REPORT

LABORATORY OF EVOLUTIONARY ALGORITHMS

Ant systems

Authors:

Pawełko Piotr

Ucherek Angelika

MAKRO – Data Science

Date of the exercise: 23.05.2022, 06.06.2022 r.

Date of the report submission: 13.06.2022 r.

1. Simulating the ants colony behavior in an environment

1.1. Source code:

```
Rm=0;
Lm=0;
m=1000;
k=20;
d=2;

for i=1:m
    Pr=((Rm+k)^d)/((Rm+k)^d+(Lm+k)^d);
    Pl=1-Pr;

    r=rand();
    if r <= Pr
        Rm=Rm+1;
        Lm=Lm;
    else
        Rm=Rm;
        Lm=Lm+1;
    end
end
disp('Number of ants that have used the right branch')
disp(Rm)
disp('Number of ants that have used the left branch')
disp(Lm)
```

1.2. Results:

```
Number of ants that have used the right branch
   187

Number of ants that have used the left branch
   813
```

2. Simulating the ants colony behavior in an environment where the nest and the food are separated by a double, asymmetric bridge.

2.1. Source code:

```
shortest_way = 2;
longer_way = 3;
alpha = 1;
beta = 5;
rho = 0.5;
Rm=0;
Lm=0;
m=1000;
k=20;
d=2;
Pr1=[];
```

```
for i=1:m
    % The probability PR(m) with which the (m + 1)-th ant selects the
    right branch is:

    Pr=((Rm/shortest_way+k)^d)/((Rm/shortest_way+k)^d+(Lm/longer_way+k
    )^d);
    Pr1=[Pr1, Pr];
    Pl=1-Pr;

    r=rand();
    if r <= Pr
        Rm=Rm+1;
        Lm=Lm;
    else
        Rm=Rm;
        Lm=Lm+1;
    end
end
plot([1:1000],Pr1); title('Probability of choosing the right bridge
branch')
xlabel('Ants');ylabel('Probability')
disp('The number of ants that used the right bridge branch:')
disp(Rm)
disp('The number of ants that used the left bridge branch:')
disp(Lm)
```

2.3. Results

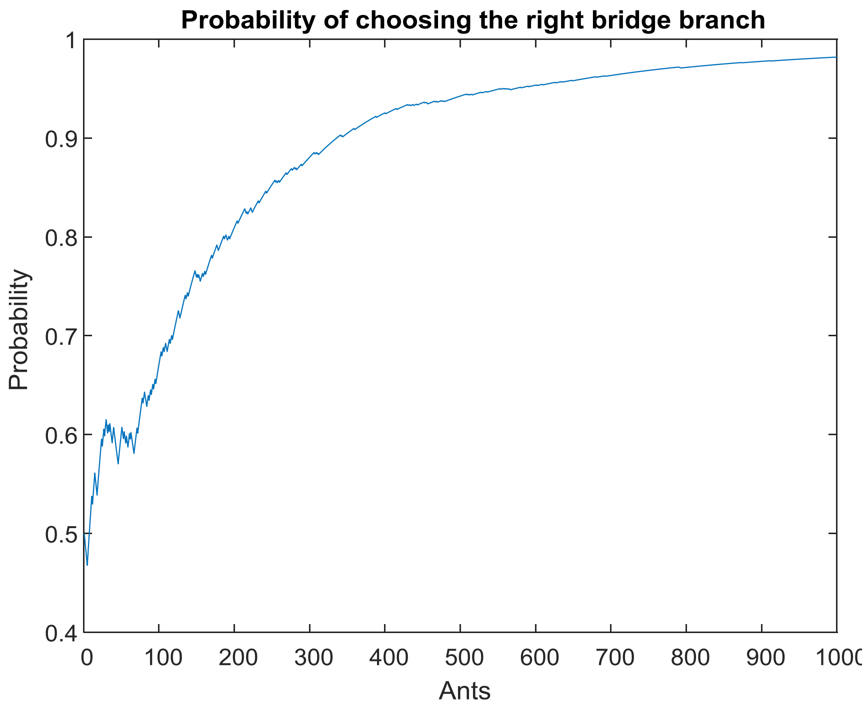

Figure 1. Probability of choosing the right branch.

```
The number of ants that used the right bridge branch:
   873

The number of ants that used the left bridge branch:
   127
```

