© 2014 IJSRSET | Volume i | Issue i | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

ANT COLONY OPTIMIZATION

Jethva Harikrishna
Department of Computer Engineering L. D. College of Engineering, Ahmedabad (Gujarat-India)

ABSTRACT

Ant colonies, and more generally social insect societies, are distributed systems that, in spite of the simplicity of their individuals, present a highly structured social organization. As a result of this organization, ant colonies can accomplish complex tasks that in some cases far exceed the individual capacities of a single ant.

Real ants are capable of finding the shortest path from their nest to a food source without visual sensing. They are also able to adapt to changes in the environment. "Ant Colony Optimization" is an algorithm which searches for the solution of the problem under consideration in the way similar to real ants. It tries to make use of real ant abilities to solve various optimization problems. In this report study of simple ant algorithms has been done. Also, as an example they are applied on famous Traveling Salesman Problem. Finally, some results are tabulated comparing these algorithms with other optimization heuristics.

Keywords
AOC, TSP, Ant Colony, ACS, MAS

I. INTRODUCTION

Ant colonies have always fascinated human beings. Ants appeared on the earth some 100 millions of years ago, and like human beings, they can be found virtually everywhere on the earth. Ants are undoubtedly one of the most successful species on the earth today, and they have been so for the last 100 million years. What particularly strikes the occasional observer as well as the scientist is the high degree of social organization that these insects can achieve in spite of very limited individual capabilities.

Ant algorithms are inspired by the behavior of ants. They try to simulate mathematically what these ants do in the real environment, and use these simulations to solve other problems of interest. The first algorithm which can be classified within this framework was presented in 1991 (see [1], [2]) and, since then, many diverse variants of the basic principle have been proposed.

ACO belongs to a class of algorithm, whose first member, known as Ant system was initially proposed by 3 men, namely A. Colomi, M. Dorigo , V. Maniezzo(see [1], [2])

Ant algorithms are multi-agent systems in which the behavior of every single agent, called artificial ants (or simply ants), is inspired by the behavior of real ants (see [3]). An agent is an autonomous entity with an ontological commitment and agenda of its own. Ontological commitment is a statement in which the existence of one thing is presupposed or implied by asserting the existence of another. So, Multi-Agent System can be defined as a system composed of several agents, capable of mutual interaction. The interaction can be in the form of message passing or producing changes in their common environment (i.e. direct communication or indirect communication). A very good example of MAS is human organizations and societies.

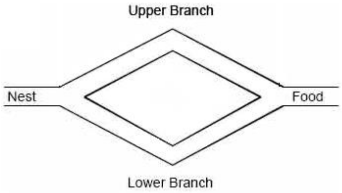


Figure 1: Experimental apparatus for the bridge experiment

The experimental observation is that, after some time (i.e. few minutes) most of the ants use the shortest path, furthermore it is also seen that probability of selecting the shortest branch increases with the increase in difference in the lengths of the two paths.

The emergence of this shortest path selection behavior can be explained in terms of

1) Autocatalysis (i.e. positive feedback) and,
2) Differential path length

And it is made possible by an indirect form of communication known as stigmergy. Stigmergy is a particular form of indirect communication (mediated by local modifications in the environment) used by social insects to coordinate their activities (see [3], [4]). By exploiting the stigmergic approach to coordination, researchers have been able to design a number of successful algorithms in such diverse application fields as combinatorial optimization, routing in communication networks, graph drawing and partitioning, and so on (see [6]). Argentine ants, while going from the nest to the food source and vice versa, deposit a chemical substance known as pheromone, on the ground (see [3]). Now, when they arrive at a decision point that is at a point, then they make a probabilistic choice which is further based on the amount of pheromone they smell on the branches. This behavior has an autocatalytic effect because of the very fact that choosing a path will increase its probability that it will be chosen again by the future ants as they will increase the amount of pheromone on the path which is used by them.

