Modified Ant Colony Optimization with mutation and Reset Pheromone for Travelling Salesman Problem

Chiabwoot Ratanavilisagul[1]

Faculty of Applied Science, Department of Computer and Information Science, King Mongkut's University of Technology North Bangkok (KMUTNB), Bangkok, Thailand
*E-mail: chaibwoot@gmail.com, chaibwoot@hotmail.com

ABSTRACT

Ant Colony Optimization (ACO) algorithm is a stochastic algorithm. It is used for solving combinational optimization problem. The ant colony walks along pheromone density from nest of ant to feeding sources. It leads to create shortest path from nest of ant to feeding sources. Ordinarily, ACO encounters the problem of trapping in local optimum. To solve this problem, the mutation technique is applied with ACO. However, the mutation technique cannot improve searching performance of ACO. This paper proposed improving ACO by the result from mutation technique is applied with pheromone of ants. In addition, when the problem of trapping in local optimum occurs, pheromone of ants is reset to solve this problem. The proposed technique is tested on twenty maps from the Traveling Salesman Problem Library (TSPLIB) and gives more satisfied search results in comparison with ACOs.

Keywords: ant colony optimization, travelling salesman problem, optimization, mutation operator

1. INTRODUCTION

Reference (1) proposed Ant Colony Optimization algorithm (ACO) in 1991. It get inspired from the foraging behavior of ant colony (2), (3), (4). ACO has been well-known used for solving optimization problems (5) such as job scheduling problem, traveling salesman problem (TSP), network routing and vehicle routing problem etc. ACO endeavor to search for shortest paths from ant's nest to feeding sources. Each ant walks along the pheromone density and releases pheromone among walking. This behavior causes the pheromone density in short path. It is very high. On the other hand, the pheromone density in long path is very low (6), (7).

The benefits of ACO, it can search for the good solution rapidly. Moreover, it is efficient for solving TSP and similar problems (8), (23), (24), (25). For cons of ACO, it has the high opportunity to trap in the local optimum (8), (9), (23), (24).

To solve problem of trapping in the local optimum, many algorithms such as Particle Swarm Optimization (PSO) applied with the mutation technique (15), (16). Many researchers (8), (12), (13), (14) enhance diversity of ant colony by adding the mutation technique to the process of ACO. The experiment results of these technique showed that mutation techniques can enhance searching performance of the standard ACO and obtain better solution than the standard ACO.

Reference (8) proposed the mutation technique is similar to mutation strategy of the Genetic Algorithms (GA). The mutation technique can enhance diversity of ant colony.

Reference (13) proposed the mutation technique apply with ACO. The concept of this algorithm can summarize as follows. The mutation technique is applied with the best ant. Then, if the result from mutation is better than the result from the best ant, the best ant is replaced by it. This algorithm is called that ant colony optimization algorithm with mutation mechanism (MACO).

Reference (14) proposed mutation technique apply with ACO. The concept of this algorithm can summarize as follows. The mutation is applied with some ants. Then, if the result from mutation is better than the mutated ant, the mutated ant is replaced by it. This algorithm is called that ant colony optimization algorithm with uniform mutation using self-adaptive approach (ACOMS).

From the experiment results show the performance of MACO and ACOMS in solving combinatorial optimization problems is better than that of the previous modified ACO.

The cons of ACOMS and MACO, the evaluation calls of these techniques are more than that of the standard ACO per a generation. It means the number of guessing solutions that are more than the standard ACO per a generation. So, it is not fair. If evaluation calls are equal, these techniques may obtain solutions worse than ACO.

In addition, the results from the mutation cannot guide searching next generation. So, the mutation technique which is added cannot improve searching of ACO. But, these techniques enhance number of guess solutions. Surely, solutions of these techniques are better than solutions of standard ant colony if the number of generations is equal.

For process of ACO, the pheromone is determinant for creating tours of a solution or ant. Hence, this paper proposes the result from mutation is better than the mutated ant. That uses for improving pheromone. Next generation, process of ACO has the high opportunity to create better solution than the result from mutation. It can enhance searching performance of ACO. However, this technique leads to trap in local optimum. Therefore, the pheromone is updated from the best result only. Ant colony is rarely diversity of population. To handle trapping local optimum problem, this paper proposed a novel pheromone reset technique apply with ACO when ant colony happens to trapping in local optimum. The proposed algorithm can get better solutions when it is compared by standard ACO and other compared ACO algorithms.

