# Solving Traveling Salesman Problem with Probabilistic Elitist Ant Colony Optimization

Amrita Chatterjee, Eunjin Kim, Hassan Reza
Department of Computer Science
University of North Dakota
Grand Forks, ND, 58202-9015, USA
amrita.chatterjee@und.edu, ejkim@cs.und.edu, reza@cs.und.edu

## Abstract

Ant Colony Optimization (ACO) is a population-based metaheuristic algorithm for optimization problem, inspired by foraging behavior of ants in ant colony. One of its variants, the elitist ACO, further reinforces itself with the additional pheromone deposit to find the best path. Even though this usually leads to converging on the solution faster, it also has the drawback of getting stuck in local minima. In this paper, we describe a variation on elitist ACO where the pheromone contribution of best path is further predicated by a probability factor. This probabilistic elitist ACO often produces better solution for TSP, at the cost of higher number of iterations. Some experimental results of this probabilistic elitist ACO is presented

# 1 Introduction

Ant Colony Optimization (ACO) is a population-based metaheuristic algorithm for an optimization problem, inspired by foraging behavior of ants in ant colony [9]. ACO is expected to find the near optimal solutions like various nature-inspired metaheuristic algorithm or evolutionary algorithm such as particle swarm optimization algorithm (PSO), artificial bee colony algorithm, differential algorithm (DE), etc.

Ant algorithms are based on the behavior of ants in ant colonies. The ants are capable of finding the shortest route from their nest to a food source and coming back to the original place [2]. Initially they start their travel randomly at the same speed with no clue on how to find the shortest path. The shorter the path is, the more pheromones get accumulated; hence, more ants follow the route and form a shorter closed tour. Pheromone is the chemical substance excreted by the ants and it works as a mechanism of stigmergy among ants [9]. It attracts other ants in search for food. The attractiveness of a given path depends on the intensity of pheromones that the ants have deposited during their travel. The motion of ants is stochastic. Hence initially the ants choose any routes randomly, but they get to choose preferable, i.e. shorter path for their food collection and return. Since the pheromone has the characteristics of evaporation, the paths which are less likely traversed by ants have less quantity of pheromones. As the longer path has also more evaporation of deposited pheromone, the quantity of pheromone on each edge of path is indirectly proportional to the edge of the path. Therefore, the ants find the shortest path in their indirect communication via the amount of deposited pheromone on their path as the pheromone is a mechanism of the sign-based stigmergy among the ants. This behavior is simulated with virtual ants in the ACO algorithms. In the original version of the ACO, many iterations are needed to find the optimal solution [2]. On the other hand, in the elitist version of ACO, converges to local minima sooner, however may skip searching for more optimal path [2].

A hybrid version of the algorithm with probability $p$ of choosing original version of ACO and the elitist version of ACO behavior is discussed in this paper. Two new versions of the algorithms are developed: Probabilistic Elitist ACO (fixed value of $p$) and Dynamic Probabilistic ACO. This hybrid approach does not converge to local minima earlier but still finds for more optimal solution, and have less iterations as compared to the original ACO.

# 2 Background

## 2.1 Travelling Salesman Problem

Traveling Salesman Problem (TSP) is a combinatorial optimization problem to find the shortest path. TSP is a NP-hard problem where there exists no algorithm to solve it in polynomial time; the optimal solution may be obtained in exponential amount of time. In the map of a given number of cities, the salesman travels every city exactly once and must return to the original city making a whole closed tour. Thus, in the graph model of

TSP, the starting node and the ending node are identical. In this paper. There are several practical applications of TSP such as planning bus lines, regular distribution of resources, finding of customer service route, etc. [3]. There are other application areas as well which are not related to travelling routes or non-path finding problems like chain diagram optimization, application in crystallography, industrial robot control, computer motherboard components layout, drilling holes for electric circuit. [3]

## 2.2 Ant System

Ant System was the first proposed ACO algorithm [15]. It introduced a model of pheromone deposition.

$$\tau_{ij} = (1- \rho) \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta{\tau_{ij}}^{k}$$

where $\tau_{ij}$ is the pheromone deposit in the edge (*i, j*), ρ is the evaporation rate and $\Delta\tau^{k}_{ij}$ is the pheromone update on the edge (*i, j*) by each ant *k* in the colony of *m* ants.