Coming back to the experiment by Goss et al at the beginning of the experiment since there is no pheromone on any of the branches, therefore ants going from nest to the food will choose any one of the two branches with equal probability, but due to the differential branch length, the ants choosing the shorter path/branch will reach to the food source earlier (so this path now, has the higher probability of getting selected due to the fact that ants released the pheromone trail in the forward journey) and will therefore choose the shorter path with higher probability then the longer one. New pheromone will be released on the chosen path making it even more attractive for the subsequent ants. As the process iterates, rate of deposition of the pheromone on the shorter branch is higher than that is on the longer path. Therefore, the shorter path is more and more frequently selected by the ants until, eventually all ants end up using the short path. (see [3], [5])

In this report simple ant colony optimization techniques are discussed. Some stress is laid on, how it is engineered and put to work so that a colony of artificial ants can find good solutions to difficult optimization problem. Also how these ant algorithms can be modified in order to enlarge the domain of application of these algorithms. In the last chapter a comparative study has been done to compare the performance of ACO with other optimization algorithms.

II. METHODS AND MATERIAL

Simple Ant Colony Optimization

2.1 Simple-ACO

In this section a very simple ant-based algorithm is presented to illustrate the basic behavior of the ACO meta-heuristic and to put in evidence its basic components. The main task of each artificial ant, similarly to their natural counterparts, is to find a shortest path between a pair of nodes on a graph on which the problem representation is suitably mapped.

2.2 Experiments conducted

2.2.1 Single bridge experiment

This experiment was conducted by Deneubourg et al., in 1990, (see the figure above), (see [7]) to study the ants foraging in the controlled conditions. Here a colony of ants (Linepithema humile) is taken and separated by a food source by a bridge (as shown above) where both branches have equal lengths. Ants are then left free to move between the food source and the ant colony. Thereafter, their behaviour is observed and plotted on the graph (as shown above). Observation of the experiments are that initially after a transitory phase all ants converge to a single path that is despite of equal lengths of the paths all ants after some time used the same path. This basically proves the existence of the pheromone, (discussed in section 1.3). Initially, no branch has any pheromone on it, which are therefore selected by the ants with equal probability, but as more and more ants use the upper path the rate of deposition of the pheromone on the upper path is greater, though some ants do take lower path but as more ants choose upper path it bias the choice of future ants and hence after some time all ants use the upper path only.

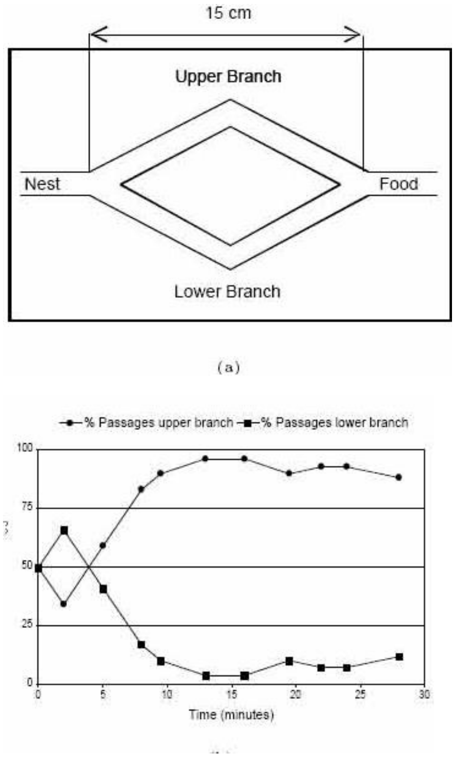


Figure 2: Single bridge experiment. (a) Experimental setup. (b) Results for a typical single trial, showing the percentage of passages on each of the two branches per unit of time as a function of time. Eventually, after an initial short transitory phase, the upper branch becomes the most used. After Deneubourg et al., 1990.

Let Um and Lm be the number of ants using the upper branch and the lower branch respectively, after m ants has crossed the bridge so Um + Lm = m. Now, the probability that the (m+1)th ant chooses upper branch Pu (m) is

$$P_U(m) = \frac{(U_m + k)^h}{(U_m + k)^h + (L_m + k)^h}$$

Where k, h are the model parameters for fitting the experimental measures

h= degree of nonlinearity (n higher → faster triggering for one of the branches)

k= degree of attraction of an unmarked branch (k greater → greater amount of pheromone needed to make a non-random choice)

2.3 Mathematical modelling

Let G = (N, A) be a connected graph with n=|N| nodes. The simple ant colony optimization (S-ACO) algorithm can be used to find a solution to the shortest path problem defined on the graph G, where the solution is a path on the graph connecting a source node‘s’ to the destination node‘d’ and the length of connection can be defined as say, either equal to the physical length of the connection or, using any other conversion factor as specified in the problem under consideration.

Please see the figure below.

