Ant Colony Optimization

Saad Ghaleb Yaseen **Nada M. A.AL-Slamy**
Department of Management Information Systems
College of Economics & Business
Al-Zaytoonah University of Jordan

Abstract
Ant Colony Optimization (ACO) is model for bio-simulation due to their relative individual simplicity and their complex group behaviors. This paper introduces ACO as a distributed algorithm that is applied to solve Traveling Salesman Problem (TSP). Our case study is Jordan's Seaport Motion, ACO principles are applied to find best rout for goods propagation from Al-Aqaba Seaport to inside Amman.

1 . Introduction

Ant Colony Optimization (ACO) is a paradigm for designing meta heuristic algorithms for combinatorial optimization problems. A Meta heuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. In other words, a meta heuristic is a general-purpose algorithmic framework that can be applied to different optimization problems with relatively few modifications[1]. Examples of meta heuristics include simulated annealing [2], tabu search [3], iterated local search [4], evolutionary computation [5], and ant colony optimization.

Meta heuristic algorithms are algorithms which, in order to escape from local optima, drive some basic heuristic: either a constructive heuristic starting from a null solution and adding elements to build a good complete one, or a local search heuristic starting from a complete solution and iteratively modifying some of its elements in order to achieve a better one. The meta heuristic part permits the low-level heuristic to obtain solutions better than those it could have achieved alone, even if iterated. Usually, the controlling mechanism is achieved either by constraining or by randomizing the set of local neighbor solutions to consider in local search [6].

The first algorithm which can be classified within this framework was presented in 1991 by Marco Dorigo with his PHD thesis "Optimization, learning, and Natural Algorithms", modeling the way real ants solve problems using pheromones, and, since then, many diverse variants of the basic principle have been reported in the literature, (as shown in table 1). Real ants are capable of finding the shortest path from a food source to their nest. While walking ants deposit pheromone on the ground and follow pheromone previously deposited by other ants, the essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions [1]. In ACO, a number of artificial ants build solutions to an optimization problem and exchange information on their quality via a communication scheme that is reminiscent of the one adopted by real ants.

To find a shortest path, a moving ants lay some pheromone on the ground, so an ant encountering a previously trail can detect it and decide with high probability to follow it. As a result, the collective behavior that emerges is a form of a positive feedback loop where the probability with which an ant choose a path increases with the number of ants that previously chose the same path [7].

Table 1 a non-exhaustive list of successful ant colony optimization algorithms (in chronological order).

Algorithm	Authors	Year	References
Ant System	Dorigo Et Al	1991	[6]-[8]
Elitist As	Dorigo Et Al	1992	[7],[8]
Ant –Q	Cambardella & Dorigo	1995	[9]
Ant Colony System	Dorigo & Cambardella	1996	[10]-[12]
Max-Min As	Stutzle & Hoos	1996	[13]-[15]
Rank –Based As	Bullnheimer Et Al	1997	[16],[17]
Ants	Maniezzo	1999	[18]
Bwas	Cordon Et Al	2000	[19]
Hyper-Cube As	Blum Et Al	2001	[20],[21]

Ant colony optimization is an iterative distributed algorithm. At each iteration, a set of artificial ants (cooperating agents) are considered. Each of them builds a solution by walking from vertex to vertex on the graph with the constraint of not visiting any vertex that it has already visited in her walk. At each step of the solution construction, an ant selects the following vertex to be visited according to

Manuscript received June 5, 2008.
Manuscript revised June 20, 2008.

a stochastic mechanism that is biased by the pheromone: when in vertex i, the following vertex is selected stochastically among the previously unvisited ones. In particular, if j has not been previously visited, it can be selected with a probability that is proportional to the pheromone associated with edge (i, j). At the end of an iteration, on the basis of the quality of the solutions constructed by the ants, the pheromone values are modified in order to bias ants in future iterations to construct solutions similar to the best ones previously constructed. [8][9].

2. A combinatorial Optimization Problem

The ACO system contains two rules:

1. Local pheromone update rule, which applied whilst constructing solutions.
2. Global pheromone updating rule, which applied after all ants construct a solution

Furthermore, an ACO algorithm includes two more mechanisms: trail evaporation and, optionally, daemon actions. Trail evaporation decreases all trail values over time, in order to avoid unlimited accumulation of trails over some comp onent. Daemon actions can be used to implement centralized actions which cannot be performed by single ants, such as the invocation of a local optimization procedure, or the update of global information to be used to decide whether to bias the search process from a non-local perspective [4][6][8]

The ACO meta-heuristic can be applied to discrete optimization problems characterized as follows.

