# Fuzzy Expert Ants to speed up big TSP Problems using ACS

Fardin Abdali Mohammadi, Abdolhosein Fathi

*Faculty of Computer Engineering, Razi University, Kermanshah, Iran.*
*Emails: fardin.abdali@razi.ac.ir, ahfathi@razi.ac.ir*

## ABSTRACT

Ant colony algorithms are a group of heuristic optimization algorithms that have been inspired by behavior of real ants foraging for food. In these algorithms some simple agents (i.e. ants), search the solution space for finding the suitable solution. Ant colony algorithms have many applications to computer science problems especially in optimization, such as machine drill optimization, and routing. This group of algorithms have some sensitive parameters controlling the behavior of agents, like relative pheromone importance on trail and pheromone decay coefficient. Convergence and efficiency of algorithms is highly related to these parameters. Optimal value of these parameters for a specific problem is determined through trial and error and does not obey any rule. Some approaches proposed to adapt parameter of these algorithms for better answer. The most important feature of the current adaptation algorithms are complication and time overhead. In this paper we have presented a simple and efficient approach based on fuzzy logic for optimizing ACS algorithm and by using different experiments efficiency of this proposed approach has been evaluated and we have shown that the presented concept is one of the most important reasons in success for parameter adapting algorithms.

**Keywords:** Ant Colony System, fuzzy, Optimization, Parameter Adapting, Traveling Salesman Problem (TSP).

## 1. INTRODUCTION

Real ants at the time of finding food and in the way back to nest deposit some pheromone on the traveled path. This helps other ants to exploit the food source place. Deposited pheromone on the ground is evaporated gradually. Usually ant colony problems are modeled by a graph. Paths are modeled by graph edges and amount of pheromone on each edge is the weight of that edge. Each ant try to find the shortest tour start traveling graph from an initial node. Ants in each node choose the next node according to a probabilistic state transition rule (ants prefer to move to nodes which are connected by short edge with a high amount of pheromone) and change the amount of edge pheromone by applying a pheromone updating rule.

In ant colony algorithms each ant has a limited form of memory called tabu list in which the current partial tour is stored. Initially each ant is placed on a randomly chosen node. Start from initial node each ant moves from a node to another one. At a node i, the ant k, chose a still unvisited node j probabilistically using a probabilistic state transition function. After all ants have constructed a tour, the pheromones are updated. This is typically done by first lowering the pheromone trail strengths by a constant factor (pheromone decay coefficient) and then the ant are allowed to deposit pheromone on the edges they have visited. Pheromone evaporation enables the algorithm to forget previous bad decisions. The trail update is done in such a form that edges contained in the shorter tours and/or visited by many ants receive a higher amount of pheromone and are therefore chosen with a higher probability in the following iterations of the algorithm.

Ant colony algorithms have some parameters like relative pheromone importance on trail and pheromone evaporation coefficient that convergence and efficiency of algorithms is highly related to them. Usually desirable value of these parameters regarding the problem is determined through trial and error.

This paper is organized as follows. Related works is addressed in Section II. Section III addresses ACS Algorithm. The concept of Fuzzy Expert Ant is addressed in Section IV, ACS Optimization with Fuzzy Expert Ant is addressed in Section V, results are addresses in section VI, and finally the conclusions are presented in Section VII.

## 2. RELATED WORKS

In [2] some studies on effects of parameters of ant colony algorithms have been done. In [3] applying different Experiments on trial and error process have been tried to show that choosing parameters of ant colony algorithms in a special interval produce better results. ACS algorithm is one of the ant colony algorithms that compare to other ant colony algorithms have shown better results. Researches have done on optimizing ACS algorithm by adapting parameters of algorithm. Marcin and Tony adjusted the parameters of ACS algorithm using genetic algorithms (Meta-ACS algorithm) [4] and showed that this approach can help improving efficiency of ant colony algorithms. Abdali and Meybodi using Learning Automata adapt parameters of ACS algorithms (ACSLA algorithm) and showed the efficiency of Learning Automata in adaptation of parameters and optimization of ACS algorithm [5]. Finally, Abdali and Neemat Bakhsh improve ACS by choosing each ant parameters based on a Normal distribution [10]. Recently, by introducing SIMD based GPU processors, parallel versions of ant coloy algorithms are developed and big TSP problems are find an approximate good annswer in a less time required by contrary serial algorithms [11][12]. But, even in parallel algorithms, all ants are restricted by a fixed value for their parameters and this fact could trap algorithms into local minimum.