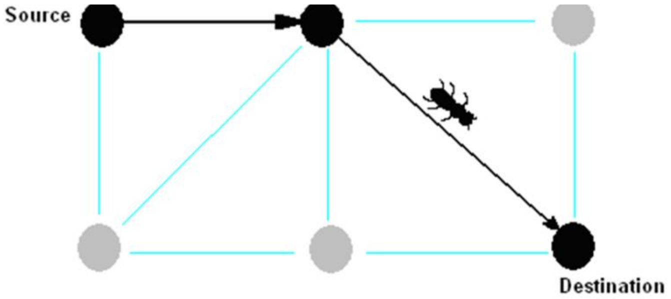


Figure3: ants build solution that is paths from the source to the destination node The ant choosing the dark line will arrive earlier to the destination and will be therefore first to bias search of ants moving back to the source node

To simulate the pheromone trail mathematically a variable τij is introduced, where τ is the artificial pheromone trail associated with the arc (i, j) of the graph. (see [3],[4],[5]) Pheromone trails are read and written by the ants. The amount (i.e. the magnitude) of the pheromone trail is proportional to the utility, as estimated by the ants, of using that arc to build good solutions. So, a path that is used by the ants more frequently will have larger value of τ as compared to the value of τ for a less visited path.

Each ant applies a step-by-step constructive decision policy to build solutions to the problem. At each node local information, maintained on the node itself and/or on its outgoing arcs, is used in a stochastic way to decide the next node to move to. The decision rule of an ant ‘k’ located in node ‘i’ uses the pheromone trail τij, to compute the probability with which it should choose the node ‘j’ as the next node is (see [3],[4]):

(j ϵ N_i, where N_i is the set of one step neighbours of node i)

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}}{\sum_{j \in N_i} \tau_{ij}} & \text{if } j \in N_i \\ 0 & \text{if } j \notin N_i \end{cases}$$

While building the solutions ants deposit the pheromone information on the arcs they use. In S-ACO ants deposit a constant amount $\Delta\tau$ of pheromone (see [3],[4],[5],[6],[7]) . Consider an ant that at time instant t moves from node i to node j. It will change the pheromone trail i.e. the value of τ_{ij} as follows

$$\tau_{ij} \leftarrow \tau_{ij}(t) + \Delta\tau$$

(see [3]) ,[5]) so, by this rule, which simulates real ants' pheromone depositing on arc (i, j), an ant using the arc connecting the node i to the node j increases the probability that the ants will use the same arc in the future. As in the case of real ants, autocatalysis and differential path length are at work to favour the convergence of all the ants to the shortest possible path(and thus to the solution of the problem under consideration.)

The technique is very simple however this simplicity leads to some problems and one of them is: the solution may have path that is locally optimized but not globally. So, to avoid the quick convergence of all the ants towards a sub-optimal path, an evaporation mechanism is added (see [3], [5]) i.e. similar to real pheromone trails, artificial pheromone trails 'evaporate'. In this way the pheromone intensity decreases automatically, and hence favouring the exploration of the different arcs during the search process. The evaporation is carried out in a simple way, decreasing the pheromone trail exponentially,

$$\tau \leftarrow (1-\rho)\ \tau$$

where $\rho \in (0,1]$, at each iteration of the algorithm (see [3]) . Preliminary experiments run with the S- ACO using a simple graph modeling, the apparatus as shown in the Figure 1 have shown that the algorithm effectively finds the shortest path between the simulated nest and the simulated food sources. Also is has been noticed with experiments that as we increase the complexity of the searched graphs, for example increasing the number of branches on different length connecting the food source and the ant colony. Then, the behaviour of the algorithm tend to be less stable and the value given to the parameter becomes critical.

III. RESULT AND DISCUSSION

3.1 Traveling salesman problem (TSP)

In the traveling salesman problem 'n' number of cities are given which are connected by each other and the problem is of finding a shortest closed tour which visits all the cities in such a way that no city is visited twice i.e. the problem is equivalent to finding the shortest Hamiltonian path on graph with 'n' vertices.

3.2. Modeling of the system

3.2.1 Artificial ants or the ant system

(see [8], [9]) An artificial ant is an agent which moves from city to city on a TSP graph. It chooses the city to move to using a probabilistic function both of trail accumulated on edges and of a heuristic value, which was chosen here to be a function of the edges length. Artificial ants probabilistically prefer cities that are connected by edges with a lot of pheromone trail and which are close-by. Initially, m artificial ants are placed on 'm' randomly selected cities. At each time step they move to new cities and modify the pheromone trail on the edges used this is termed as local trail updating.

