

The Effect of Updating the Local Pheromone on ACS Performance using Fuzzy Logic

Abdellatif El Afia, Safae Bouzbita, Rdouan Faizi

National School of Computer Science and Systems Analysis (ENSIAS), Mohammed V University in Rabat, Morocco

Article Info

Article history:

Received Feb 21, 2017

Revised May 13, 2017

Accepted May 27, 2017

Keyword:

Ant colony algorithm

Fuzzy logic controller

Parameter adaptation

ABSTRACT

Fuzzy Logic Controller (FLC) has become one of the most frequently utilised algorithms to adapt the metaheuristics parameters as an artificial intelligence technique. In this paper, the ξ parameter of Ant Colony System (ACS) algorithm is adapted by the use of FLC, and its behaviour is studied during this adaptation. The proposed approach is compared with the standard ACS algorithm. Computational results are done based on a library of sample instances for the Traveling Salesman Problem (TSPLIB).

Copyright © 2017 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Abdellatif El Afia,

National School of Computer Science and Systems Analysis (ENSIAS),

Mohammed Ben Abdallah Regragui Avenue, Madinat Al Irfane, BP 713, Agdal Rabat, Morocco.

Email: a.elafia@um5s.net.ma

NOMENCLATURES

J Random proportional rule

Q Random variable uniformly distributed between [0, 1]

q_0 Parameter to control exploration and exploitation

$j_k(r)$ Set of cities not yet visited by ant k positioned on city r

L_{nn} length of a nearest neighbour tour and n is the number of cities

L_{best} length of the globally best tour found from the beginning of the algorithm

S_{μ}^2 length standard deviation

Greek Symbols

$\tau(r, s)$ pheromone amount between city r and city s

$\eta(r, s)$ Heuristic information between city r and city s

β Parameter that determines the relative importance of pheromone versus heuristic value

ξ Local pheromone evaporation parameter

τ_0 Initial value of the pheromones

ρ Global pheromone evaporation parameter

μ Mean of lengths

1. INTRODUCTION

The Ant Colony System (ACS) is one of the most effective extensions of the basic Ant System (AS) algorithm (Dorigo and Gambardella, 1996) [1-3]. The ACS algorithm is described in pseudo-code (Figure 1). The procedure of ACS technique consists of three steps: Pheromone initialization, Construct Ants Solutions, and Global pheromone updating.

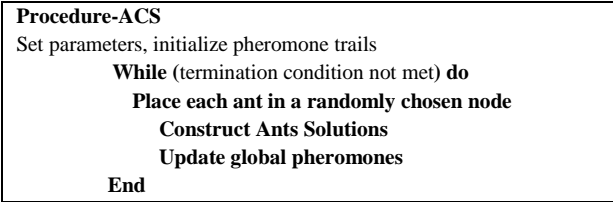


Figure 1. The Ant Colon System procedure.

The Ant Colony System applied to Transport Salesman Problem can be formulated as follow: m ants are initially placed on n cities randomly, each ant builds a complete solution in the Construct Ants Solution using the so called pseudo-random proportional rule Equation (1) and "Equation (2):

$$j = \operatorname{argmax}_{u \in J_k(r)} [\tau(r, u)] [\eta(r, u)]^\beta \quad \text{if } q \leq q_0 \quad (1)$$

Where:

- $J_k(r)$: the set of cities not yet visited by ant k positioned on city r .
- $\tau(r, s)$: the pheromone amount between city r and city s .
- $\eta(r, s)$: the heuristic information between city r and city s .
- β : the parameter that determines the relative importance of pheromone versus heuristic value.

That is, the best edge is chosen with a probability q_0 . Otherwise, with probability $(1 - q_0)$, an edge is selected by biased exploration according to the following Equation:

$$p_{rs}^k = \begin{cases} \frac{[\tau(r, s)] \cdot [\eta(r, s)]^\beta}{\sum_{u \in J_k(r)} [\tau(r, u)] [\eta(r, u)]^\beta} & \text{if } s \in J_k(r) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

During the construction of solutions, each ant modifies the amount of pheromone on the visited edges by applying the local updating rule:

$$\tau(r_k, s_k) = (1 - \xi) \tau(r_k, s_k) + \xi \tau_0 \quad (3)$$

Where:

- ξ : the local pheromone evaporation parameter.
- τ_0 : the initial value of the pheromones.