- **$C = \{c1; c2; : : : ; c_{NC}\}$ is a finite set of *components*.**

  $L = \{ l_{cicj} \mid (c_i; c_j) \in \tilde{C}\}; |L| \leq N_C^2$ is a finite set of possible *connections/transitions* among the elements of C, where $\tilde{C}$ is a subset of the Cartesian product C x C.
- **$J_{cicj} \equiv J(l_{cicj}\ ; t)$ is a *connection cost* function associated to each $l_{cicj} \in L$, possibly parameterized by some time measure t.**
- **$\Omega \equiv \Omega(C; L; t)$ is a finite set of *constraints* assigned over the elements of C and L.**
- **$s = <c_i, c_j, \ldots, c_k, \ldots>$ is a sequence over the elements of C (or, equivalently, of L). A sequence s is also called a *state* of the problem. If S is the set of all possible sequences, the set $\tilde{S}$ of all the (sub) sequences that are feasible with respect to the constraints $\Omega(C; L; t)$, is a subset of S. The elements in ~ S define the problem's *feasible states*. The length of a sequence s, that is, the number of components in the sequence, is expressed by |s|.**
- **Given two states s_1 and s_2 a *neighborhood structure* is defined as follows: the state s_2 is said to be a neighbor of s_1 if both s_1 and s_2 are in S, and the state s_2 can be reached from s1 in one logical step (that is, if c_1 is the last component in the sequence determining the state s1, it must exists c2 2 C such that $l_{c1c2} \in L$ and $s_2 \equiv <s_1, c_2>$). The neighborhood of a state s is denoted by N_s.**
- **Ψ is a *solution* if it is an element of $\tilde{S}$ and satisfies all the problem's requirements. A *multi-dimensional* solution is a solution defined in terms of multiple distinct sequences over the elements of C.**
- **$J_\Psi (L, t)$ is a *cost* associated to each solution Ψ. $J_\Psi(L, t)$ is a function of all the costs J_{cicj} of all the connections belonging to the solution Ψ.**

Figure 2 Meta heuristic ACO characteristics

Although each ant of the colony is complex enough to find a feasible solution to the problem under consideration, good quality solutions can only emerge as the result of the collective interaction among the ants. Each ant makes use only of private information and of information local to the node it is visiting.

Ants of the colony have the following properties:

- **An ant searches for minimum cost feasible solutions $\hat{J}_{\Psi} = \min_{\Psi} \hat{J}_{\Psi}(L, t)$.**
- **An ant k has a memory M^k that it can use to store information on the path it followed so far. Memory can be used to build feasible solutions, to evaluate the solution found, and to retrace the path backward.**
- **An ant k in state $s_r = <s_{r-1}, i>$ can move to any node j in its *feasible neighborhood* N_k^i ,defined as $N_k^i = \{j | (j \in N_i) \wedge (<s_r, j> \in \tilde{S})\}$.**
- **An ant k can be assigned a *start state* s_s^k and one or more *termination conditions* e^k. Usually, the start state is expressed as a unit length sequence, that is, a single component.**
- **Ants start from the start state and move to feasible neighbor states, building the solution in an incremental way. The construction procedure stops when for at least one ant k at least one of the termination conditions e^k is satisfied.**
- **An ant k located on node i can move to a node j chosen in N_i^k . The move is selected applying a probabilistic decision rule.**
- **The ants' probabilistic decision rule is a function of (i) the values stored in a node local data structure $A_i = [a_{ij}]$ called *ant-routing table*, obtained by a functional composition of node locally available pheromone trails and heuristic values, (ii) the ant's private memory storing**
- **its past history, and (iii) the problem constraints.**
- **When moving from node i to neighbor node j the ant can update the pheromone trail τ_{ij} on the arc (i,j).This is called *online step-by-step pheromone update*.**
- **Once built a solution, the ant can retrace the same path backward and update the pheromone trails on the traverse arcs. This is called *online delayed pheromone update*.**

Figure 4 ACO Properties

3. ACO for the Traveling Salesman Problem

The first algorithm that applies an ACO based algorithm to a more general version of the ATSP problem is Hybrid Ant System for the Sequential Ordering Problem (HAS-SOP[ ]).HAS-SOP was intended to solve the sequential ordering problem with precedence constraints (SOP). The SOP in an NP-hard combinational optimization problem first formulated by Escudero [6]to design heuristics for a production planning system. In the traveling salesman problem, a set of cities is given and distance between each of them is known. The goal is to find the shortest tour that allows each city to be visited once and only once. The TSP is the problem of finding a minimal length Hamiltonian circuit on the graph G=(C; L). A Hamiltonian circuit of graph G is a closed tour visiting once and only once all the NC nodes of G. Its length is given by the sum of the lengths of all the arcs of which it is composed. A variable called pheromone is associated with each edge and can be read and modified by ants [10] [11]