3. Implementation the Ant System to solve the traveling salesman problem.

During the implementation of ant systems, the traveling salesman problem was based on the same city map that was used during the first laboratory classes on genetic algorithms.

3.1. Source code:

```
x = [0 3 6 7 15 12 14 9 7 0];
y = [1 4 5 3 0 4 10 6 9 10];
Coords = [x ; y]';
N = size(Coords,1);


%create a distance matrix
Distances = zeros(N,N);
for i=1:N
    for j=1:N
        Distances(i,j)= ((x(i) - x(j)).^2 + (y(i) - y(j)).^2).^(1/2);
    end
end


alpha = 1;
beta = 5;
p = 0.5;
m = N;
tmax = 100;
dmax = max(max(Distances));
Tau0 = 1/dmax;
Tau = Tau0*(ones(N,N)-diag(ones(1,N)));


for t=1:tmax
    ants = zeros(m,N);
    ants(:,1) = randi([1 N], [m 1]);
    % Order of visting cities
    for i=1:N-1
        ants(:,i+1) = selectCity(ants(:,1:(i)), m, alpha, beta, Tau,
Distances, N);
    end



% Calculate whole distance for each ant
    ants_rout = zeros(m,1);
    for i=1:m
        for j=1:N-1
            ants_rout(i) = ants_rout(i) +
Distances(ants(i,j),ants(i,j+1));
        end
        ants_rout(i) = ants_rout(i) + Distances(ants(i,N),ants(i,1));
    end
```

```
    % Calculate the total quantity of phermone
     dTau = zeros(N,N);
     for i=1:m
         for j=1:N-1
             dTau(ants(i,j),ants(i,j+1)) = dTau(ants(i,j),ants(i,j+1))
+ 1/ants_rout(i);
         end
         dTau(ants(i,N),ants(i,1)) = dTau(ants(i,N),ants(i,1)) +
1/ants_rout(i);
     end
     dTau = dTau + dTau';
     Tau = (1-p).*Tau + dTau;
end
index = find(ants_rout == min(ants_rout));
graph_form(ants(index(1),:), ants_rout(index(1)), Coords, N)

disp('Sequence of cieties:')
disp(ants(index(1),:))
disp('Total distance: ')
disp(ants_rout(index(1)))
```

3.2. Function definitions

```
    function city = selectCity(ants, m, alpha, beta, T, D, N)
     cities = randperm(N, N);
     city = zeros(m,1);

     for i=1:m
         pos = ants(i, end);
         remaining = cities(~ismember(cities,ants(i,:)));
         A = T(pos,remaining).^alpha .* 1./D(pos,remaining).^beta;
         A = A./sum(A);

         if length(A) == 1
             city(i) = remaining(1);
         else
             r = rand;
             threshold = 0;
             j = 1;
             while threshold <= r
                 threshold = threshold + A(j);
                 j = j + 1;
             end
             city(i) = remaining(j-1);
         end
     end
end


       function graph_form(route,distance,Coords,N)
       %graphic representation of solution
           x = Coords(:,1);
           y = Coords(:,2);
           figure
       %towns position with their 'names'
               scatter(x,y,'o r','filled')
               Names = 1:N;
               Names = cellstr(num2str(Names(:)));
               text(x, y, Names)
```

```
   r = route([i i+1]);
        line([x(r(1)),x(r(2))], [y(r(1)),y(r(2))], 'LineWidth',
0.8,...
            'Color', 'magenta');
    end
    r = route([1 N]);
    line([x(r(1)),x(r(2))], [y(r(1)),y(r(2))], 'LineWidth', 0.8,...
            'Color', 'magenta');
%solution description
    title(['Ant Systems solution' newline ...
            'Map of cities with the best route marked' newline ...
            'The minimal tour distance is equal to '
num2str(distance)]);
    axis([min(x)-1 max(x)+1 min(y)-1 max(y)+1])

end
```