In the global pheromone step, once all ants have built a solution, the amount of pheromone on edges is modified again by the best ants, using the global updating rule:

$$\tau(r_k, s_k) = (1 - \rho) \tau(r_k, s_k) + \frac{\rho}{L_{best}} \quad (4)$$

Where:

- L_{best} : the length of the globally best tour found from the beginning of the algorithm.
- ρ : the global pheromone evaporation parameter.

Several approaches have been proposed to adapt the parameters on Ant Colony algorithms [4-8]. Espacially the adaptation using Fuzzy Logic, in this part the most important are presented. In the first approach, Amir et al., developed in their work, a Fuzzy Logic Controller (FLC) to adapt the parameters and q_0 automatically throughout the run according to definite performance measures which are the error of the best-so-far tour compared to the best-known tour to the TSP problem and the variance among the solutions found by the population of ants.

Also, in [9] FLC was used to improve ACO. In their work, a solution is constructed by an ant based on pheromone trails and heuristic information for solving the reliability problem for a series system, in order

to find one technology for each subsystem. The Fuzzy set of this approach is the heuristic information related to the subsystem in consideration. In [10] Olivas et al., suggested an improved ACO by dynamically adapting the responsible parameter for the evaporation of the pheromone with the use of fuzzy logic. In this paper, authors tried to control the abilities for diversification and intensification of the search space. To do so, they used two metrics that measure the algorithm performance, which are diversity and iteration as inputs of the Fuzzy system and the parameter as output. On the other hand, Fuzzy Logic was used to vary other metaheuristic parameters. Valdez et al [11] described a hybridization of PSO and GA using Fuzzy Logic for decision making and parameters adaptation. Thus, three Fuzzy systems were proposed; the first one is responsible for deciding which are the best results of the FPSO + FGA, and the two second ones are in charge of changing the values of the crossover K_1 the mutation K_2 , the social acceleration c_1 , and the cognitive acceleration c_2 . In [12] a Fuzzy Logic approach was proposed to dynamically adapt the cognitive and the social factors c_1 and c_2 to improve the convergence and diversity of the population in PSO algorithm. Three Fuzzy Systems were modelled for adapting the c_1 and c_2 parameters, with respect to three performance measures, which are the diversity of the swarm, the average error, and the iterations of the algorithm. The performance of ACS depends strongly on the values set to parameters. Thus, varying parameters while solving a problem can enhance the performance of the algorithm. In this paper, the Fuzzy Logic was used to dynamically adapt the pheromone parameters of ACS based on the approach proposed by Olivas et al [10]. To do this, first we applied the Fuzzy Logic to the ξ parameter, then we compared the obtained results with the standard ACS algorithm.

This paper is constructed as follows: in Section 2 we outline the proposed method. The experimental part is discussed in the third section. From our experimental results, we give some conclusions and suggestions for future work in Section fourth.

2. RESEARCH METHOD

To dynamically adapt the local pheromone parameter in ACS, we have used a Fuzzy Logic System [14] that is characterized by two inputs and one output. In fact, as inputs in each case we used a proportion of elapsed iterations and a measure of the diversity of the colony compared to the best ant, while the output is the parameter to be adapted, in our case the parameter. Where,

$$\text{Iteration} = \frac{\text{Current iteration}}{\text{Total of iterations}} \quad (5)$$

$$\text{Diversity} = \frac{1}{m} \sum_{i=1}^m \sqrt{\sum_{j=1}^n (x_{ij}(t) - \bar{x}_j(t))^2} \quad (6)$$

where, Current iteration is the number of past iterations, and total of iteration is the entire number of iterations needed for testing the algorithm, m is the population size, i is the number of the ant, n is the entire number of dimensions, j is the number of the dimension, x_{ij} is the j dimension of the ant i , \bar{x}_j is the j dimension of the current best ant of the colony.