## 2.3 Ant Colony System

Ant Colony System is a distributed algorithm where ants play the role of cooperating agents in order to find the optimal solution. The ACS is built on top of Ant System in order to improve the efficiency when it is applied to both symmetric and asymmetric TSPs. The communication among the ants is made via the pheromone. Pheromones are deposited by the ants on the edges of the rote they travel. In ACS the ants use a pseudorandom proportional rule. The probability of an ant to move to a particular city from the current city is determined by the parameter $q_0$ and random variable q. ACS also introduces Local Pheromone Update.

## 2.4 Original Ant Colony Optimization Algorithm for TSP

Ant System was first applied to TSP. It utilizes a graph representation where each of the edges has cost measure, desirability measure. The pheromone is updated at every run time by the ants. For symmetric instances of the TSP, the desirability measure is $\tau(r,s) = \tau(s,r)$ and for asymmetric measure its $\tau(r,s) \neq \tau(s,r)$. Ants completes a closed tour by the probabilistic state transition rule. This is also known as random-proportional rule. In this rule it helps the ants to take decision on which city to travel from a current city and then the next and so on in order to finish one full tour. The transition probability to choose the next city *s* from *r* is:

$$P_k(r,s) = \begin{cases} \dfrac{[\tau(r,s)]\cdot[\eta(r,s)]^{\beta}}{\sum_{u\epsilon J_{k(r)}}[\tau(r,u)]\cdot[\eta(r,u)]^{\beta}}, & if\ s\epsilon J_k(r) \\ 0, & otherwise \end{cases}$$

where $\tau(r, s)$ is the pheromone amount in $(r, s)$ which implies a posteriori effectiveness of the move from $r$ to $s$, $\eta(r, s)$ is the inverse of distance which implies a priori effectiveness of the move from $r$ to $s$ with its amplifying parameter $\beta$. $J_k(r)$ is the set of neighborhood nodes that are to be visited by ant $k$ positioned on city $r$. This formula of transition rule shows the probability of ant $k$ to travel from city $r$ to $s$ if s belongs to the set of cities otherwise its '0'.

Global Updating Rule is applied once all the ants have completed their tour. Pheromones get updates eventually. Since the pheromones have the nature of evaporating thus the paths which do not gets refreshed or not desirable by the ants will have less pheromones. This rule was intended so that the total pheromone updated is equivalent to the remaining pheromone (after evaporation) added to the refreshed pheromone. This rule is applied at the end. Only globally best ant can construct the route (which deposited more pheromone) Thus, the intensity of pheromones is proportional to the shorter paths. The pheromone update is done by:

$$\tau(r,s) \longleftarrow (1-\alpha)\cdot\tau(r,s) + \sum_{k=1}^{m}\Delta\tau_k(r,s)$$

where

$$\Delta\tau_k\,(r,s) = \begin{cases} \dfrac{1}{L_k} & if\ (r,s)\epsilon\ tour\ done\ by\ ant\ k, \\ 0 & otherwise. \end{cases}$$

$L_k$ is the length of the tour performed by ant k, m is the number of ants. This pheromone updating rule is done in order to give higher reinforcement to better solutions. The pheromone updating formula is the combination of both the residue pheromone after it got evaporated and the refreshed pheromone. The pheromones which are located on the edges of the graph act as distributed long-term memory which is not present within the ants. The indirect communicated which is formed by this is termed as stigmergy [3]. The ant algorithm was found advantageous in finding optimal solution when dealt with smaller number of cities but didn't prove successful for larger number of cities.

Ant Colony System introduced the three main rules: i) state transition rule ii) global updating rule and iii) local pheromone updating rule. A certain number of ants are initialized in a certain number of city -graph. Each of the ants start for their tour by the use of both heuristic greedy search and by information obtained from the previous ants. The ants update the pheromone on the paths they are traversing by the local pheromone updating rule. Once all the tours are completed by the ants the pheromones get updated globally while applying the global updating rule. Thus, the edges with more pheromones are visited by more ants. This generates the shortest path.

The ant chooses to go to the next city is by applying the following rule:

$$s = \begin{cases} argmax_{u \epsilon J_k(r)} \{[\tau(r,u)] \cdot [\eta^{\beta}(r,u)]\} \\ \quad if\ q \leq q_0\ (exploitation) \\ S \quad otherwise\ (biased\ exploration) \end{cases}$$

Where $J_k$ is the neighborhood of the city where an ant $k$ is currently located.

The state transition rule favors the tendency of the ants to move to the next city which contains more pheromone and thus the shortest path.