TSP Ant Systems:

A number m of ants is positioned in parallel on m cities. The ants' start state, that is, the start city, can be chosen randomly, and the memory Mk of each ant k is initialized by adding the current start city to the set of already visited cities (initially empty). Ants then enter a cycle, which lasts NC iterations, that is, until each ant has completed a tour. During each step an ant located on node i consider the feasible neighborhood, reads the entries a_{ij}'s of the ant-routing table Ai of node I, computes the transition probabilities, moves to the new city, and updates its memory. Once ants have completed a tour, they use their memory to evaluate the built solution and to retrace the same tour backward and increase the intensity of the pheromone trails τ_{ij} of visited connections l_{ij}. This has the effect of making the visited connections become more desirable for future ants. Then the ants die, freeing all the allocated resources. [12][13].

The pheromone trail information is changed during problem solution to reflect the experience acquired by ants during problem solving. Ants deposit an amount of pheromone proportional to the quality of the solutions they produced: the shorter tour generated by an ant, the greater amount of pheromone it deposits on the arcs which it used to generate the tour. This choice helps to direct search towards good solutions.

Ant's memory allow it to compute the length of the tour generated and to cover the same path backward to deposit pheromone on the visited arcs. The ant-routing table Ai =[$a_{ij}(t)$] of node i, Ni is the set of all the neighbor nodes of node i], is obtained by the following functional composition of pheromone trails $\tau_{ij}(t)$ and local heuristic values η_{ij}[5] :

$$a_{ij} = \frac{[t_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in A_i}[t_{il}]^{\alpha}[\eta_{il}]^{\beta}} \forall_j \in N_i$$

Where α and β are two parameters that control the relative weight of pheromone trail and heuristic value. The heuristic values used $\eta_{ij}=1/J_{cicj}$, where J_{cicj} is the distance between cities I and j. in other words, the shorter distance between two cities I and j, the higher heuristic value η_{ij}. The probability P_{ij}^{k} (t) with which at the t-th algorithm iteration an ant k located in city I chooses the city j Є N_i^k to move to is given by the following probabilistic decision rule[10]:

$$P_{ij}^{k}(t) = \frac{\alpha_{ij}(t)}{\sum_{l \in A_i}[\alpha_{il}}$$

Where N_i^k Є N_i is the feasible neighborhood of node i for k (that is, the set of cities ant k has not yet visited) as defined by using the ant private memory Mk and the problem constraints. If α =0, the closet cities are more likely to be selected: this corresponds to a classical stochastic greedy algorithm (with multiple starting points since ants are initially randomly distributed on the nodes). If β =0, only pheromone amplification is at work: this method will lead to the rapid emergence of stagnation, that is, a situation in which all ants make the same tour which, in general, is strongly sub-optimal. An appropriate trade-off has to be set between heuristic value and trail intensity.

After all ants have completed their tour, each ant k deposits a quantity pheromone $\Delta\tau^{k}(t)=1/J_{\psi}^{k}(t)$ on each connection l_{ij} that it has used, where $J_{\psi}^{k}(t)$ is the length of tour $\Psi^{k}(t)$ done by ant k at iteration:

$$\tau_{ij}(t) \leftarrow \tau_{ij}(t) + \Delta\tau^{k}(t),$$
$$\forall l_{ij} \in \psi^{k}(t), = 1,\ldots\ldots, m$$

After pheromone updating has been performed by the ants, pheromone evaporation is triggered; the following rule is applied to all the arcs L_{ij} of the graph G [5]:

$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t)$$

Where ρ Є (0,1] is the pheromone trail decay coefficient [10].

4. Case Study

We take Amman Seaport Motion as a case study for TSP problem. ACO algorithm is applied to find sufficient solution to the problem of propagation of goods from aqaba seaport to Amman, where the merchant reside (those are the importance of the goods). In general, Amman is subdivided into four parts: north, west, east, and south area. At the same time each area is also subdivided into a set of sub-area, which are the final destinations of the goods. Because Aqaba seaport is one of the more important seaport, all competitive in shipment cost are done in an open market that depend on demonstration and demanding constrains.