A set of maps in the TSPLIB (18), (19) is used to compare the standard ACO (6) by source code from (17), MACO (13), ACOMS (14), and the proposed technique. The results show that the solution quality of the proposed technique is better than other compared ACO algorithms in the TSPLIB.

The rest of this paper is organized as follows. Section 2 explains TSP, the standard ACO, and ACO with mutation. Section 3 explains Modified Ant Colony Optimization with mutation and Reset Pheromone. Section 4 explains the case studies of TSP, the experiment setup and presents the experiment results. Section 5 concludes the paper with a brief summary.

2. RELATED WORK

2.1 Travelling Salesman Problem

Traveling salesman problem (TSP) (20) is popular combinational discrete optimization problem and NP-hard problem. This problem has been used for many engineering applications such as computer networks and the design of hardware devices (27). As amount of cities are increased, searching the optimal tour becomes very hard. The salesman attempts to travel all cities only once and to create a closed tour of the shortest path. A complete weighted graph G = (V, A) can be used to represent a TSP. Where V is the set of n cities and A is the set of paths fully

connecting all cities. Each edge (d_{ij}) is the distance between cities i and j. d_{ij} is as follows:

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad (1)$$

$V = \{1,.., n\}$, $A = \{(i, j): i, j \in V\}$,
$d(i, j) = d(j, i)$, Edge $(i, j) \in A$

2.2 Ant colony optimization

This paper shows applying ACO with TSP. An ant (A) is the set of paths fully connecting all cities. So, ant is solution of TSP. The fitness function of TSP is as follows:

$$fitness\_of\_ant = \sum_{i=1}^{n-1} d_{i,i+1} + d_{n,1} \qquad (2)$$

ACO can be described briefly as follow. Initially, each edge has an initial pheromone $\tau_{ij}(0)$ between two cities. The next step is select city of ant. The first city of each ant is randomly selected, and then each ant selects next city according to probability function as follows:

$$P_{i,j}^{k} = \begin{cases} \dfrac{[\tau_{i,j}(t)]^{\alpha}[n_{i,j}]^{\beta}}{\sum_{t \in allowed}[\tau_{i,j}(t)]^{\alpha}[n_{i,j}]^{\beta}} & , j \in allowed \\ 0 & , otherwise \end{cases} \qquad (3)$$

Where β is a parameter which determines the relative importance of pheromone versus distance (β > 0), $\eta_{ij} = 1/d_{ij}$ is the inverse of the distance, τ_{ij} is the pheromone, and $P_{i,j}^{k}$ is the probability of ant k chooses to move from city i to city j.

The result from formula Eq. (3) that causes selecting edges which are shorter and have a greater amount of pheromone. After selection cities of ant has completed, the fitness of ant is calculated by Eq. (2). The fitness of each ant is used to update pheromones according to Eq. (4). This process continues until a stopping criterion is met.

$$\Delta\tau_{i,j}(t+n) = \rho\tau_{i,j} + \Delta\tau_{i,j} \qquad (4)$$

$$\Delta\tau_{i,j} = \sum_{k=1}^{m} \tau_{i,j}^{k} \qquad (5)$$

$$\Delta\tau_{i,j}^{k} = \begin{cases} \dfrac{Q}{L_k} & ,if\_the\_k^{th}\_ant\_uses\_edge(i,j)\_in\_its\_tour \\ 0,\_otherwise \end{cases} \qquad (6)$$

Where is calculated by Eq. (6), $\Delta\tau_{ij}$ is the sum of new enhanced pheromones at this edge is degree of dissipate for pheromones, ρ is a coefficient such that (1 - ρ) represents the evaporation of trail between time t and t+n, $\tau_{i,j}^{k}$ is the amount of pheromones of the k^{th} ant at its edge between t and t+n, and Q is consistent value, L_k is fitness of ant k.