Procedure CAS-FLC

1. Calculate Iteration from "Eq.(5)" and Diversity from "Eq.(6)"
2. Fuzzify the Iteration and Diversity using the membership functions described in figures 3 and 4.
3. Evaluate the fuzzy rules using "Eq. (7)".
4. Defuzzify the results of evaluation phase and output the crisp value for ξ using "Eq. (8)"
 - For each** ant in the population **do**
 - If** $q < q_0$ **then** with probability q_0 choose the node to move to using "Eq. (1)"
 - Else** with Probability $(1 - q_0)$ choose the node to step to using "Eq. (2)"
 - Update pheromone amount using "Eq. (3)"
 - Until** all ants build a solution

Figure 2. Adaptive local parameter using FLC.

In Figure 2, we described the adaptation of ξ parameter using the FLC after one established iteration. In facts, in each iteration ants build solutions incrementally to the problem, by moving through neighbour components solution of the problem, using the transition rule described by Equations (1) and (2). While moving, an ant modifies the amount of pheromone using the local pheromone updating procedure described by Equation (3). Once all ants have terminated a solution the amount of pheromone is modified

again using the global pheromone updating procedure. The obtained information from the first iteration are used to calculate the first inputs of the FLC algorithm, and so on.

2.1. Fuzzification

The Fuzzification process is the first step in the FLC system, which corresponds to transforming the crisp values into fuzzy grades using the membership functions. This process facilitates the application of the rule set. For each input variable, we define three membership functions (MF), to define a qualitative category for each one: {Low, Medium, High}. In practice, there are various shapes of membership functions that can be used in the fuzzification process, for example: Triangular MFs, Trapezoidal MFs, Gaussian MFs, Generalized bell MFs, π -Shaped Membership Function, S-Shaped Membership Function [15]. In this work, we used the triangular membership functions that are very popular, easy to implement, and respond to our needs. The input variables with their membership functions are illustrated in Figure 3 and Figure 4.

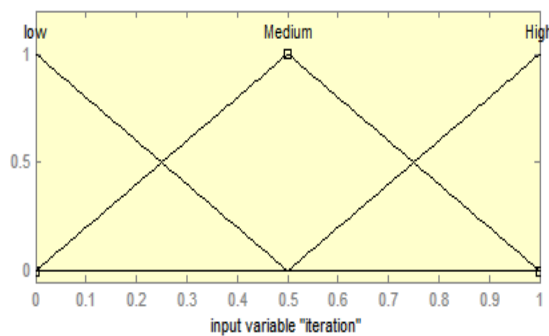


Figure 3. Iteration as input variable.

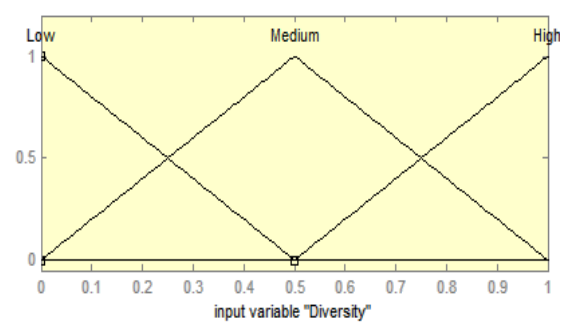


Figure 4. Diversity as input variable

Figure 3 illustrates the three triangular membership functions of the iteration input variable with a range from 0 to 1. In Figure 4 the three triangular membership functions of the Diversity input variable are shown with a range from 0 to 1. Let x the crisp value of Iteration and y the crisp value of Diversity. For each input we calculate the degree of membership: μ_{Low} , μ_{Medium} , and μ_{High} . Where, Low=[0, 0.5], Medium=[0, 1], and High=[0.5, 1]

2.2. Rule Evaluation

In fuzzy logic, there are various ways to define a fuzzy rule. Indeed, they can be divided into three main classes: the fuzzy conjunction, the fuzzy disjunction, and the fuzzy implication [16]. In this work, we used a Mamdani's fuzzy conjunction fuzzy rule. Once the variables and membership functions are designed, we define the rule base which is composed by IF- Then rules. In fact, the rule base is developed according to some knowledge about the ACS algorithm and the chosen metrics. The rules (Figure 5) of the proposed fuzzy system with Iteration and Diversity as input and ξ as output are as follows:

<p>If (Iteration is Low) and (Diversity is Low) then (ξ is Low) If (Iteration is Low) and (Diversity is Medium) then (ξ is MediumLow) If (Iteration is Low) and (Diversity is High) then (ξ is Medium) If (Iteration is Medium) and (Diversity is Low) then (ξ is MediumLow) If (Iteration is Medium) and (Diversity is Medium) then (ξ is Medium) If (Iteration is Medium) and (Diversity is High) then (ξ is MediumHigh) If (Iteration is High) and (Diversity is Low) then (ξ is Medium) If (Iteration is High) and (Diversity is Medium) then (ξ is MediumHigh) If (Iteration is High) and (Diversity is High) then (ξ is High)</p>
--

Figure 5. IF-THEN rules of our fuzzy system

To evaluate the fuzzy rules we used the Min fuzzy set operation, assuming that we are using the Mamdani's conjunction operator (AND). For each rule we return the lowest value from the calculated degrees of membership of the two inputs.

$$\mu_i = \min_{j,k=1,2,3} \{ \mu_j(x), \mu_k(y) \} \quad i = 1, 2, \dots, 9 \tag{7}$$

Where, i is the index of the rule, j and k are indexes for the fuzzy sets {Low, Medium, High} for x and y. Then, the results of all rules are summed together to produce a set of fuzzy outputs.

2.3. Defuzzification

After the evaluation of the rules, we obtain a fuzzy output that's need to be transformed to a crisp value using one of the defuzzification methods. In fact, the commonly used techniques for defuzzification are: Mean of Maximum (MOM) method, Center of Gravity method, and the height method [16], [17]. Figure 6 shows the ξ parameter as output variable, with a range from 0 to 1, and granulated into five triangular membership functions. Where, output set is: $S = \{ \frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6} \}$.

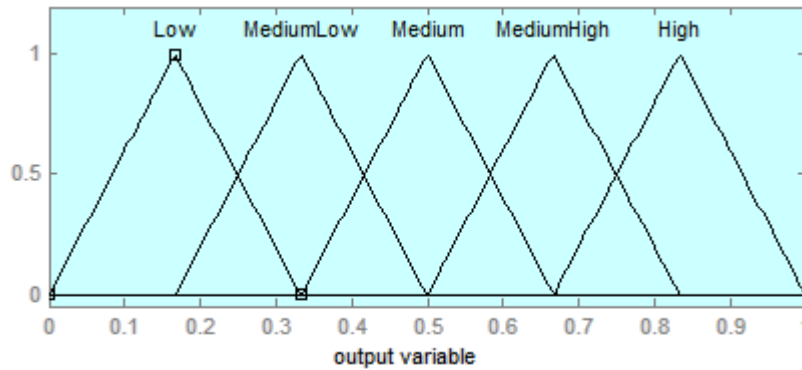


Figure 6. ξ as output variable

In order to obtain the output variable in crisp value, we defuzzify the obtained results from the inference process using the Center of Gravity (COG) algorithm described by Equation 8:

$$\xi = \frac{\sum_{i=1}^9 [w_i \mu_i]}{\sum_{i=1}^9 [\mu_i]} \tag{8}$$

Where, p=9 is the number of output membership function, w_i is the singleton of output membership function, and μ_i the result of all rule evaluation as shown in Table 1.

Table 1. The singleton of the output membership function

		X		
		Low	Medium	High
Y	Low	$w_1 = 1/6$	$w_4 = 2/6$	$w_7 = 3/6$
	Medium	$w_2 = 2/6$	$w_5 = 3/6$	$w_8 = 4/6$
	High	$w_3 = 3/6$	$w_6 = 4/6$	$w_9 = 5/6$

Where, w_1 corresponds to Low membership function of the output, w_2 corresponds to Medium Low membership function of the output, w_3 corresponds to Medium membership function of the output and so on.

3. RESULTS AND DISCUSSION

To test the effectiveness of the proposed algorithm, we compared it to the standard ACS with a set of benchmark TSP instances from the TSPLIB [18]. Table 2 tabulates the size and the optimal tour length of each instance. We run the ACS algorithm with the following parameters: $\beta = 2$, $q_0 = 0.9$, $\rho = 0.1$ and $m=10$, which were proven to be the best setting parameters for ACS performance [19]. Programming by Matlab R2013a, the instances will be run 30 times separately, 1000 iterations each time. The stop condition is: 100 as the maximum number of ACS iterations without improving. The initial position of ants is set randomly on all experiments. The best comparison results are shown in Table 2.