When all the ants have completed a tour, the ant that made the shortest tour modifies the edges belonging to its tour termed global trail updating, by adding an amount of pheromone trail that is inversely proportional to the tour length.

These are three ideas from natural ant behaviour that we have transferred to our artificial ant colony (see [8]):

(i) The preference for paths with a high pheromone level,
(ii) The higher rate of growth of the amount of pheromone on shorter paths, and
(iii) The trail mediated communication among ants.

Artificial ants were also given a few capabilities which do not have a natural counterpart, but which have been observed to be well suited to the TSP application, they are (see [8]):

(i) Artificial ants can determine how far away cities are, and
(ii) They are endowed with a working memory M_k used to memorize cities already visited (the working memory is emptied at the beginning of each new tour, and is updated after each time step by adding the new visited city).

3.2.2 Mathematical translation of the concepts

There are many different ways to translate the above principles into a computational system in order to solve the TSP.

3.2.2.1 Transition rule

An artificial ant *k* in city “*i”* chooses the city “*j”* move to among those which do not belong to its working memory M_k by applying the following probabilistic formula:

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}(t)]^{\beta}}{\sum_{h \in J_i^k} [\tau_{ih}(t)]^{\alpha} \cdot [\eta_{ih}(t)]^{\beta}}$$

- If j has been visited then p() =0
- Note α β are parameters – trades off global vs local information
- If α is small then the closest cities are favored – classic greedy algorithm
- If β is small only pheromone +ve feedback at work – may choose non-optimal paths too quickly

3.2.2.2 ACS (Ant Colony System) transition rule

An alternative approach to calculate the probability with which an artificial ant *k* in city “*i”* chooses the city “*j”* move to among those which do not belong to its working memory M_k is

$$p_{ij}^{k}(t) = \begin{cases} \arg\max_{u \notin M_k} [\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}(t)]^{\beta} \quad \text{Favors exploiting problem knowledge in trails and heuristic} & \text{if } q \leq q_0 \\ \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}(t)]^{\beta}}{\sum_{h \in J_i^k} [\tau_{ih}(t)]^{\alpha} \cdot [\eta_{ih}(t)]^{\beta}}, \ \alpha=1 \quad \text{(Favors exploration)} & \text{otherwise} \end{cases}$$

Where $\tau_{ij}(t)$ is the amount of pheromone trail on edge (*i*, *j*), $\eta_{ij}(t)$ is a heuristic function, which was chosen to be the inverse of the distance between cities *i* and *t*, α is a parameter that treads off local vs global information, β is a parameter which weighs the relative importance of pheromone trail and of closeness, *q* is a value chosen randomly with uniform probability in [0, 1], q_0 ($0 \leq q_0 \leq 1$) is a parameter. where p_{ij}^{k} is the probability with which ant *k* chooses to move from city *i* to city *j*.

3.2.3 Pheromone trail updating

The pheromone trail is changed both locally and globally (see [3], [5], [8], [10]). Global updating is intended to reward edges belonging to shorter tours. Once artificial ants have completed their tours, the best ant deposits pheromone on visited edges; that is, on those edges that belong to its tour. (The other edges remain unchanged.) The amount of pheromone Δτ(i, *j*) deposited on each visited edge (*i*, *j*) by the best ant is inversely proportional to the length of the tour, the shorter the tour the greater the amount of pheromone deposited on edges. This manner of depositing pheromone is intended to emulate the property of differential pheromone trail accumulation, which in the case of real ants was due to the interplay between the length of the path and continuity of time.

3.2.3.1 Global trail updating

The formula for global trail updating is (see [8], [10]):

$$\tau_{ij}(t+1) = (1-\rho)\ \tau_{ij}(t) + \rho\ \Delta\tau_{ij}(t)$$

$$\Delta\tau_{ij}(t) = Q/L_k$$

Q in a constant and L_k is the total length traveled by the kth ant during the compete tour. ρ is the factor introduced to account for trail evaporation.Global trail updating is similar to a reinforcement learning scheme in which better solutions get a higher reinforcement

3.2.3.2 Local trail updating

Local updating is intended to avoid a very strong edge being chosen by all the ants. Every time an edge is chosen by an ant its amount of pheromone is changed by applying the local trail updating formula (see [10]):

$$\tau_{ij}(t+1) = (1-\rho)\ \tau_{ij}(t) + \rho\ \tau_0$$

$$\tau_0 = 1/(n.L_{nn})$$

where n = number of cities and L_{nn} is a the length of the best greedy tour. Local trail updating is also motivated by trail evaporation in real ants.

