Tutorial on Ant Colony Optimization

Budi Santosa

Professor at Industrial Engineering

Institut Teknologi Sepuluh Nopember, ITS

Surabaya

Email: budi_s@ie.its.ac.id

http: bsantosa.com

Ant Colony Optimization (ACO) studies artificial systems that take inspiration from the *behavior of real ant colonies* and which are used to solve discrete optimization problems." First introduced by Marco Dorigo in 1992. Originally applied to Traveling Salesman Problem. Natural behavior of ants have inspired scientists to mimic insect operational methods to solve real-life complex optimization problems. By observing ant behavior, scientists have begun to understand their means of communication. Ant-based behavioral patterns to address combinatorial problems - first proposed by Marco Dorigo.

Ants secrete *pheromone* while   traveling from the nest to food, and vice versa in order to communicate with one another to find the shortest path
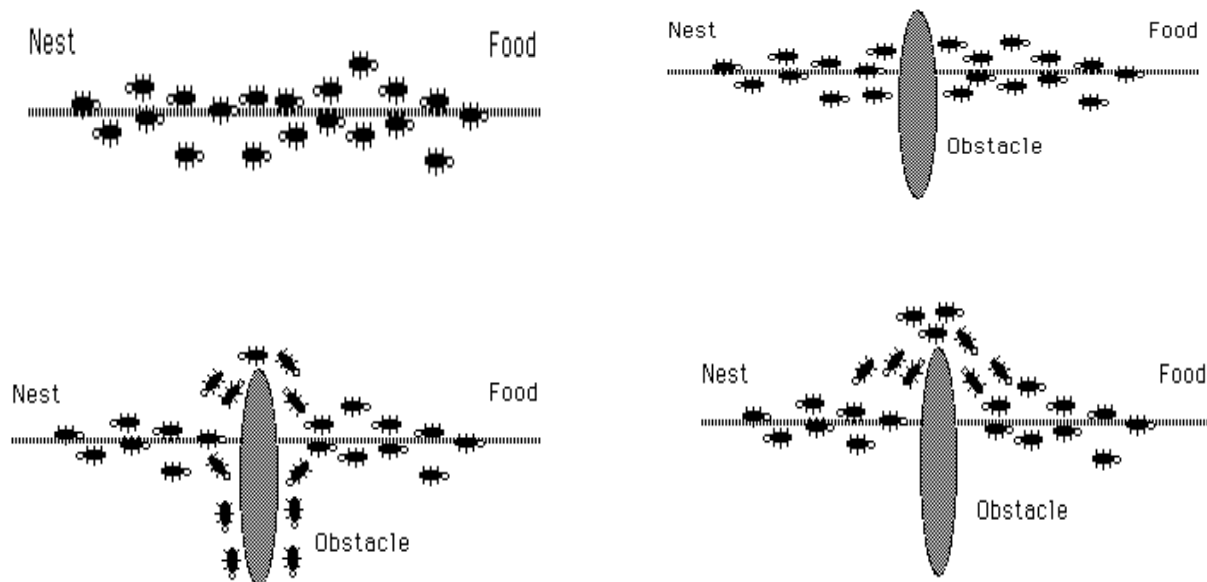
Ants are forced to decide whether they should go left or right, and the choice that is made is a random decision. Pheromone accumulation is faster on the shorter path. The difference in pheromone content between the two paths over time makes the ants choose the shorter path.

The more ants follow a trail, the more attractive that trail becomes for being followed. Different optimization problems have been explored using a simulation of this real ant behavior

An ant k at node r will choose the destination node s at a later stage with probability

$$p_k(r,s) = \begin{cases} \tau(r,s)^\alpha \eta(r,s)^\beta / \sum_{u \in M_k} \tau(r,u)^\alpha \eta(r,u)^\beta, & for\ s \in M_k \\ 0, & otherwise \end{cases} \qquad (8.1)$$

where α indicates the degree of importance of pheromone, β-degree visibility and $u \in M_k$ is a choice that belongs ant k (neighborhood) when he was at node r. Neighborhood of ant k at node r will contain all the nodes that can be addressed directly connected to the node r, except nodes that previously already visited. Visibility usually be replaced with 1 / (distance between node r to node s)

$$\Delta \tau_{i,j}^{(k)} = \begin{cases} \frac{cf_{best}}{f_{worst}}, & jika\ (i,j) \in lintasan\ terbaik\ global \\ 0, & untuk\ yang\ lain \end{cases} \qquad (8.5)$$