Table 2. Characteristics of TSP benchmark instances.

TSP	att48	berlin52	ch130	d198	eil51	eil76	eil101	kroA100	lin105	Pr226
Number of cities	48	52	130	98	51	76	101	100	105	226
est known solutions	10628	7542	6110	15780	426	538	629	21282	14379	80369

3.1. Comparison on the Solution Accuracy

Comparing the running of ACS with a fixed set of parameters against the proposed method shows better results for both minimum and average length in most of the instances, especially when dynamically adapting the parameter. In Table 3, it can also be observed that in instances of small sizes the best found solutions for the two algorithms are almost the same with a privilege in the average solutions for the proposed algorithm. But in large size instances there is a big difference between the best solutions that are found for each of the two algorithms. Thus, it can be concluded that as the size of problem becomes larger as the proposed algorithm can offer better results. The experimental results reflect the role of learning to provide the best solutions.

Table 3. Summary of results using ACSFL algorithm for TSP instances

TSP	ACSFL			ACS		
	Min	Avg	CPUtime	Min	Avg	CPUtime
att48	33523.70	33715.49	42.99	33523.70	33692.67	38.75
berlin52	7544.36	7546.53	32.43	7544.36	7556.46	29.45
ch130	6246.24	6348.35	1093.7	6234.56	6371.55	1676.9
d198	16032.71	16327.04	2590.2	16147.38	16414.42	1739.1
eil51	428.98	432.94	116.38	428.98	435.48	111.1
eil76	548.49	556.31	328.6	552.92	558.02	388.08
eil101	646.44	662.76	480.9	657	669.03	387.8
kroA100	21285.44	21611.54	410.18	21355.28	21748.18	438.56
lin105	14382.99	14524.95	254.9	14382.99	14559.82	460
Pr226	80468.49	81853.71	1353.7	80763.10	82127.78	3448.2

3.2. Comparison on the Convergence Speed

It can be noted from the Table 3, that the time of finding the best length for the proposed algorithm when adapting the parameter outperforms both the conventional ACS and the proposed algorithm when varying the parameter. However, it can be seen that the time of finding the same best length is less in the proposed algorithm when adapting the parameter than the two others. The Figures 7(a), 7(b), 7(c) and 7(d) below show the result of running both algorithms on four chosen instances of TSP benchmarks which are: eil51, kroA100, eil101, and lin105.

The Figures actually go in line with these observations, since a very big difference is noticed between the solutions found by the three algorithms, the proposed algorithm when varying parameter converges to a better solution than the conventional one and the proposed algorithm when adapting the parameter in all the figures. In addition to the quality of solution, there is a faster convergence in the proposed algorithm when dynamically adapting the parameter.