The set C , composed from these finite set of area (for example:Amman - West include Na'aur, Sualeh, …). Thus, the set L is a finite set of the available paths among the element of C. figure 1 specify an example of our problem. The connection cost function $\mathbf{J_{cicj}}$ associated to each path in L, as defined in section 2 , is represented by Ton Propagating Cost (TPC) function. TPC function depend on four factors, which are:

a. Distance measured in kilometer .
b. Payload Volume measured in kilogram.
c. Backward Payload, its value 1 or 0 .
d. Competitiveness, positive real value obtained by experimental work.

Since the above factors did not have the same priorities, they must multiplied by some parameters. The value of such parameters can be set by experience work on the ACO algorithm, or in other word by try and error approach. We set to payload volume factor more priority than the other factors.

Furthermore, a finite set of *constraints* assigned over the elements of C and L

Ω ≡ Ω(C; L; t) given in section 2, may include :

a. Destinations.
b. Goods type, since some of them are sensitive or may be destroyed over the time for example .
c. Demonstration and Demanding constrains

Table 2 specify ships movement that arrived in Aqaba seaport in 2007 year. From the table its clear that the process of goods propagation is not systematic. There is no clear procedure to define and compute ton driving cost.

Accordingly, the injection of ACO algorithm(as described in the previous sections) in goods propagation process, help to find a systematic, effective procedure to find good (but may be not the optimum) paths for goods propagation with respect to the TPC function and constrains set listed above.

Figure 1, specify an example of possible construction graphs for a Amman five-area with respect to Aqaba seaport, where components are associated with (a) the edges or with (b) the vertices of the graph.

Date	Weight in Ton	Payload	Destination	TSP in JD
3/1/2007	2052	Rice	Aqabq	2.55
4/1/2007	52876	Corn	South-Amman Al-jeza	9
8/1/2007	4537	Sugar	North-Amman Marka	9
28/1/2007	1482	Oils	Weast-Amman Al-mopher	9
29/3/2007	555	Rice	Center-Amman	8.5
4/5/2007	48460	Corn	North-Amman	8
15/7/2007	4680	Tree stem	Aqaba	7.5
21/8/2007	15048	Sugar	Middel-Amman	7.5
…..	……	…..	…..	…

Table 2: ships movement that arrived in Aqaba seaport in 2007 year.

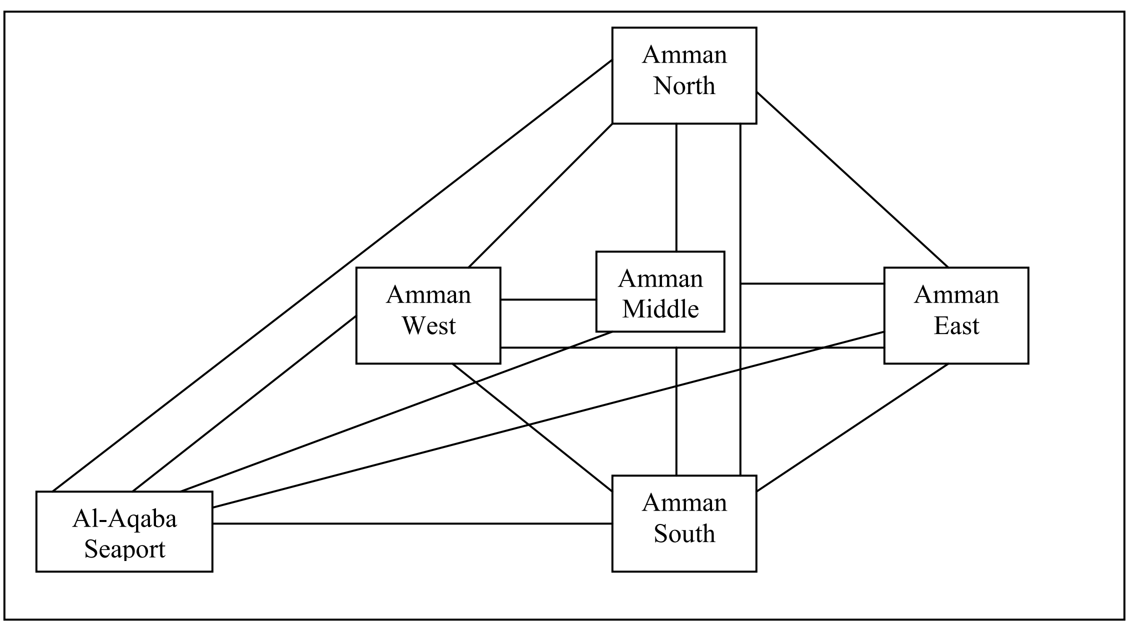

Figure 1: Possible construction graphs for Amman five area and Al-Aqaba seaport.