Along increase in number of cities cost of finding optimized tour is increased and heuristic algorithms don’t obtain a good solution. In this paper using fuzzy ant we have improve ACS algorithm for big problems. The presented approach has been compared to ACS algorithm. Simulation results have shown preference of presented algorithm.

## 3. ACS ALGORITHM

ACS algorithm is one of the best ant colony algorithms that compared to others produces better results [1]. Also ACS algorithm [2] produced better results on many TSPs compared to some genetic algorithms (GA), simulated annealing (SA), and evolutionary programming (EP) [1]. Difference among various ant colony algorithms is in the way the next node chosen and the way of updating pheromone by ants. In ACS algorithms not only ant through passing the edges make them updated but also at the end of every iteration the ant that has constructed the best tour, update the pheromone of generated tour for another time. Another difference there is between ACS algorithm with other ant colony algorithms and that is ants in choosing the next node make use of 2 formulas. Choice of one of these two formulas is made by a random variable, means that an ant positioned on node r chooses the city s to move to by applying the rule given by Eq. (1)

$$s = \begin{cases} \max_{j\in N_i^k} \{\tau_{ij}(t).\eta_{ij}^{\beta}\} & \text{if } q \prec Q_0 \\ \textit{according to AS equation} & \text{otherwise} \end{cases} \qquad (1)$$

where τij(t) is the pheromone, ηij is the inverse of the distance between node i and node j, $N_i^k$ is the set of cities that remain to be visited by ant k positioned on city r (to make the feasible solution), and β is a parameter which determines the relative importance of pheromone versus distance (β>0). q is a random number uniformly distributed in [0 .. 1], and Q0 is a parameter (0≤Q0≤1). The parameter Q0 determines the relative importance of exploitation versus exploration: Every time an ant in city r has to choose a city s to move to, it samples a random number 0≤q≤1. If q≤ Q0 then the best edge (according to Eq. (1)) is chosen (exploitation), otherwise an edge is chosen according to AS equation (biased exploration) [4]. Also in ACS algorithm in every iteration, the ant that construct the best tour update amount of its tour pheromone using the following formula:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \rho\Delta\tau_{ij}^{best}(t) \qquad (2)$$

Where Δτijbest(t) is amount of pheromone that the best ant deposit in iteration t on edge (i,j) and ρ is pheromone decay coefficient. The best ant can be the best current iteration ant or the best ant from the beginning of algorithm execution to the current iteration.

## 4. FUZZY EXPERT ANT

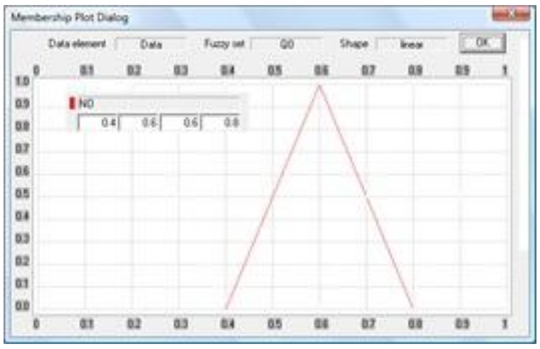
Abdali Mohammadi, ET. Al. shows that distributing ant parameters cause more heuristics and so, give better results [10]. They show that using different values for each ant parameter, lead to better results. Fuzzy Expert Ants utilize this concept and give it a mathematical model. This paper also adopt ant parameters using simple fuzzy rules.

Unlike traditional ant colony algorithms, each Fuzzy Expert Ant has its own parameters. These parameters don't have crisp value; instead they have a fuzzy value with a triangle membership functions. Each membership function represented with a triple (beginning,top,end) corresponds with left, top, and right apex of the triangle.

Regarding the presented results in [3] we have chosen parameters of membership functions so that value of each ant parameters to be close to good values. For instance, memberships function of Ant parameter Q0 represented in Fig. 1. Table 1 show membership functions settings for other parameters.

Each Fuzzy Expert Ant adopts its membership functions parameters with simples "IF …. THEN …" rules. These rules written so that adopts fuzzy parameters

based on the generated tour length of related ant and best tour funded so far. Table 2 shows some of these rules.
This schema is very similar with constant structures Learning Automata adaptation.