2.3 Ant colony optimization with mutation

Reference (13) proposed MACO. This paper uses concept of simple mutation operation of Genetic Algorithm (GA) (21), (26). The concept of MACO is can be concluded as follows: After update pheromone step, randomly choose one or more elements in the local best solution (S_{nbest}), change them. The result from mutation is called that (S'_{nbest}). If (S'_{nbest}) is better than (S_{nbest}), replace (S_{nbest}) by (S'_{nbest}).

Reference (14) proposed ACOMS. The concept of ACOMS is can be concluded as follows: The mutation operator is applied with ant. The new ant is generated from mutation. If the fitness of the new ant is better than that of ant, replace ant with the new ant.

From the experiment results show that the quality solution of both ACOMS and MACO is better than the previous modified ACO algorithms in solving TSP.

However, the cons of ACOMS and MACO are using over evaluation call because both techniques in each generation are updated by mutation technique and ACO technique. Hence, they can guess solutions more than ACO in each generation. As well as, they can obtain better solutions than the standard ACO.

On the other hand, if the evaluation calls of ACO, ACOMS and MACO are defined to equal, ACOMS and MACO may obtain worse solutions than ACO in some maps.

In ACO process, the pheromone creates tours of an ant or a solution. If pheromone occurs changing, it affects searching solution next generation.

Both techniques can improve solution by applying mutation technique. They can obtain better solutions than the standard ACO. But, the results from the mutation of both techniques are not applied with pheromone. So, the mutations of both techniques are not affects searching solutions next generation. As well as, they cannot improve searching of ACO. Ant colony has still moved along previous path. It leads to trap in local optimum.

3. MODIFIED ANT COLONY OPTIMIZATION WITH MUTATION AND RESET PHEROMONE

As previously mentioned, the mutation operator can improve solution of ACO. The result from improving pheromone affects to search for solution of ACO next generation. Surely, if the pheromone is updated from good solution, it has the high opportunity to create better solutions. In the same case, if the mutation is applied with good solution, it has the high opportunity to create better solutions. Hence, this research proposed the best ant is selected after ACO step create tours each generation. Then, the best ant is applied with the mutation operation. The best result from the mutation, it is used to update pheromone. The pheromone is only updated by the best result from mutation. The result from improving pheromone, it affects to search for solution next generation. This algorithm can improve searching for solution and get better solution.

However, the result from pheromone is only updated by the best solution. It leads to trap in local optimum. The pheromone is created from only the best solution. It is rarely the diversity of population. It occur the trapping in local optimum problem over the standard ACO.

The trapping in local optimum problem occur with ACO. It cause from pheromone which is stagnation. The pheromone which is stagnation creates the same solution over and over again, that no better solutions can be found anymore (6), (9). Therefore, the unchanged number of consecutive solution can indicate state of ant colony when ant colony traps in local optimum. Generally, resetting should be used when colony trap or trend to trap in local optimum (15), (16) to distribute ant colony to search for other areas.

To handle the trapping in local optimum problem, this paper proposed reset pheromone when all ants occur trapping in local optimum. In order that pheromones occur changing, it creates the new tours and can jump out local optimum. The easy techniques to indicate the trapping state of the ant colony is to monitor the unchanged number of the best consecutive solution. Thereby, this paper proposed the reset period (RP) considered from the unchanged number of consecutive solution.

The result from searching both ACO algorithm and the mutation technique can enhance searching performance concurrently.

Moreover, the reset pheromone is applied when ant happen trapping in local optimum to solve trapping in local optimum problem. Hence, this paper proposed the mutation technique and reset technique apply with ACO. The proposed technique is called Modified Ant Colony Optimization with mutation Pheromone and Reset Pheromone (ACOMR). Pseudo code of ACOMR is shown below:

```
Initialize edges (i, j) and pheromone (τ_ij)
While termination condition ≠ true do
   For each number of ants
    Random the start city
    Reset the best solution of this generation (BG)
    For until cites in list of an ant is full
        Choose the next city with probability according to formula (3)
        The chose city is not repeat city in list of this ant
        Insert chose city into list of this ant
    End For
    Evaluate fitness of each ant
    If fitness of each ant is better than fitness of BG
        BG = this ant
    End If
    If fitness of each ant is better than fitness of solution
        Update fitness of solution = fitness of this ant
    End If
   End For

   Apply the mutation operation with BG only as follow:
   For city i begin 0 to the number of cities
    Random the city j
    Swap city i and city j (i≠j)
    Evaluate fitness of the result from swap
    If fitness of the result from swap is better than fitness of BG
        BG = the result from swap
    Else
        Swap city i and city j
    End If
   End For
   Update pheromone by BG only according to formula (4)
   Times of BG consecutive unchanged ++
   If the times of BG consecutive unchanged >= reset period
    All pheromones in table are initialized
    Set the times of BG consecutive unchanged to 0
   End If
End While
```