4.3 The algorithm

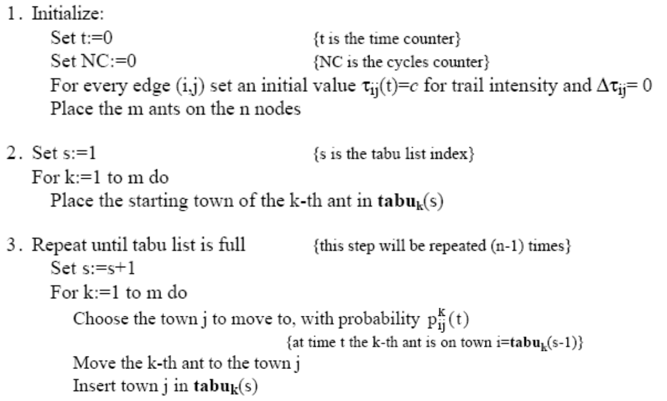
The algorithm to solve the problem can be stated as follows:

At time zero an initialization phase takes place during which ants are positioned on different towns randomly and, initial values $\tau_{ij}(0)$ for trail intensities are set on the edges. The first element of each ant's tabu list is set equal to its starting town. (*tabu list* can be defined as the data structure associated with an ant where it keeps the information of the cities already visited so that it should not go to that city again, and $tabu_k$ means the tabu list on ant k.) Thereafter ants start their journey and moves from city to city by applying the probabilistic rules which further depends upon:

i. pheromone trail τ_{ij}, which gives information on how many ants had already passed from this route
ii. The visibility (as defined by M. Dorigo (see [9])) η_{ij} which says that closer a city more desirable it is.

Setting α to 0 implies that trail level is no longer considered, and a stochastic greedy algorithm with multiple starting point is obtained. After n iterations the tabu list of all the ants will be full then for each ant L_k (i.e the distance traveled by k^{th} ant) is calculated and the value of minimum L is stored rest all values are deallocated and then depending upon the scheme of pheromone update the level of pheromone are updated. This process is iterated NC_{MAX} times or all the ants use the same tour (this phenomena is known as *stagnation behaviour)* where NC_{MAX} is the user defined parameter and then L = min L_k (over NC_{MAX} iterations) is calculated, which in turn is the solution given by the algorithm.

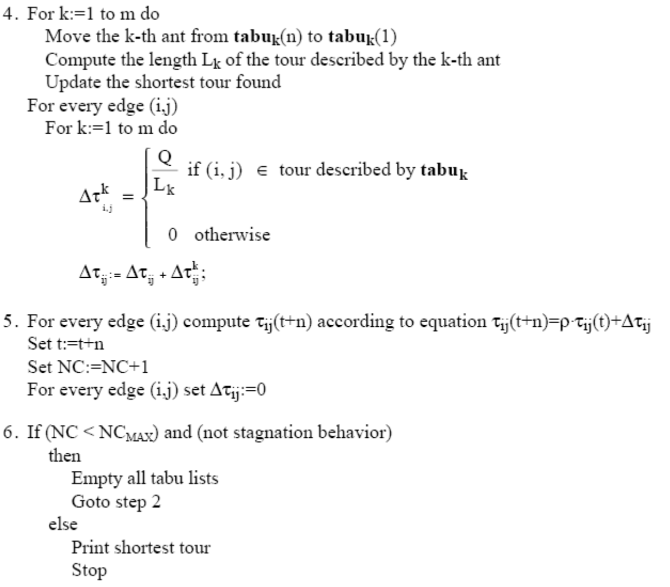
Below a formal *ant-cycle algorithm* or simple ant colony optimization algorithm is given:



```
1. Initialize:
      Set t:=0                              {t is the time counter}
      Set NC:=0                             {NC is the cycles counter}
      For every edge (i,j) set an initial value $\tau_{ij}(t)=c$ for trail intensity and $\Delta\tau_{ij}= 0$
      Place the m ants on the n nodes

2. Set s:=1                                 {s is the tabu list index}
   For k:=1 to m do
      Place the starting town of the k-th ant in $\mathbf{tabu}_k(s)$

3. Repeat until tabu list is full           {this step will be repeated (n-1) times}
      Set s:=s+1
      For k:=1 to m do
         Choose the town j to move to, with probability $p_{ij}^k(t)$
                                  {at time t the k-th ant is on town i=$\mathbf{tabu}_k(s-1)$}
         Move the k-th ant to the town j
         Insert town j in $\mathbf{tabu}_k(s)$
```



```
4. For k:=1 to m do
      Move the k-th ant from $\mathbf{tabu}_k(n)$ to $\mathbf{tabu}_k(1)$
      Compute the length $L_k$ of the tour described by the k-th ant
      Update the shortest tour found
   For every edge (i,j)
      For k:=1 to m do
         $\Delta\tau_{i,j}^k = \begin{cases} \frac{Q}{L_k} & \text{if } (i,j) \in \text{tour described by } \mathbf{tabu}_k \\ 0 & \text{otherwise} \end{cases}$
         $\Delta\tau_{ij} := \Delta\tau_{ij} + \Delta\tau_{ij}^k;$

5. For every edge (i,j) compute $\tau_{ij}(t+n)$ according to equation $\tau_{ij}(t+n)=\rho\cdot\tau_{ij}(t)+\Delta\tau_{ij}$
   Set t:=t+n
   Set NC:=NC+1
   For every edge (i,j) set $\Delta\tau_{ij}:=0$

6. If ($NC < NC_{MAX}$) and (not stagnation behavior)
      then
         Empty all tabu lists
         Goto step 2
      else
         Print shortest tour
         Stop
```