Membership Plot Dialog

Data element: Data
Fuzzy set: Q0
Shape: linear

NO: 0.4 | 0.6 | 0.6 | 0.8

**Figure 1.** Fuzzy Value of $Q_0$ parameter

**Table 1** Membership Parameter Setting Used in this study

| Parameter | Beginning | Ending | Top |
|---|---|---|---|
| A | 1 | 5 | 3.5 |
| Q0 | 0.4 | 0.8 | 0.6 |
| B | 1 | 5 | 3 |

**Table 2** Membership Parameter Setting Used in this study

| Rule # | Rule |
|---|---|
| 1 | IF generated_tour<best_global_tour AND Q0<0.8 THEN Q0.Top+=0.1 |
| 2 | IF generated_tour<best_global_tour AND Q0=0.8 THEN Q0.Top=0.4 |
| … | … |

## 5. ACS OPTIMIZATION WITH FUZZY EXPERT ANT

In this section a method for optimizing of ACS algorithm based on Fuzzy Expert Ant Paradigm with fuzzy parameters β, α, and Q0 has been presented. These parameters are interfering in decision making of ants and updating of pheromone. This algorithm like other ant colony algorithms uses some ants to find a solution. In contrary to ACS algorithm that all ants use a global parameters with equal value, in the presented algorithm that we have called it ACS-FZANT for short, each ant for choosing the next node and updating of pheromone uses its own parameters. In the proposed algorithm every ant is equipped with 3 Fuzzy variables, that each of them indicates one of the parameters. Each ant uses their own Fuzzy variables (parameters) in making a decision to choose the next node and updating the pheromone. From the beginning of algorithm execution, value of each variable for each ant is chosen according to Table 1 with this exception that value of the top part of memberships is chosen randomly.

At the end of each iteration, these parameters will be adopted using fuzzy rules. Similar to Meta-ACS and ACSLA algorithms values of these variables change during algorithm execution. At the end of any iteration, only 3 rules are fireable for each ant. This causes simplicity and higher speed to this algorithm compared to Meta-ACS.

**Table 3.** The Comparison of ACS-FZANT to ACS

| Problem Type | Lin318 | | rat783 | | u2152 | |
|---|---|---|---|---|---|---|
| Algorithm | Ave. | STD | Ave. | STD | Ave. | STD |
| *ACS* | 52486 | 843 | 12193 | 241 | 96432 | 1107 |
| *ACS-FZANT* | 50746 | 820 | 11145 | 124 | 78325 | 1055 |
| Number of iterations | 50 | | 50 | | 50 | |

Pseudo-code of ACS-FZANT is like ACS with this difference that each ant has its own parameters and value of these parameters determined from the beginning of algorithm execution. At each decision step, each ant defuzzify its parameters and uses them in making decision or pheromone updating.

## 6. RESULTS

Simulation has been done using an Intel Dual core 2.2 MHz, 1 GB RAM PC. Number of algorithm iterations for each graph has been shown in the last row of simulation results table. Number of iterations for both algorithms is equal to compare ACS-FZANT with ACS. The ACS parameter settings are given in table 3. In our experiments, in both algorithms we have made use of 20 ants in constructing tours except experiment C. Results given are averages of best tour lengths over 10 runs for each graph and standard deviation values for the results. After that in first experiment presented algorithm has been compared with ACS algorithm. Then in the following experiments effect of ants' number for obtaining solution and the speed of their good solution obtainment has been considered.

**Table 4.** ACS Parameter Setting Used in this study

| Parameter | value |
|---|---|
| $Q_0$ | 0.9 |
| $\rho$ | 0.1 |
| $\beta$ | 2 |

### a) The comparison of ACS-FZANT to ACS

In this experiment ACS-FZANT algorithm has been simulated on different graphs. Simulation has been performed on 3 known graphs (lin318, rat783, u2152, [6]). Simulation result has been presented in Table 4. By comparing simulation results, we conclude that ACS-FZANT for big problems meet our object better than ACS. Also standard deviation for obtained results in ACS-FZANT is lower than that of ACS and this show that presented algorithm is more stable. From our results, the ACS-FZANT algorithm like ACS Algorithm does not guarantee finding the optimal solution for a problem.

### b) Considering convergence of ACS-FZANT algorithm

In this experiment we consider convergence of ACS-FZANT algorithm. ACS-FZANT has been run on rat783 graph. Experiment results shown that the rate at which good solutions were found was observed to be quicker using ACS-FZANT algorithm and ACS-FZANT algorithms for finding a solution need less iteration. In other word its convergence speed to a good solution is higher. Fig. 2 has shown the results of this experiment. Each curve is result of averaging 10 run for each algorithm. As it is obvious from the figure the rate at which a good solution was found is quicker. Other graphs also have been tried and produced similar results.

### c) Effect of ants' number

In this experiment effect of ants' number has been considered in ACS-FZANT and ACS algorithms. Results from these experiments have been shown in Table 5.