**Local Updating Rule:** The edges of the graph get updated each time by the local update pheromone rule. An ant finishes each iteration

$$\tau\ (r,s) \rightarrow (1-\rho) \cdot \tau\ (r,s) + \rho \cdot \Delta\ \tau\ (r,s).$$

The pheromone level is changed by the above expression.

**Global Updating Rule:** The globally best ant which constructed the shortest route is allowed to deposit pheromone. This is performed at the end after all the ants have completed their tours. The following rule is applied while the pheromone is updated:

$$\tau(r,s) \longleftarrow (1-\alpha) \cdot \tau(r,s) + \alpha \cdot \Delta\tau(r,s)$$

where $\alpha$ the pheromone decay parameter. The pheromone update on the edge $(r, s)$, $\Delta\tau(r,s)$ is:

$$\Delta\tau(r,s) = \sum_k \Delta_{rs}^k$$

where

$$\Delta_{rs}^{k} = \begin{cases} \frac{Q}{L_k} if\ ant\ k\ uses\ edge\ (r,s)\ in\ its\ tour. \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad otherwise. \end{cases}$$

# 3 Related Work

In [1], Dorigo et.al proposed a new model of Ant Colony Optimization to solve Travelling Salesman problem. Ants have the capability to remember the previous paths they have traversed and thus it will facilitate to find their best-so-far solution in less time. They have shown ants with memory can find better solution in less time. They introduced the idea of memory of earlier solutions in order to make use of the previous best solution which was constructed by the ants who traversed the route before. Here they have proposed the algorithm having ants with memory named as Mant ACO. It was found that Mant algorithm found optimum value faster in small sized problem. Mant is simple and interesting approach.

In [3] exact algorithms or heuristics are termed in order to reach optimal solution. Some of the exact algorithms mentioned are dynamic programming, explicit and implicit enumeration, branch and bound method. But there are some disadvantages of these methods. They work fine with limited number of nodes. (40-80). Thus, heuristics come to play when there is large scale of problems. Heuristics solve specific types of problems. Heuristics comprise metaheuristics. The advantage of metaheuristics is that they show only the way on how to apply the procedures in order to find solutions. Ant Colony Optimization belongs to the category of metaheuristics. Thus, ACO is used in order to solve problems like TSP.

# 4 Methodology

## 4.1 Elitist ACO

Elitist is a variation on the original ACO algorithm where the best solution deposits additional pheromone on each edge that constitutes the current best solution during the global update rule application. Given by the equation:

$$\Delta\tau(r,s) = \sum_{k} \Delta_{rs}^{k} + \partial\tau(r,s)$$

where

$$\partial\tau(r,s) = \begin{cases} \frac{Q_{val}}{L_{gb}} & if\ (r,s) \epsilon\ global\ best\ tour, \\ 0 & otherwise. \end{cases}$$

and $L_{gb}$ is the current global best tour and $Q_{val}$ is a parameter of the number of elite ants . This heuristic prioritizes the pheromone deposited by the ants that are part of the current global best path.

## 4.2 Probabilic Elitist ACO: Our Variation of the Elitist-ACO

Our approach in this study is similar to the elitist ACO, but the additional pheromone contribution from the global best path is not deterministic, rather controlled by another probability factor, $p \in [0, 1]$:

$$\partial\tau(r,s) = \begin{cases} \frac{Q_{val}}{L_{gb}} & if\ (r,s) \epsilon\ global\ best\ tour\ with\ probability, \boldsymbol{p}, \\ 0 & otherwise. \end{cases}$$

If $p < 0$, use the same value of p from the previous iteration.

The elitist version of the algorithm has a tendency to get stuck in local minima due to the recurring pheromone contribution from the current best path starting from very first iteration. The probability factor prevents this saturation from happening and in some sense this variant can be thought of as a hybrid between the original and elitist ACO, thus benefitting from best path reinforcement of elitist ACO as well as new exploration potential in the original version.

Furthermore, we experiment with 2 different ways of choosing the value of the probability factor.

### 4.2.1 Static Probabilistic Elitist-ACO (SPE-ACO):

The 1st approach is to use a fixed static value of $p$. We choose $p$=0.5 to ensure both the elitist and non-elitist update rule has equal chance of being employed. This has potential to prevent saturation at some local minima of the function to be optimized. The downside is that on problem instances where elitist update indeed would have led to quick convergence, however, this method still has equal probability of choosing the non-elitist update rule. Intuitively we want non-elitist update rule to be more likely to be applied when the solution is far from convergence while elitist approach is more fruitful near convergence. Thus, we propose the adaptive dynamic probabilistic elitist approach to capture this behavior.

