение: 8207.6068



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| α | β | ρ | Nпок | N |
| 3 | 4 | 0.2 | 100 | 50 |

Найденное решение: 7663.5851



**Зависимость работы муравьиного алгоритма от параметров**

При увеличении размера популяции муравьев (Nпок) точность алгоритма экспоненциально возрастает. Однако, при увеличении итераций, существенно возрастает время, необходимое для вычислений, поэтому оптимальным является значение близкое к 100.

При увеличении размера популяции колонии (N) точность алгоритма экспоненциально растет. Так как большее количество муравьев амортизирует точность и дает больше возможных вариантов в некоторых ситуациях. Однако, заметного прироста при сильном увеличении не наблюдается. Оптимальным значением является 50 и его окрестности.

При увеличении коэффициента испарения феромона (ρ) точность алгоритма имеет неоднозначное поведение. Поскольку при слишком больших и маленьких значениях муравьи начинаются непропорционально опираться только на один из двух параметров (феромоны и эвристика), то найденному решения в большинстве случаев будет не хватать недостающего влияния параметра. Оптимальным являются значения ρ ∈ [0,5; 0,25]

**Сравнение с классическим ГА**

Муравьиный алгоритм имеет большое преимущество перед классической реализации ГА для задачи коммивояжера. МА так же имеет более четкую настройку, в отличие от классического ГА, что позволяет подстраивать алгоритм в зависимости от представленного графа.

Алгоритм при его правильной настройке выдает результат очень близкий к оптимальному. Добиться того же с помощью эволюционного алгоритма из 3-й лабораторной сложно.

Также, эволюционный алгоритм во многом опирается на случайность, что в МА минимизировано и делает результаты более стабильными. Тогда как ГА от запуска к запуску может выдавать сильно отличающиеся результаты.

**Вывод**

В ходе выполнения лабораторной работы были получены основные навыки разработки и использования муравьиного алгоритма для решения задачи коммивояжера. А также, выражены зависимости используемых констант и точности выполнения алгоритма.