4. EXPERIMENTS AND RESULTS

4.1 Parameters Setting

Parameters are as follows for all experiments: $\rho = 0.05$, $Q = 1$, $\beta = 1$, $\alpha = 1$, as suggested by (17), $\tau_0 = (n \times L_{nn})^{-1}$ where n is the number of cities, and L_{nn} is the length of list of ant that is produced by the nearest neighbor heuristic (22). The number of experiments of each map is set as 20 runs. To guarantee fairness of performance measurement, evaluation calls of all the algorithms are set equation. The number of evaluation of each run in experiments is 1,000,000 evaluations. The number of ants is set as 50. The non-ACO parameters are as follows: The compared algorithm parameters (MACO, ACOMS) are set according to suggested by the original papers. For proposal algorithm, the reset period (RP) is set as 30.

All experimented algorithms are not used by the well-known path improvement heuristics such as 2-opt (10), and Lin–Kernighan (11). This research is conducted by a personal computer of AMD FX-8320 with 3.5 GHz CPU, 8 GB RAM and Visual C++ 2010 as the programming language. All maps are used in experiment are from TSPLIB (18), (19).

4.2 The measures of algorithm performance

The measures of algorithm performance in the experiments are as follows: the average best fitness value (ABF) is the average of best fitness in the final generation from all running. ABF indicates the solution searching efficiency of an algorithm. The closer the ABF to the optimum point of a method, the better the method. SR is the success round. SD is the standard deviation. SR

and SD indicate the solution searching reliability of an algorithm.

4.3 Experiment of proposed algorithm

Table 1. Comparative Results of ACO, MACO, ACOMS, and ACOMR on TSP.

Techniques Problem name	Optimum Point	ACO ABF	SD	SR
burma14	3323	3323.00	0.00	20
ulysses16	6859	6859.00	0.00	20
gr17	2085	2085.00	0.00	20
gr21	2707	2707.00	0.00	20
ulysses22	7013	7023.35	11.44	11
fri26	937	937.00	0.00	20
bayg29	1610	1620.90	10.53	6
bays29	2020	2041.60	10.58	0
Oliver30	420	420.00	0.00	20
dantzig42	699	709.25	5.21	0
swiss42	1273	1308.70	10.86	0
eil51	426	439.30	5.61	1
berlin52	7542	7847.60	71.22	0
brazil58	25395	26516.10	213.24	0
st70	675	722.50	7.24	0
pr76	108159	116410.00	1000.80	0
eil76	538	576.50	8.65	0
rat99	1211	1356.65	15.50	0
eil101	629	713.55	8.24	0
kroA100	21282	23220.80	266.32	0

Table 1. (Continued).

Techniques Problem name	Optimum Point	MACO ABF	SD	SR
burma14	3323	3323.00	0.00	20
ulysses16	6859	6859.00	0.00	20
gr17	2085	2085.00	0.00	20
gr21	2707	2707.00	0.00	20
ulysses22	7013	7014.50	2.60	15
fri26	937	937.00	0.00	20
bayg29	1610	1636.65	11.03	1
bays29	2020	2030.75	8.62	2
Oliver30	420	420.00	0.00	20
dantzig42	699	704.35	5.18	3
swiss42	1273	1278.90	7.56	11
eil51	426	433.05	5.23	3
berlin52	7542	7675.90	90.12	0
brazil58	25395	26275.40	186.07	0
st70	675	701.25	14.58	2
pr76	108159	115423.00	1594.01	0
eil76	538	570.65	8.31	0
rat99	1211	1350.05	17.53	0
kroA100	21282	22607.90	314.73	0

Table 1. (Continued).