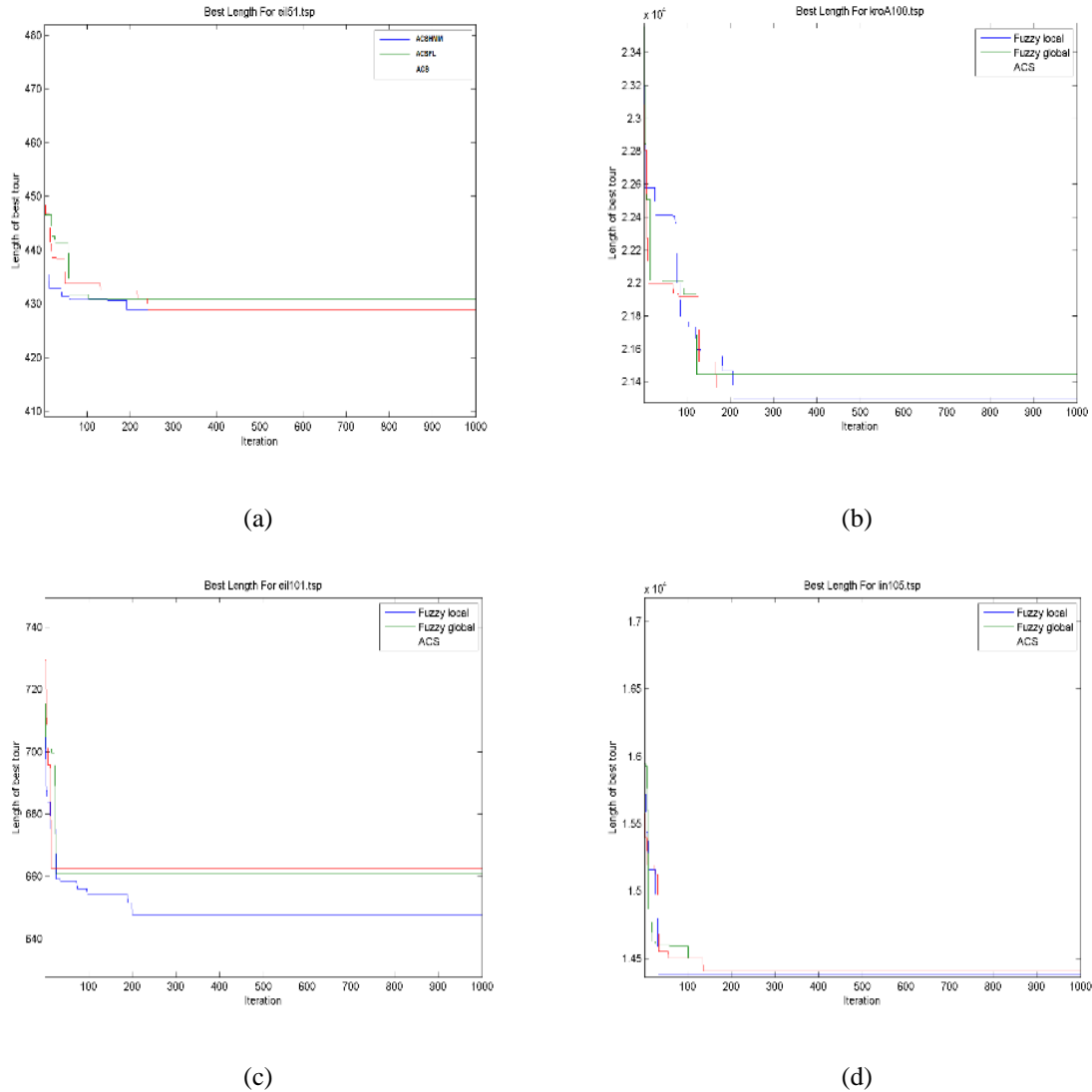


Figure 7. (a) Sample run on eil51 instance, (b) Sample run on kroA100 instance, (c) Sample run on eil101 instance, (d) Sample run on lin105 instance

3.3. Statistical Test

To compare the proposed method with the standard one, we have used the T-Test as a statistical test that compares the means of lengths returned by the proposed method and the standards ACS according to the following Equation:

$$t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{S^2_{\mu_1} - S^2_{\mu_2}}{n}}} \tag{9}$$

Where, n is the lengths size.

The results obtained from applying this test are illustrated in "Table 4". The 30 experiments for each instance are the used parameters for the tests, the null hypothesis ($H_0: \mu_1 \leq \mu_2$) says that the proposed method returns greater lengths of averages when compared with the other method, while the alternative hypothesis ($H_1: \mu_1 > \mu_2$) says that the proposed algorithm returns better average when compared with the other method, the level of significance is 5 percent, and the critical value $t_0 = 1.699$, so the rejection region is for all values of T-Test lowers than t_0 . From the results in table 4, the proposed method fail to reject the null hypothesis only in 2 instances, and this is for the smallest problems which are the easiest ones, however the proposed method can achieve better results with level of significance of 5 percent in all other results.

Table 4. Results of comparison using T-Test

TSP	att48	berlin52	ch130	d198	eil51	eil76	eil101	kroA100	lin105	Pr226
ACS	-1.306	-0.611	-3.634	-2.490	-3.002	-3.823	-4.678	-2.273	-3.487	-3.157
Fuzzyglobal	0.590	-2.225	-2.126	-3.432	-3.385	-5.292	-3.959	3.990	-5.663	-4.938

4. CONCLUSIONS

This paper proposed a new evolved Ant Colony System Algorithm based on a fuzzy system so as to dynamically adapt the local pheromone parameter that has a crucial impact in avoiding falling in the local best optimum during the construction solution phase. The simulation result on TSP showed that the proposed method that dynamically adapt the local pheromone parameter has higher convergence speed and better quality of optimal solution. In other words, the effect of local updating is to make a better use of pheromone information to explore new best solutions. The proposed method when adapting parameter gives a flexible control of pheromone information which balances between the exploration search and exploitation then finding better solutions.