The elitist version of the algorithm has a tendency to get stuck in local minima due to the recurring pheromone contribution from the current best path starting from very first iteration. The use of probability factor, *p*, may prevent it from an early convergence to the local minima by reducing the pheromone deposit. This approach with a probability factor can be considered as a hybrid between the original-ACO and elitist- ACO, thus benefitting from best path reinforcement of elitist ACO as well as new exploration potential in the original-ACO.

**4.2.2 Adaptive Dynamic Probabilistic Elitist-ACO (ADPE-ACO):**

To capture the behavior of starting out with original update rule being most likely while increasing the probability of elitist updates as we approach convergence, we can define probability *p* adaptive as a function of current best tour length of the current iteration as follows:

$$p = 1 - \frac{the\ current\ best\ global\ tour\ length}{maximum\ global\ tour\ length}$$

If $p < 0$ because the current best tour length is worse than maximum global tour length, use the same *p* from the previous iteration.

Since the true maximum global tour length is unavailable at the beginning of search, we use the tour length obtained in the first iteration of the algorithm itself as the maximum global length in the expectation of the tour length decreases as the search progresses through iteration. Note that our probability ensures that first few iterations will use the pheromone updates of original approach, non-elitist (i.e. $p = 0$). As the iteration proceeds, however, the amount of pheromone update is more taken from the current global best tour which is improved from the previous iteration, the probability to take an update from the elitist is increased. Thus, the update rule of the elitist ACO will be used with more likelihood. Hence, this dynamic probabilistic elitist approach trades the exploration and exploitation using the decreasing *p*; more exploration with the lower *p* at the beginning of the search while more exploitation with the higher *p* using the best tour as the search proceeds.

# 5 Experiments

We generated our input data for the comparative study of the ACO algorithm variants in the following manner. We consider a 2-dimensional square of side 100 units. We pick random non-repeated integral (*x,y*) pairs to generate the list of cities. The Euclidean distance among the city points is the distance between cities for the travelling salesman problem. We generated datasets for 50 and 100 cities for our experiment, where each dataset contains 100 instances of the randomly generated TSP problem. We use number of cities multiplied by 100 as an upper bound of the TSP tour length to initialize the algorithm. Similarly, an arbitrary constant c=20 multiplied by number of cities is used as an upper bound on iterations.

We first present a case study on a 30 city, 10 instance small dataset to highlight some of the key findings. We then present results on the bigger datasets.

For each instance of dataset in the case study, the following results are examined:

i) Number of iterations to convergence

ii) The tour length

Alongside original ACO and elitist ACO, we have incorporated the static value of probability '*p*' to be 0.5 and the discussed adaptive dynamic probability for the pheromone update rule.

# 6 Results

From the original approach, we have obtained the following results from 10 different random problem instances with corresponding required iterations to converge and Tour length:

| Solution Tour Length | | | | |
|---|---|---|---|---|
| TSP instance | Original ACO | Elitist ACO | Static Probabilistic elitist ACO | Dynamic Probabilistic Elitist ACO |
| 0 | 452.312 | 453.079 | 452.312 | 452.312 |
| 1 | 481.137 | 478.069 | 481.137 | 477.923 |
| 2 | 444.356 | 444.356 | 445.391 | 444.356 |
| 3 | 494.77 | 487.918 | 494.821 | 487.352 |
| 4 | 468.841 | 464.9 | 473.739 | 458.675 |
| 5 | 469.742 | 472.831 | 463.905 | 458.454 |

| | | | | |
|---|---|---|---|---|
| 6 | 420.538 | 420.538 | 420.538 | 420.538 |
| 7 | 481.068 | 479.125 | 471.907 | 470.699 |
| 8 | 494.329 | 492.287 | 499.828 | 503.535 |
| 9 | 501.443 | 498.12 | 492.164 | 495.098 |

Table 1: Comparisons on Tour Length for30 cities and 100 ants

Table 1 shows the solution tour length for each instance of problems for four ACO algorithms for 30 cities and 100 ants. The optimal or the best tour lengths are only recorded. For example, for the instance of '0', the number of iterations is 390 and the best tour length is 452.312. The distance between the two nodes are measured by Euclidean distance with *x* and *y* co-ordinates.