An ant k when passing a segment will leave pheromone . The amount of pheromone contained in the segment ij after passed by ant k is given by the formula

$$\tau_{i,j} \leftarrow \tau_{i,j} + \Delta \tau^k \qquad (8.2)$$

With the increasing value of pheromone on segment i-j, then the probability of this segment to be chosen by ants at the next iteration increases. Once a node is passed then the pheromone evaporation will occur with the following rules

$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j}, j\ ;\ \forall (i,j) \in A \qquad (8.3)$$

where ρ ∈ (0, 1] is the evaporation rate parameter and A represents segments that have been passed by ant k as part of the path from the nest to get to the food. The decline in the amount of pheromone allows ants to explore a different path during the search process. It also would eliminate the possibility of choosing the path that is not good enough. The amount of pheromone is added to the segment i - j by ant k is given as

$$\Delta \tau_{i,j}^{(k)} = \frac{Q}{L_k} \qquad (8.4)$$

where Q is a constant and Lk is the total distance traveled by ant k after returning to the nest (in the case of the TSP is the path traversed by the salesman). Q value is usually

determined by the user. In many cases the value of Q= 1 gives good results. Or it could also be implemented in the following way

$$\Delta\tau_{i,j}^{(k)} = \begin{cases} \frac{cf_{best}}{f_{worst}}, & if\ (i,j) \in best\ global\ path \\ 0, & otherwise \end{cases} \quad (8.5)$$

where fbest is the best value of the objective function and fworst is the worst value of the objective function. While c is a constant to control the scale of the global pheromone updating. The higher value of c the more pheromone is added to the global path of the best and the better ability to exploit. Equation (8.5) is intended to provide more amount of pheromone on the path of global best (overall).

ACO Algorithm for TSP

- Randomly place ants at the cities

- For each ant:

    1. Choose a not yet visited city   until a tour is completed

    2. optimize the tour

    3. Update pheromone $\tau_{ij} + = 1/length(tour)$

- Evaporate Pheromone $\tau_{ij} = (1-\rho)\cdot\tau_{ij}$

Example

Suppose we have a TSP problem with 5 cities. The distance between cities is given by the following matrix

d =

	1	2	3	4	5
1	0	10	12	11	14
2	10	0	13	15	8
3	12	13	0	9	14
4	11	15	9	0	16
5	14	8	14	16	0

Determine the best route with minimum total distance by using the ACO.

Step 1:
Suppose the number of ants, N = 3. We determine the visibility (h) between the cities by taking the inverse of the distance, 1 / d. we get

h =

	1	2	3	4	5
1	0	0.1000	0.0833	0.0909	0.0714
2	0.1000	0	0.0769	0.0667	0.1250
3	0.0833	0.0769	0	0.1111	0.0714
4	0.0909	0.0667	0.1111	0	0.0625
5	0.0714	0.1250	0.0714	0.0625	0

Then for each segment between cities we give the initial pheromone value τ_0, suppose this value is the same for all segments, namely 1.

thoinit =

```
1  1  1  1  1
1  1  1  1  1
1  1  1  1  1
1  1  1  1  1
1  1  1  1  1
```

Step 2:
Determine the probability of moving from one city to another. In this case city 1 is designated as a departure city. Then from the last city visited we are going back to city 1. As city 1 is chosen as the beginning city 1 then city will be taboo to visit again, so that the level of visibility of the city 1 is made =0.

h =

	1	2	3	4	5
1	0	0.1000	0.0833	0.0909	0.0714
2	0	0	0.0769	0.0667	0.1250
	0	0.0769	0	0.1111	0.0714
3	0	0.0667	0.1111	0	0.0625
4					