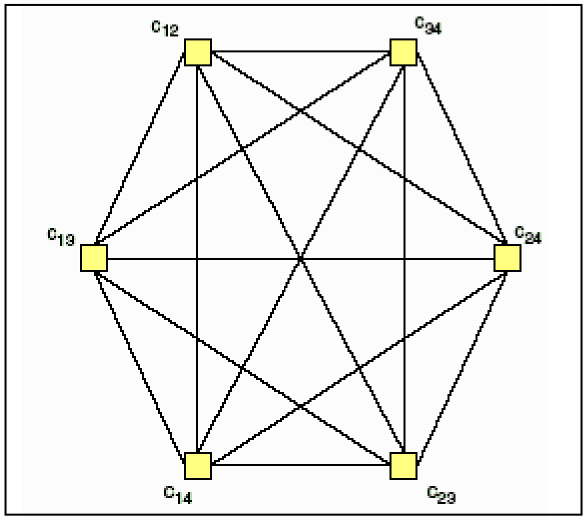

Figure 2: Example of possible construction graphs for each area inside Amman

Ant Algorithm:

Repeat a and b until either accepted minimum cost feasible solution is optained, or the iteration limit is exceeded:

a .Construct Ant Solutions

At each iteration construct feasible state for each ant, then construct a solution based on connection cost and constrain functions as defined above. For each solution we compute the cost associated with it.

At each step , each ant computes a set of feasible expansions to its current state, and moves to one of these in probability. As specified in the previous section, for ant k, the probability of moving from state t to state ņ depends on the combination of two values:

- the attractiveness of the move, as computed by connection cost function $\mathbf{J_{cicj}}$ and the finite set of constraints $\mathbf{\Omega(C; L; t)}$.

- the trail level of the move, indicating how proficient it has been in the past to make that particular move: it represents therefore an a posteriori indication of the desirability of that move.

- When ant-routing table is obtained, if α =0, the closet cities are more likely to be selected: this corresponds to a classical stochastic greedy algorithm (with multiple starting points since ants are initially randomly distributed on the nodes). If β =0, only pheromone amplification is at work: this method will lead to the rapid emergence of stagnation, that is, a situation in which all ants make the same tour which, in general, is strongly sub-optimal. An appropriate trade-off has to be set between heuristic value and trail intensity. By experiment work on this problem we set ($\beta > \alpha$) for example $\beta = 0.7$ and $\alpha = 0.5$.

b. Update Pheromones

It is used to increase the pheromone values associated with good or promising solutions, and decrease those that are associated with bad ones.

- Decreasing all the pheromone values through *pheromone evaporation* -> allows "forgetting"-> favors exploration of new areas

- Increasing the pheromone levels associated with a **chosen set** of good solutions -> makes the algorithm converge to a solution

- ρ (0; 1] is a parameter called evaporation rate (the initial amount of pheromone τ_{ij} is set to a small positive constant value on all arcs). We set $\rho = 0.0001$ on all arcs.

5. Conclusion

1. ACO is a class of algorithms, whose first member, called Ant System, was initially proposed by Colorni, Dorigo and Maniezzo. The main underlying idea, loosely inspired by the behavior of real ants, is that of a parallel search over several constructive computational threads based on local problem data and on a dynamic memory structure

containing information on the quality of previously obtained result.

2. The functioning of an ACO algorithm can be summarized as follows: a set of computational concurrent and asynchronous agents (a colony of ants) moves through states of the problem corresponding to partial solutions of the problem to solve. They move by applying a stochastic local decision policy based on two parameters, called trails and attractiveness. By moving, each ant incrementally constructs a solution to the problem. When an ant completes, or during the construction phase, the ant evalutes the solution and modifies the trail value on the components used in its solution. This pheromone information will direct the search of the future ants.

3. In ACO, an artificial ant builds a solution by traversing the fully connected construction graph G (C, L), where C is a set of vertices and L is a set of edges. This graph can be obtained from the set of solution components C in two ways: components may be represented either by vertices or by edges.

4. Artificial ants move from vertex to vertex along the edges of the graph, incrementally building a partial solution. Additionally, ants deposit a certain amount of pheromone on the components; that is, either on the vertices or on the edges that they traverse. The amount of pheromone deposited may depend on the quality of the solution found. Subsequent ants use the pheromone information as a guide toward promising regions of the search space.