Techniques Problem name	Optimum Point	ACOMS ABF	SD	SR
burma14	3323	3323.00	0.00	20
ulysses16	6859	6859.00	0.00	20
gr17	2085	2085.00	0.00	20
gr21	2707	2707.00	0.00	20
ulysses22	7013	7031.05	9.15	4
fri26	937	937.00	0.00	20
bayg29	1610	1639.75	11.40	0
bays29	2020	2045.60	10.93	0
Oliver30	420	420.00	0.00	20
dantzig42	699	710.90	7.42	1
swiss42	1273	1311.35	10.00	0
eil51	426	440.45	6.13	0
berlin52	7542	7856.05	106.11	0
brazil58	25395	26614.30	228.03	0
st70	675	722.75	14.84	0
pr76	108159	117107.00	1264.44	0
eil76	538	584.10	7.08	0
rat99	1211	1370.05	18.90	0
kroA100	21282	23169.60	202.73	0

Table 1. (Continued).

Techniques Problem name	Optimum Point	ACOMR ABF	SD	SR
burma14	3323	3323.00	0.00	20
ulysses16	6859	6859.00	0.00	20
gr17	2085	2085.00	0.00	20
gr21	2707	2707.00	0.00	20
ulysses22	7013	7013.00	0.00	20
fri26	937	937.00	0.00	20
bayg29	1610	1610.00	0.00	20
bays29	2020	2020.00	0.00	20
Oliver30	420	420.00	0.00	20
dantzig42	699	699.00	0.00	20
swiss42	1273	1273.00	0.00	20
eil51	426	426.00	0.00	20
berlin52	7542	7542.00	0.00	20
brazil58	25395	25421.10	47.43	2
st70	675	675.00	0.00	20
pr76	108159	109551.00	708.40	2
eil76	538	538.00	0.00	20
rat99	1211	1216.55	8.21	10
kroA100	21282	21324.10	89.28	15

From the experimental results of Table 1 show that the quality solution of ACOMR is better than that of ACO, MACO, and ACOMS because of its lowest ABF all tested maps. The reliability of ACOMR is better than that of ACO, MACO, and ACOMS because of its lowest SD and its highest SR all tested maps. Hence, ACOMR outperforms ACO, MACO, and ACOMS in case of reliability and quality solution. The experiment can prove that the pheromone very affect to searching performance of ACO. The proposed algorithm only improves the pheromone. The quality solution and the reliability of proposed algorithm are better than that of other compared algorithms (MACO,

and ACOMS) which directly improve ants or solution. In some maps, ABF and SD of both MACO and ACOMS are higher than that of ACO, such as BAYG29, In this case show that if number of evaluation call is equal, MACO and ACOMS may obtain worse solutions than ACO in some cases.

5. CONCLUSION

The mutation operator is applied with ACO to solve trapping in local optimum. Generally, the mutation operation is applied with ant. These algorithms cause number of evaluation calls that are increased. Moreover, they cannot improve searching of ACO. From these cons, this paper proposed the results from the mutation update pheromone of ants to improve searching of ACO. But, it suffers the trapping in local optimum problem. To solve trapping in local optimum problem, the resetting pheromone is applied with ACO when ant colony trapped in local optimum. The proposed technique is called that ACOMR. ACOMR can solve trapping in local optimum problem and improve searching of ACO. Hence, it can get better solutions. From the experimental results show that the proposed ACOMR outperforms ACO, MACO, and ACOMS with regard to the reliability and quality of solutions in all maps.

6. REFERENCES

[1] A. Colorni, M. Dorigo, V. Maniezzo, “Distributed optimization by ant colonies,” Proceedings of European Conference on Artificial Life, Paris, France, pp. 134-142, 1991.

[2] J. L. Denebourg, J. M. Pasteels, and J. C. Verhaeghe, “Probabilistic behavior in ants: A strategy of errors?,” J. Theoret. Biol., vol. 105, pp. 259-271, 1983.