5 0   0.1250   0.0714   0.0625   0

Then we calculate the possibility to visit other cities from city 1 using the formula in equation (8.9) .

$$p_k(r,s) = \begin{cases} \tau(r,s)^\alpha \eta(r,s)^\beta \Big/ \sum_{u \in M_k} \tau(r,u)^\alpha \eta(r,u)^\beta, & if\ s \epsilon M_k \\ 0, & otherwise \end{cases}$$

(8.9)

Where $p_k(r, s)$ is probability move from city r to city s of ant k, τ is pheromone level, $\eta(r, s)$ is visibility of cities r and s, M_k set of cities possibly visited by ant k, α weight of pheromone τ and β is weight to control visibility. Using $\alpha=1$ and $\beta=2$.

Suppose ants 1 from city 1 will choose the next city, then we calculate

$\tau(1,s)^1 \eta(1,s)^2$ from 1 city to all towns
1 x 0:12 = 0:01 (from city 1 to city 2)
1 x 0.08332 = 0.0069 (from city 1 to city 3)
1 x 0.09092 = 0.0083 (from city 1 to city 4)
1 x 0.07142 = 0.0051 (from city 1 to city 5)

$\tau \times \eta^\beta$

$\alpha = 1 \quad \beta = 2$

0.01 + 0.0069 + 0.0083 0.0051
The total number of $\tau(1,s)^1 \eta(1,s)^2$ = 0.0303

So probability of ant 1 going from city 1 to all cities respectively calculated by the formula

$$p_1(1,s) = \tau(1,s)^1 \eta(1,s)^2 / \Sigma \tau(1,s)^1 \eta(1,s)^2$$

Obtained the probability to go from

City 1 to 2 = 0.01 / 0.0303 = 0.3299

City 1 to 3 = 0.0069 / 0.0303 = 0.2291

City 1 to 4 = 0.0083 / 0.0303 = 0.2727

City 1 to 5 = 0.0051 / 0.0303 = 0.1683

The cumulative number of these probabilities is

City 2 = 0.3299

City 3 = 0.5590

City 4 = 0.8317

City 5 = 1

Step 3:

We generate a random number r to determine the choice of cities to be visited by comparing it with the value of the cumulative probability. Suppose the generate random numbers r = 0.6841, then we can compare it with the cumulative value of probability. This number is located below the 0.8317 on the cumulative amount. Thus the city to visit is city 4.

Since city 4 has been selected then 4th column of the matrix, the visibility set to 0 so that

$$
h = \begin{array}{ccccc}
1 & 2 & 3 & 4 & 5 \\
0 & 0.1000 & 0.0833 & 0 & 0.0714 \\
0 & 0 & 0.0769 & 0 & 0.1250 \\
0 & 0.0769 & 0 & 0 & 0.0714 \\
0 & 0.0667 & 0.1111 & 0 & 0.0625 \\
0 & 0.1250 & 0.0714 & 0 & 0
\end{array}
$$

From here we calculate again the value of τ
$\tau(4,s)^1 \eta(4,s)^2$ from city 4 to all of the cities (except city 1)

1x 0.06672 = 0.0044 (from city 4 to city 2)

1x 0.11112 = 0.0123 (from city to city 4 3)

1x 0.02 = 0.0 (from city 4 to city 4)

1x 0.07142 = 0.0039 (from city to city 4 5)

The total number of $\tau(1,s)^1\eta(1,s)^2 = 0.0207$

So probability ant 1 going from city 4 to other cities respectively calculated by the formula
$$p_1(4,s) = \tau(4,s)^1\eta(4,s)^2 / \Sigma\tau(4,s)^1\eta(4,s)^2$$

Obtained the probability to go from

City 4 to 2 0.0044 / 0.0207 = 0.2147

City 4 to 3 0.0123 / 0.0207 = 0.5965

City 4 to 4 0.0 / 0.0207 = 0.0

City 4 to 5 0.0039 / 0.0207 = 0.1887

Suppose we generate a random number r = 0.4024.
Total cumulative

City 2 = 0.2147
City 3 = 0.8113

Because the random numbers r = 04024 <0.8113, then the next city selected is city 3. Thus the route formed for ant 1 so far is 1-4- 3.