3.3. Results

Minimal total distence traveled:

```
55.0441
```

Sequence of cities to be visited ensuring the minimal total distance traveled:

```
2    3    4    8    6    5    7    9    10    1
```

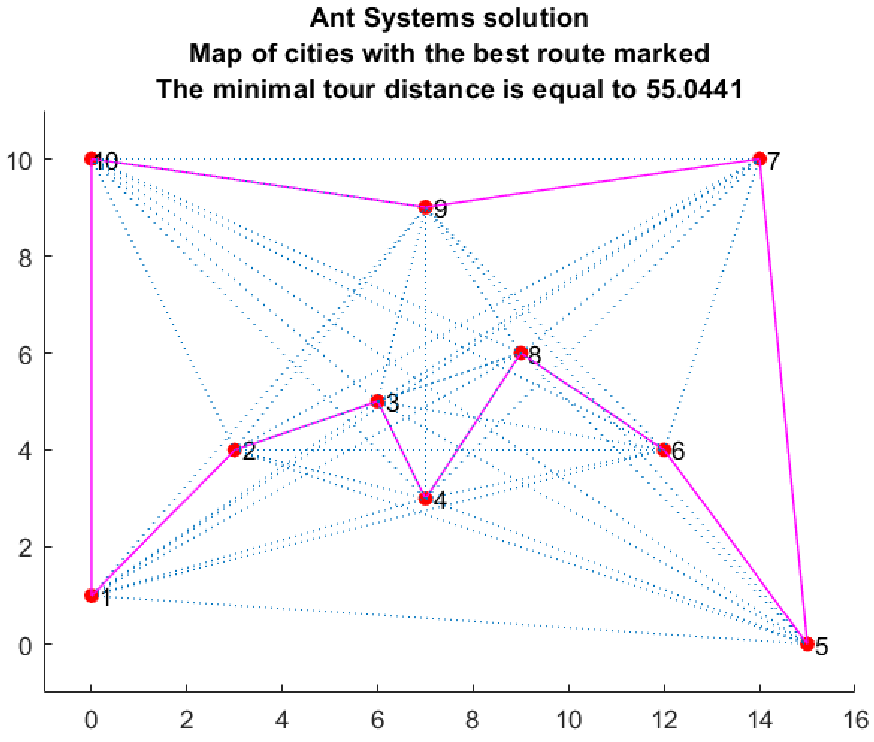

Figure 2. An example of a city network with a Ant systems solution.

Discussion of results and conclusions

It can be noticed that in tasks 1 and 2 concerning the simulation of ant colony behavior, the number of ants choosing one of the two branches is not proportional, that is, the vast majority of all ants chose one branch. In the event that the right branch of the bridge had a shorter route, the probability of selecting this branch initially increased rapidly with the increase of iteration, and then stabilized towards 1.

In the case of the task of determining the optimal route for the traveling salesman problem, the minimum total distance to be traveled by a colony of ants is exactly the same as in the case of the traveling salesman problem solved by the genetic algorithm. It follows that ant systems can also be a very useful tool in the event of problems with determining the most optimal route. It should also be noted that when implementing the traveling salesman problem using genetic algorithms, 1000 iterations were needed to determine the optimal route, while in the case of ant systems, already 100 iterations of the program operation is sufficient to determine the shortest route. This may suggest that ant systems are a more optimal and efficient tool for solving routing problems than genetic algorithms. Based on the conducted and earlier laboratories, it can be concluded that the algorithms modeled on the natural world are a very good and perspective tool for solving common problems.