[3] J. L. Denebourgand S. Goss, “Collective patternsand decision-making,” Ethology, Ecology & Evolution, vol. 1, pp. 295-311, 1989.

[4] S. Goss, R. Beckers, J. L. Denebourg, S. Aron and J. M. Pasteels, “How trail laying and trail following can solve foragingproblems for ant colonies,” in Behavioral Mechanisms of Food Selection, R. N. Hughes Ed., NATO-AS1 Series. Berlin: Springer-Verlag, vol. 20, 1990.

[5] X. Yang and J. s. Wang, “Application of improved ant colony optimization algorithm on traveling salesman problem, ” 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, 2016, pp. 2156-2160.

[6] M. Dorigo, V. Maniezzo, A. Colorni, “Ant system: optimization by a colony of cooperating agents,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 26, pp. 693–698, pp.1-13, February 1996.

[7] 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 - 66, April 1997.

[8] X. Song, B. Li, and H. Yang, “Improved Ant Colony Algorithm and its Applications in TSP,” Sixth International Conference on Intelligent Systems Design and Applications, october 2006.

[9] A. Colorni and, M. Dorigo, “Heuristics from nature for hard combinatorial optimization problems,” International Trans Operational Research, pp. 1–21, 1996.

[10] S. Lin., “Computer solutions of the traveling salesman problem,” Bell Syst. J., vol. 44, pp. 2245–2269, 1965.

[11] S. Lin and B. W. Kernighan, “An effective heuristic algorithm for the traveling salesman problem,” Oper. Res., vol. 21, pp. 498–516, 1973.

[12] Q. WU, and J. ZHANG, “ACO With the Characteristic of Mutation”, Computer research and development, pp. 240-245, 1999.

[13] N. Zhao, Z. L. Wu, Y. Q. Zhao, and T. F. Quan, “Ant colony optimization algorithm with mutation mechanism and its applications,” Expert Syst. Appl. , vol. 37, pp. 4805-4810, July 2010.

[14] R. S. Jadon, U. Datta, “Modified ant colony optimization algorithm with uniform mutation using self-adaptive approach for travelling salesman problem,” Computing, Communications and Networking Technologies, 2013.

[15] C. Ratanavilisagul, “Dynamic population size and mutation round strategy assisted modified particle swarm optimization with mutation and

reposition," International Electrical Engineering Congress, pp. and449–452, 2016.

[16] C. Ratanavilisagul, and B. Kruatrachue, "A modified particle swarm optimization with mutation and reposition," International Journal of Innovative Computing, Information and Control, vol. 10, pp. 2127–2142, December 2014.

[17] http://yarpiz.com/53/ypea103-ant-colony-optimization

[18] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95

[19] http://stevedower.id.au/blog/research/oliver-30

[20] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy-Kan and D. B. Shmoys Eds., "The Travelling Salesman Problem," New York: Wiley, 1985.

[21] D. E. Goldberg, "Genetic Algorithm in Search," Optimization and Machine Learning, Addison-Wesley, pp. 372, 1989.

[22] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, "An analysis of several heuristics for the traveling salesman problem," SIAM J. Comput.,vol. 6, pp. 563–581, 1977.

[23] V. Selvi, and R. Umarani, "Comparative Analysis of Ant Colony and Particle Swarm Optimization Techniques," International Journal of Computer Applications, vol. 5., pp. 1–6, August 2010.

[24] Y. Zhang, Z. Pei, J. Yang, Y. Liang, "An Improved Ant Colony Optimization Algorithm Based on Route Optimization and Its Applications in Travelling Salesman Problem," IEEE, BIBE, pp. 693–698, 2007.

[25] M. Dorigo and, LM. Gambardella, "A study of some properties of ant-Q . In: Voigt H-M, Ebeling W, Rechenberg I, Schwefel H-S, eds. Proceedings of the 44th International Conference on Parallel Problem Solving from Nature," Berlin: Springer-Verlag, pp. 656–665, 1996.

[26] J. H. Holland, "Adaptation in Natural and Artificial Systems," University of Michigan press, pp. 211, 1979.

[27] G. Laporte, "The Traveling Salesman Problem – an overview of exact and approximate algorithms," European Journal of Operational Research 59 pp. 231–247, 1992.