For the ACO-Elitist approach we generated the results the same way as we did for the Original-ACO approach to generate the iterations and the tour lengths (optimal/shortest length). Table 2 shows the number of iterations and for respective problem instances in each approach. The results show that in static probabilistic variant the number of iterations are more compared to other approaches. However, the dynamic variant is able to achieve similar quality result in less iterations.

| Number of Iterations | | | | |
|---|---|---|---|---|
| TSP Instance | Original ACO | Elitist ACO | Static Probabilistic elitist ACO | Dynamic Probabilistic Elitist ACO |
| 0 | 390 | 330 | 510 | 90 |
| 1 | 240 | 480 | 240 | 480 |
| 2 | 180 | 210 | 240 | 330 |
| 3 | 240 | 180 | 150 | 210 |
| 4 | 90 | 60 | 390 | 480 |
| 5 | 150 | 120 | 90 | 60 |

| | | | | |
|---|---|---|---|---|
| 6 | 120 | 60 | 90 | 90 |
| 7 | 270 | 150 | 330 | 120 |
| 8 | 150 | 90 | 510 | 150 |
| 9 | 510 | 180 | 480 | 240 |

Table 2: Comparisons of the Number of Iterations for 30 cities and 100 ants.

Figure 1 shows convergence speed to the optimal tour in 4 approaches. Our approach of ADPE-ACO converged to a steady state faster than the other three approaches. But, at the end there is a change in the graph. It shows that our method has still gone further searching for optimal path. Unlike the elitist-ACO and SPE-ACO, ADPE-ACO didn't remain at a certain steady state which may be considered as the plateau of local minima length, but it made further improvement after the plateau. Though a greater number of iterations was required for ADPE-ACO compared to the other original ACO and elitist approach, but ADPE-ACO was more successful in finding the shorter length tour of the near optimal solution.

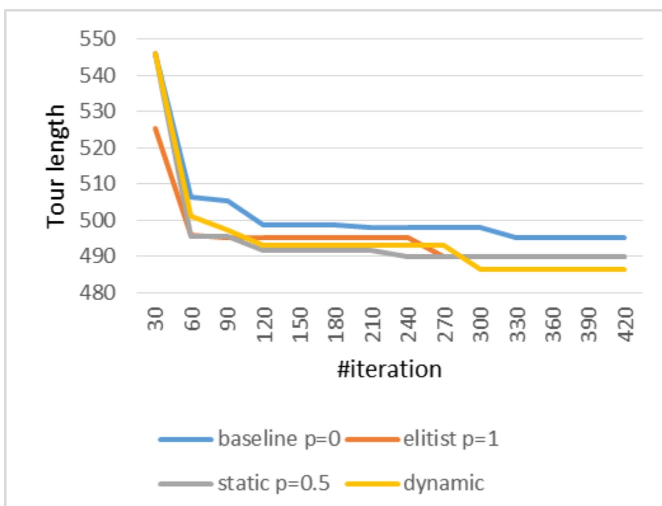


Figure 1: Comparison of the convergence speed.

We ran the 4 algorithms on 100 problem instances of 100 cities and 200 ants. Their results are summarized in Tables 3 and 4.

| TSP Instance | Original ACO | Elitist ACO | Static Probabilistic elitist ACO | Dynamic Probabilistic Elitist ACO |
|---|---|---|---|---|
| 0 | 864.218 | 861.576 | 874.16 | 844.962 |
| 1 | 836.949 | 840.198 | 824.723 | 825.277 |
| 2 | 858.032 | 853.081 | 883.401 | 869.011 |
| 3 | 843.834 | 837.238 | 863.536 | 851.31 |
| 4 | 852.295 | 844.843 | 840.718 | 850.013 |
| 5 | 829.119 | 839.951 | 825.754 | 832.84 |
| 6 | 801.579 | 801.265 | 801.09 | 801.858 |
| 7 | 806.322 | 804.324 | 804.276 | 814.155 |
| 8 | 804.88 | 804.671 | 804.601 | 794.419 |
| 9 | 853.326 | 848.031 | 855.191 | 841.165 |

Table 3: Tour length in the instances of 100 cities and 200 ants (only 10 instances shown out of 100)

| | elitist p=1 | original p=0 | static p=0.5 | dynamic elitist |
|---|---|---|---|---|
| mean | 846.403 | 845.886 | 845.688 | 844.977 |
| stdev | 36.307 | 38.384 | 38.304 | 37.524 |
| # of best results | 21 | 26 | 30 | 26 |
| rank | 4 | 2 | 1 | 2 |

Table 4: Average tour length of 100 cities 200 ants.