This step is repeated for all of ants. Of each route passed through by every ant, we will know the total distance traversed by each ant. Suppose the calculation we did in the first iteration for all 3 ants we obtained the following

Ants 1: 1-4 - 3 - 5-2 -1
Ants 2: 1 - 4-2 - 5 - 3-1
Ants 3: 1 - 4-5 - 2 - 3-1

With a total distance of each route is 52, 60 and 60. The total distance is used to update the pheromone level τ using the formula (8:10).
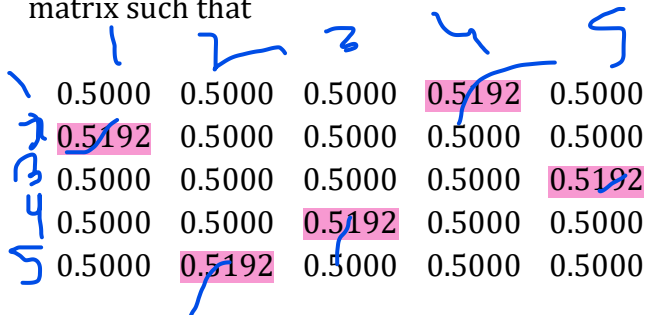$$\tau_{r,s} \leftarrow (1-\rho)\tau_{r,s} + \Sigma_{k=1}^{N}\Delta\tau_{r,s}^k \quad (8.10)$$

Step 4:

All ants will provide additional pheromone to the pheromone matrix according to the route passed through.

Ants will provide additional pheromone by 1/total distance i.e 1/52 = 0.0192. In addition, evaporation will occur at (1-0.5) x1 = 0.5, where ρ, evaporation coefficient = 0.5. So the total pheromone becomes 0.5192. In other segments which are not passed by ants, only evaporation will occur without the addition of pheromone.

These changes will be given to the segments passed through by ants 1 at the pheromone matrix such that

```
   1      2      3      4      5
1  0.5000 0.5000 0.5000 0.5192 0.5000
2  0.5192 0.5000 0.5000 0.5000 0.5000
3  0.5000 0.5000 0.5000 0.5000 0.5192
4  0.5000 0.5000 0.5192 0.5000 0.5000
5  0.5000 0.5192 0.5000 0.5000 0.5000
```

Then ant 2 will provide additional pheromone for 1/60 = 0.0167 in segments traversed (Ant 2: 1-4- 2-5-3-1), as well as reduced the evaporation rate (1-0.5) x1 = 0.5. So will we get the pheromone matrix

```
0.5000 0.5000 0.5000 0.5359 0.5000
0.5192 0.5000 0.5000 0.5000 0.5167
0.5167 0.5000 0.5000 0.5000 0.5192
0.5000 0.5167 0.5192 0.5000 0.5000
0.5000 0.5192 0.5167 0.5000 0.5000
```

$$0.5192 + 0.0167$$

Ant 3also add an additional pheromone 1/60 = 0.0167. So that for the segments traversed (Ant 3: 1 - 4-5 - 2 - 3-1), the pheromone matrix becomes

```
0.5000 0.5000 0.5000 0.5526 0.5000
0.5192 0.5000 0.5167 0.5000 0.5167
0.5334 0.5000 0.5000 0.5000 0.5192
0.5000 0.5167 0.5192 0.5000 0.5167
0.5000 0.5359 0.5167 0.5000 0.5000
```

Then in the next iteration we repeat from step 2 until the maximum number of iterations is achieved.

Example
Running the code in Matlab.