REFERENCES

- [1] M. Dorigo, L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", *IEEE Transactions on Evolutionary Computation*, 1(1), 53-66, 1997.
- [2] M. Dorigo, C. Blum, "Ant Colony Optimization Theory: A Survey", *Theoretical Computer Science*, 344(2), 243-278, 2005.
- [3] M. Dorigo, L. M. Gambardella, "Ant Colonies for the Travelling Salesman Problem", *BioSystems*, 43(2), 73-81, 1997.
- [4] H. Yu, "Optimized Ant Colony Algorithm by Local Pheromone Update", Vol. 12, No. 2, pp. 984 ~ 990, 2014.
- [5] F. Xiao-rong, F. Xing-jie, "A Dynamic Multi-nest Ant Colony Algorithm for Aircraft Landing Problem", *TELKOMNIKA Indonesian Journal of Electrical Engineering*, Vol. 12, No. 3, pp. 2196 ~ 2202, 2014.
- [6] M. Jafari, H. Khotanlou, "A Routing Algorithm Based on Ant Colony, Local Search and Fuzzy Inference to Improve Energy Consumption in Wireless, Sensor Networks", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 3, No. 5, pp. 640~650, 2013.
- [7] C. Qi, "Vehicle Routing Optimization in Logistics Distribution using Hybrid Ant Colony Algorithm", *TELKOMNIKA*, Vol. 11, No. 9, pp. 5308~5315, 2013.
- [8] D.N. Le, "PSO and ACO Algorithms Applied to Optimization Resource Allocation to Support QoS Requirements in NGN", *International Journal of Information & Network Security (IJINS)*, Vol.2, No.3, June 2013, pp. 216~228.
- [9] F. Ahmadizar, H. Soltanpanah, "Reliability Optimization of a Series System with Multiple-Choice and Budget Constraints using an Efficient Ant Colony Approach", *Expert Systems with Applications*, 38(4), 3640-3646, 2011.
- [10] F. Olivas, F. Valdez, O. Castillo, "Ant Colony Optimization with Parameter Adaptation Using Fuzzy Logic for TSP Problems", In *Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization* (pp. 593-603). Springer International Publishing, 2015.
- [11] F. Valdez, P. Melin, O.Castillo, "An Improved Evolutionary Method with Fuzzy Logic for Combining Particle Swarm Optimization and Genetic Algorithms", *Applied Soft Computing*, 11(2), 2625-2632, 2011.
- [12] P. Melin, F. Olivas, O. Castillo, F. Valdez, J. Soria, M. Valdez, "Optimal Design of Fuzzy Classification Systems Using PSO with Dynamic Parameter Adaptation Through Fuzzy Logic", *Expert Systems with Applications*, 40(8), 3196-3206, 2013.
- [13] A. Sombra, F. Valdez, P. Melin, O. Castillo, "A New Gravitational Search Algorithm using Fuzzy Logic to Parameter Adaptation", *IEEE Congress on Evolutionary Computation*, 2013.
- [14] L. A. Zadeh, "Fuzzy sets", *Information and Control*, 8(3), 338-353, 1965.
- [15] Y. Bai, D. Wang, "Fundamentals of Fuzzy Logic Control-Fuzzy Sets, Fuzzy Rules and Defuzzifications", In *Advanced Fuzzy Logic Technologies in Industrial Applications* (pp. 17-36). Springer London, 2006.
- [16] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller", II. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2), 419-435, 1990.
- [17] Sheikh, M. R. I., Junji, T. (2011), "Smoothing Control of Wind Farm Output Fluctuations by Fuzzy Logic Controlled SMES", *International Journal of Electrical and Computer Engineering*, 1(2), 119.
- [18] G. Reinelt, "Tsplib Discrete and Combinatorial Optimization", 1995.
- [19] T. Stützle *et al.*, "Parameter Adaptation in Ant Colony Optimization", In *Autonomous search* (pp. 191-215), Springer Berlin Heidelberg. 2011.