# 7 Conclusion

In this paper we studied closely the three different algorithms the original ACO, elitist ACO and our approach of two probabilistic elitist-ACO: static probabilistic elitist-ACO ( SPE-ACO) and adaptive dynamic probabilistic elitist-ACO (ADPE-ACO). Our approach of both probabilistic elitist ACO reasonably outperformed the original-ACO and the elitist-ACO in the points of the average shortest tour length and the convergence speed. In particular, ADPE-ACO made further improvement to the global optimal tour length after it once reached to the steady state near to the optimal tour length for a certain number of iterations, i.e. plateau of the local minima.

# 8 Future Work

Future work will involve doing the study on a larger sample size and doing appropriate statistical analysis. Since we have taken the number of cities to be 30 in our case study, an extended version of the research can include increasing the number of vertices or cities (with respect to the terminology of TSP) and study the results accordingly. The study can be done in a more detailed way after generating results from more problem instances and more comparisons. The probability constraint and other parameters could be varied as well in order to study the difference.

# References


[1] Bifan Li, Lipo Wang, Wu Song, "Ant Colony Optimization for the Traveling Salesman Problem Based on Ants with Memory", ICNC 2008, pp. 496-501, 2008.

[2] M. Dorigo, L.M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem", IEEE Transactions on Evolutionary Computation, vol. 1, pp. 53-56, 1997.

[3] I. Brezina Jr and Z. Čičková, "Solving the Travelling Salesman Problem Using the Ant Colony Optimization," Management Information Systems, 16(4), pp. 010-014, 2011.

[4] R. Durbin, D. Willshaw, "An analogue approach to the travelling salesman problem using an elastic net method", Nature, vol. 326, pp. 689-691,1987.

[5] R.S. Parpinelli, H.S. Lopes, A.A. Freitas "Data mining with an ant colony optimization algorithm", IEEE Trans Evolutionary Computing 6 (4) (2002), pp. 321–332

[6] Z. Zhang and X. Zhang, A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation, Proceedings of the 2nd International Conference on Industrial Mechatronics and Automation, pp. 240-243, 2010.

[7] E. Feller, L. Rilling, C. Morin, Energy-aware ant colony based workload placement in clouds, in: Proceedings of the IEEE/ACM International Conference on Grid Computing (GRID), 2011, pp. 26–33.

[8] A. Colorni, M. Dorigo and V. Maniezzo, "Distributed optimization by ant colonies," the European Conference on Artificial Life, pp. 134-142, 1991

[9] Blum, Christian. "Ant colony optimization: Introduction and recent trends." Physics of Life reviews 2.4 (2005): 353-373.

[10] Dorigo, Marco, and Thomas Stützle. "The ant colony optimization metaheuristic: Algorithms, applications, and advances." Handbook of metaheuristics. Springer US, 2003. 250-285.

[11] Dorigo, Marco, Gianni Di Caro, and Luca M. Gambardella. "Ant algorithms for discrete optimization." Artificial life 5.2 (1999): 137-172.

[12] Lee, Zne-Jung, Chou-Yuan Lee, and Shun-Feng Su. "An immunity-based ant colony optimization algorithm for solving weapon–target assignment problem." Applied Soft Computing 2.1 (2002): 39-47.

[13] Maier, Holger R., et al. "Ant colony optimization for design of water distribution systems." Journal of water resources planning and management 129.3 (2003): 200-209.

[14] Hajizadeh, Yasin, Mike Christie, and Vasily Demyanov. "Ant colony optimization for history matching and uncertainty quantification of reservoir models." Journal of Petroleum Science and Engineering 77.1 (2011): 78-92.

[15] Dorigo, M., 1992. Learning and natural algorithms, PhD thesis, Politecnico di Milano, Milan, Italy

[16] Cordón, Oscar, I. Fernández de Viana, and Francisco Herrera. "Analysis of the best-worst ant system and its variants on the TSP." Mathware and Soft Computing 9.2/3 (2002): 177-192.

[17] Bullnheimer, Bernd, Richard F. Hartl, and Christine Strauss. "A new rank based version of the Ant System. A computational study." (1997).

[18] Di Caro, Gianni, and Marco Dorigo. "Ant colony optimization and its application to adaptive routing in telecommunication networks." (2004).