**Table 5** Effect of ant's number

| Problem Type | rat783 | | rat783 | | rat783 | |
|---|---|---|---|---|---|---|
| Algorithm | Ave. | STD | Ave. | STD | Ave. | STD |
| *ACS* | 12137 | 144 | 12193 | 241 | 12231 | 205 |
| *ACS-FZANT* | 11096 | 184 | 11145 | 124 | 11165 | 154 |
| Ants' Number | 10 | | 20 | | 40 | |

Increase in ants' number causes increase in time. As it is obvious from results, increase in ants number don't help improvement in algorithms and 10 or 20 ants are suitable.

### d) Similarity between ACS-FZANT with parameters adapting algorithms

Finding optimized parameters while solving a problem is not the only reason for success of adapting algorithms. It seems that cause success to these algorithms is sweeping of parameters space and using them in producing solutions. For instance in adapting algorithms each ant choose some value for parameters and with them construct a complete tour. Then regarding produced tour adapt the parameters. Obtained results in this paper show correction of this claim. In this paper we also use some kind of adaptation. In fact using different parameter values causes more exploration.

### e) Convergence of parameters

In this section, we verify the convergence of ants' parameters. The results in the various Experiments suggest that Q0 is robust at a value of 0.2. Parameter β varies between 2 and 4 with lower values being more appropriate when more accurate solutions are needed.

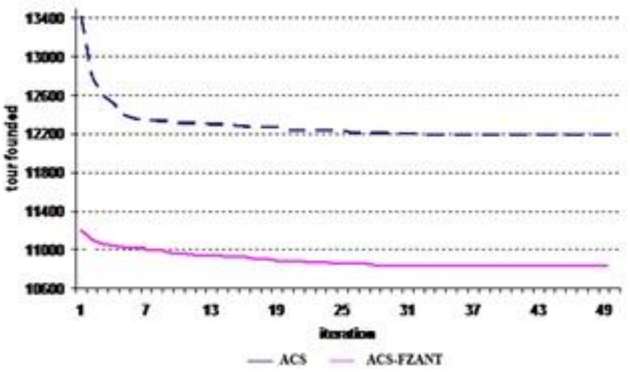


**Figure 2.** Comparing speed of solution obtainment of ACS- FZANT with that of ACS

## 7. CONCLUSIONS

In this paper by using fuzzy variables and adaptation technique we have improved ACS algorithm for big problems. Making use of simulation we have shown that proposed approach has a high efficiency in convergence speed and quality of produced solution compared to one of the best ant colony algorithms.

## REFERENCES

[1] E., Dorigo M., Theraulaz G. Swarm Intelligence: From Natural to Artificial Systems. New York: Oxford University Press, 1999.
[2] Dorigo M., Gambardella L.M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Trans. E Comp. vol. 1, pp. 53-66, 1999
[3] Coloni, A., Dorigo, M., and Maniezzo, V., An Investigation of some Property of an Ant Colony, Parallel Problem Solving from Nature, No. 2, Elsevier Science Publication, 1992
[4] Marcin L.P., Tony W., Using Genetic Algorithms to Optimize ACS-TSP, 2002, http://citeseer.nj.nec.com/, AT: February 2002.
[5] Abdali, M.F., Meybodi, M.R., Adaptation of Ant Colony Parameters Using Learning Automata, CSICC2004, Tehran, Iran, 2004

[6] Reinelt G., TSPLIB Documentation. Institut f¨ur Angewandte Mathematik, Universit ¨at Heidelberg. (1995), http://www.research.att.com/~dsj/chtsp/download.html, AT: February 2002.

[7] Dorigo M., Di Caro, G., The Ant Colony Optimization meta-heuristic, New Ideas in Optimization, McGraw Hill, London, UK, 1999, pp. 11–32.

[8] Dorigo M., Di Caro, G., The Ant Colony Optimization Meta-heuristic. New Ideas in Optimization, McGraw Hill, London, UK, 1999, pp. 11–32

[9] Dorigo M., DiCaro G.: The ant colony optimization meta-heuristic. In D.Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization. McGraw-Hill, 1999.

[10] Abdali, M.F., Fathi, A.H., and Manzori M.T., Optimizing ACS for Big TSP Problems Distributing Ant Parameters, ISCIT2006, Bangkok, Thailand, 2006.

[11] Dawson, L.; Stewart, I., "Improving Ant Colony Optimization performance on the GPU using CUDA," Evolutionary Computation (CEC), 2013 IEEE Congress on , vol., no., pp.1901,1908, 20-23 June 2013

[12] Akihiro Uchida, Yasuaki Ito, Koji Nakano, "An Efficient GPU Implementation of Ant Colony Optimization for the Traveling Salesman Problem," icnc, pp.94-102, 2012 Third International Conference on Networking and Computing, 2012