IV. CONCLUSION

Today, several hundred papers have been written on the applications of ACO. It is a true metaheuristic, with dozens of application areas. While both the performance of ACO algorithms and theoretical understandings of their working have significantly increased. There are several areas in which until now only preliminary steps have been taken and where much more research will have to be done. One of these research areas in the extension of ACO algorithms to more complex optimization problems that include (1) dynamic problems, in which the instance data, such as objective function values, decision parameters, or constraints, may change while solving the problem; (2) stochastic problems, in which one has only probabilistic information about objective function values, decision variables values, or constraint boundaries, due to uncertainty, noise, approximation, or other factors; and (3) multiple objective problems, in which a multiple objective function evaluates competing criteria of solution quality. Active research directions in ACO include also the effective parallelization of ACO algorithms and, on a more theoretical level, the understanding and characterization of the behavior of the ACO algorithms while solving a problem.

V. REFERENCES

[1] M. Dorigo, V. Maniezzo, A. Colorni (1991) The ant system: an autocatalytic optimizing process, Technical Report TR91-016, Politecnico di Milano

[2] A. Colorni, M. Dorigo, V. Maniezzo (1991) Distributed optimization by ant colonies, In Proceedings of ECAL'91 European Conference on Artificial Life, Elsevier Publishing, Amsterdam, The Netherlands, pp 134-142

[3] M. Dorigo in "New ideas in Optimization" , McGrawHill [1999] , section1 , pp 11-32

[4] Marco Dorigo , Eric Bonabeaub, Guy Theraulaz (2000) in "Ant Algorithms and Stigmergy" in Future Generation Computer Systems 851–871

[5] Vittorio Maniezzo, Luca Maria Gambardella, Fabio de Luigi, in "Ant colony optimization"

[6] M. Dorigo, Luca M. Gambardella, Gianni Di Caro (1999) in "Ant Algorithms for Discrete Optimization" published in Artificial Life, Vol.5, No.3, pp. 137-172, 1999.

[7] J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels. The self-organizing exploratory pattern of the argentine ant. Journal of Insect Behavior, 3:159–168, 1990.

[8] Marco Dorigo and Luca Maria Gambardella, in "Ant colonies for the travelling salesman problem" article in Biosystems, Volume 43, Issue 2, July 1997, Pages 73-81

[9] Marco Dorigo, Vittorio Maniezzo, Alberto Colorni in "The Ant System: Optimization by a colony of cooperative agents" in IEEE Transactions on Systems, Man, and Cybernetics–Part B, Vol.26, No.1, 1996, pp.1-13

[10] www.coro.caltech.edu/Courses/ EE141/Lecture/W8/RG_EE141_W8ACO.pdf