>> d=[0   10   12   11   14

10   0   13   15   8

12   13   0   9   14

11   15   9   0   16

14   8   14   16   0]

>> [bestrute,mincost]=acs_budi(d,20,10)

bestrute =

   1   4   3   5   2   1

mincost =

   52

Matlab Code

```
function [besttour,mincost]=acs_budi(d,iter,n_ants);
%iter=100;%number of cycles.
% d is   n xn distance matrix
%xy is coordinates of cities m x 2 (n is number of cities)
m=n_ants;%200;%number of ants.
n=length(d);%jumlah kota.
e=.5;%evaporation coefficient.
```

```matlab
alpha=1;%order of effect of ants' sight.
beta=2;%pangkat order of trace's effect.
%for i=1:n
 %    for j=1:n
  %  d(i,j)=sqrt((xy(i,1)-xy(j,1))^2+(xy(i,2)-xy(j,2))^2);
    %menghitung jarak antar kota
   % end
%end

for i=1:n%generating sight matrix.
    for j=1:n
        if d(i,j)==0
            h(i,j)=0;
        else
            h(i,j)=1/d(i,j);%inverse distance
        end
    end
end

thoinit=.1*ones(n);%initial tho.
%el=.96;%coefficient of common cost elimination.
for i=1:iter
    for i=1:m
    %ants initial placing.
    rute(i,1)=1;%ants start from city 1
    end
    %ant tour
    for i=1:m %for all ants
    mh=h;%invers distance matrix
    for j=1:n%next node
        c=rute(i,j);%choose next city to visit
        mh(:,c)=0;%for column c,inv distance=0
        tho=(thoinit(c,:).^beta).*(mh(c,:).^alpha);%update pheromone
        s=(sum(tho));%jumlah tho
        p=(1/s).*tho;%probability
 %       p
        r=rand;
        s=0;
        for k=1:n %number of city
            s=s+p(k);%use roulette wheel
            if r<=s
                rute(i,j+1)=k;%penempatan semut i di simpul berikutnya
                break
                end
            end
        end
    end

    %%%% resulting tour/route
    rute_c=horzcat(rute,rute(:,1));%add first city to the tour/route
   % %compute total distance of path traverse by each ant
    for i=1:m %number of ant
```

```matlab
    s=0;
    for j=1:n-1
       s=s+d(rute_c(i,j),rute_c(i,j+1));
    end

    f(i)=s;
    end
    jaraktot=f;
    [minf,idk]=min(f);
    ter=rute_c(idk,:);
    %f=f-el*min(f);%elimination of common cost.
    %update edge belonging to Lbest

  thoinit=(1-e)*thoinit;
  for i=1:m % for all ant
  %  for i=idk;
   for j=1:n-1 % for all cities
       dt=1/f(i);%inverse total distance, delta pheromone
       %update pheromone

thoinit(rute_c(i,j),rute_c(i,j+1))=thoinit(rute_c(i,j),rute_c(i,j+1))+
dt;%updating traces.
   %    thoinit(rute_c(i,j),rute_c(i,j+1))=(1-
e)*thoinit(rute_c(i,j),rute_c(i,j+1))+dt;%updating traces.

   end
   end
  %thoinit;
  % costoa(i)=mean(cost);
   %[mincost(i),number]=min(jaraktot);
   %besttour(i,:)=rute_c(number,:);
   %iteration(i)=i;
end
besttour=ter;
mincost=jartsp(besttour,d);

function jarak=jartsp(x1,dx)
%fungsi tsp
%input:
    %x1= rute tsp (contoh: 1 2 3 4 5 1, 1=kota asal, 2345=kota tujuan
    %dx= matrik jarak rute tsp
%output:
     %jarak = jarak total rute tsp

[r,c]=size(x1);
k=c-1;%jumlah kota  dalam rute tsp
s=0; %jarak/posisi awal di kota pertama
for j=1:k
   s=s+dx(x1(j),x1(j+1)); %peng akumulasian jarak rute tsp
end
jarak=s;
```

Reference

Santosa, Budi, Metoda Metaheuristik, Konsep dan Implementasi, Guna Widya, Surabaya, Indonesia, 2011.

M. Dorigo and L.M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.

M. Dorigo, V. Maniezzo, and A. Colorni. The ant system optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and CyberneticsPart B*, 26(1):29–41, 1996.