5. Ants adaptively modify the way the problem is represented and perceived by other ants, but they are not adaptive themselves.

6. The collective behavior emerging from the interaction of the different search threads has proved effective in solving combinational optimization(CO) problems.

7. Use ACO concepts for goods propagation process, help to find a systematic, effective procedure to find good (but may be not the optimum) paths for goods propagation with respect to some predefined cost and constrains functions.

References

[1] M. Dorigo, luca M. G., " Ant Colony system: A Cooperative learning approach to the Travelling Salesman Problem", IEEE transaction on evolutionary computation, Vol. 1, No. 1, 1997.

[2] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi, "Optimization by Simulated Annealing", Science, vol. 220, pp. 671-680, 1983.

[3] F. Glover and M. Laguna, "Tabu Search", Kluwer Academic Publishers, 1997.

[4] H.R. Lourenco, O. Martin, and T. Stutzle, "Iterated Local Search", in Handbook of Metaheuristics, ser. International Series in Operations Research & Management Science, F. Glover and G.Kochenberger, Eds., Kluwer Academic Publishers, vol. 57, pp. 321-353, 2002.

[5] J. Holland, "Adaptation in Natural and Artificial Systems", Ann Arbor: University of Michigan Press, 1975.

[6] A. Colorni, M. Dorigo and V. Maniezzo," Distributed optimization by ant colonies", Proceedings of ECAL 91, European Conference on Artificial Life, Elsevier Publishing, Amsterdam,1991.

[7] M. Dorigo and G. Di Caro, "The Ant Colony Optimization meta-heuristic", in New Ideas in Optimization, D. Corne et al., Eds., McGraw Hill, London, UK, pp. 11-32, 1999.

[8] M. Dorigo, G. Di Caro, and L.M. Gambardella, "Ant algorithm for discrete optimization", Artificial Life, vol. 5, no. 2, pp. 137-172, 1999.

[9] M. Dorigo, M. Birattari, and T. Stitzle, "Ant Colony Optimization: Arificial Ants as a Computational Intelligence Technique, IEEE computational intelligence magazine, November, 2006.

[10] M. Dorigo, Vittorio, Maniezzo and Alberto Colorni, "The Ant system:Optimization by A Colony of Cooperating Agents", IEEE Transaction on system, man, and cybernetics, part B, Vol. 26, No. 1, 1996.

[11] Gilmour Stephen and Mark Dras, "Understanding the pheromone system within Ant Colony Optimization", available at http://www.ics.mq.ed.au

[12] S. Lin. "Computer Solutions of the Traveling Salesman Problem", Bell systems Journal, 44, 2245-2269, 1965.

[13] Christopher M. White, Gary G. Yen, "A Hybrid Evolutionary Algorithm for Traveling Salesman Problem", school of Electrical and Computer Engineering, Oklahoma State University, Stillwater, ok 74078, USA.

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

[15] L. Bianchi, L.M. Gambardella, M. Dorigo, "An Ant Colony Optimization Approach to the probabilistic Traveling Salesman Proble", In Proceedings of PPSN-VII, Seventh International Conference on Parallel Problem Solving from Nature, Lecture Notes in Computer Science. Springer Verlag, Berlin, Germany, 2002.

[16] P. Jungie, W. Dingwei, "An Ant Colony Optimization Algorithm for Multiple Treavelling Salesman Problem", ICICIC apos; 06. first international conference on Volume 1, Issue, 30-01 Aug. 2006, page(s) : 210-213.

[17] M. Dirigo and Thomas Stutzle, "Ant Colony Optimization", MIT Press Cambridge, 2004.

Saad Ghaleb Yaseen has a PhD in Management Information Systems. He is Associate Professor and Head of the MIS Department in the Faculty of Economics and Administrative Sciences at Al-Zaytoonah University of Jordan. He had conducted over 40 specialized studies in many field such as IT, IS E-management, and Knowledge management. He is a renowned expert in the management of IT projects and a professional academician in the Middle East.

Nada M. A. Al-Slamy (1971) , PhD in Computer Science / Theory of computer Science and Artificial Intelligence. She is Assistant Professor at Management of Information System Department in the Faculty of Economics and Administrative Sciences at Al-Zaytoonah University of Jordan. She Interests in. fields : Computational theory, Artificial Intelligence, Information Security, Data Mining and Data Retrieval.