Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Implementation of K-Means and crossover ant colony optimization algorithm on multiple traveling salesman problem

To cite this article: N Kusumahardhini *et al* 2020 *J. Phys.: Conf. Ser.* **1442** 012035

Implementation of K-Means and crossover ant colony optimization algorithm on multiple traveling salesman problem

N Kusumahardhini, G F Hertono and B D Handari

Department of Mathematics, Faculty of Mathematics and Natural Sciences (FMIPA), Universitas Indonesia, Kampus UI Depok, Depok 16424, Indonesia

Corresponding author's e-mail: bevina.dh@gmail.com

Abstract. Multiple Traveling Salesman Problem (MTSP) is a generalization of the Traveling Salesman Problem (TSP). MTSP is an optimization problem to find the minimum total distance of m salesmen tours to visit several cities in which each city is only visited exactly by one salesman, starting from origin city called depot and return to depot after the tour is completed. In this paper, K-Means and Crossover Ant Colony Optimization (ACO) are used to solve MTSP. The implementation is observed on three datasets from TSPLIB with 2, 3, 4, and 8 salesmen. Analysis of results using K-Means and Crossover ACO will be compared. The effect of selecting a city as depot on the total travel distance of tour will also be analyzed.

Keywords: multiple traveling salesman problem, K-Means, crossover ant colony optimization

1. Introduction

The multiple traveling salesman problem (MTSP) is a generalization of the traveling salesman problem (TSP), which involves determining a set of routes for m salesmen who all start and end at a city called depot [1]. Each salesman will have to visit different cities exactly once starting from depot and turn back again to depot. MTSP has some possible variations based on number of depots, number of salesmen, time windows, and other restrictions.

Compared to TSP, MTSP is better in modeling the problems in everyday life as it can handle more than one salesman. Since TSP belongs to the class of NP-hard problems [2], it is obvious that MTSP is also an NP-hard problem. Some metaheuristics approach had been performed to solve MTSP, such as Modified Genetic Algorithm [3], Modified Ant Colony [4], Sweep Algorithm and Elite Ant Colony Optimization [5].

In this paper, MTSP will be solved using combination of K-means and crossover ACO based on Latah [6]. K-Means is one of clustering method, which divide a set of data into a set number of clusters. Ant Colony Optimization (ACO) is an optimization algorithm developed by Prof. Marco Dorigo in 1990s. This algorithm is inspired from ant colonies social behavior in finding the shortest path from their nest to a food source [7]. In ACO, pheromone is an important aspect that plays role for ants to choose the path to be traversed. When the intensity of the pheromone on a path is too high compared to other paths, the path tends to always be chosen by the ant or the path will continue to be exploited in the selection of path while the other path will be rarely passed. To improve the exploration of ants in search of the path, pheromone crossover is implemented so that the exploitation and exploration of ants in choosing the path will be more balanced [6]. Crossover is an operator of the Genetic Algorithm [8], which involves two parental chromosomes to obtain a descendant (child) chromosome.

2. K-Means and crossover ACO to solve MTSP

K-means is a clustering method used to divide which cities that each salesman must visit. After clustering, crossover ACO is applied in each cluster to make route for each salesman.

2.1. Dividing cities for each salesman

In MTSP, number of cities is defined as n and the number of clusters to be formed is equal to the number of salesmen ($k = m$). The purpose of K-means is to divide the set of n data points into k clusters in which each data point belongs to the cluster with the nearest mean.

To divide the cities, K-means begins by setting the set of k centroid, which is randomly selected. Within each iteration, each city will assign into particular cluster based on the Euclidean distance between cities with the nearest centroid. After that, the centroid will be recalculated using the equation as follows [9]:

$$c_j = \frac{1}{N_j} \sum_{q=1}^{N_j} x_q \tag{1}$$

where N_j is the total number of cities in cluster j, c_j is the centroid of cluster j, and x_q is the q-th city belongs to cluster j.

The process of grouping the cities to the nearest centroid and centroid recalculation will be repeated until the termination conditions of K-Means are met.

After clustering, each salesman has a set of cities to visit. The route for each salesman is determined using crossover ACO.

2.2. Forming route for each salesman

In ACO, artificial ants are used to construct solutions in a probabilistic way based on pheromone values. ACO has undergone several developments. Some of the ACO algorithms have been developed for TSP, such as Ant System (AS) [10], Elite Ant System (EAS) [7], Max Min Ant System [11], and Ant Colony System (ACS) [12].

At the beginning of any tour, the ants start at a depot and choose the next cities consecutively. The tour continues until all the cities are visited and the ants turn back to the depot again. Every ant selects the next city independently. The city selection rule is based on ACS. In ACS, ants will depart from city i to city j, where city j is chosen based on the following equation [12]:

$$j = \begin{cases} \arg\max\{\tau_{ij}\eta_{ij}^{\beta}\}, \text{if } \mathrm{q} \le \mathrm{q}_0, \\ S, \text{otherwise,} \end{cases} \tag{2}$$

where q is a value chosen randomly between 0 and 1, and q_0 is a given parameter between 0 and 1. Ant k in city i choose to walk to city j based on pheromone trail and heuristic value. If $q > q_0$ is selected, then the selection of cities to be visited is performed based on the transition rule in the AS. The rules of transition in AS is as follows [12]:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{j\in N_j}\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}, \text{if } \mathrm{j} \in \mathrm{N}_{\mathrm{i}}^{\mathrm{k}}, \\ 0, \text{otherwise,} \end{cases} \tag{3}$$

$p_{ij}^{k}(t)$ denotes the probability of ant k in city i will travel to city j at t^{th} iteration. τ_{ij} is the amount of pheromone trail in the arc (i,j). η_{ij} is the heuristic value of the arc (i,j), $\eta_{ij} = \dfrac{1}{d_{ij}}$, where d_{ij} is the distance of city i to city j. α and β are parameters that show respectively the effect of pheromone and heuristic value on path selection. N_i^k is the set of unvisited cities when ant k is in city i.

When all ants have completed their journey, pheromone trail will be updated as follows [14]:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta t_{ij}^{k}(t), \tag{4}$$

where $\tau_{ij}(t+1)$ is the amount of pheromone trail in the arc (i,j) at (t+1)th iteration ρ is the evaporative coefficient. $\Delta t_{ij}^{k}(t)$ is the amount of pheromones trail left by ant k at tth iteration, which is determined by following equation [14]:

$$\Delta t_{ij}^{k}(t) = \begin{cases} \dfrac{Q}{L_k}, \text{if arc (i, j) is passed by ant k,} \\ 0, \text{otherwise,} \end{cases} \tag{5}$$

where L_k is the cost (total distance) of the ant k tour and Q is a constant.

The pheromone trail is limited to an interval $[\tau_{\min}, \tau_{\max}]$ in each iteration. Based on Max Min Ant System, $\tau_{\max}$ is defined as follows [5]:

$$\tau_{\max} = \frac{1}{\rho F}, \tag{6}$$

where F is the total cost (distance) of the best solution that has been found and ϱ is the evaporation coefficient. Then $\tau_{\min}$ is defined as follows [5]:

$$\tau_{\min} = \frac{\tau_{\max}(1-P_{dec})}{(\frac{n}{2}-1)P_{dec}} \tag{7}$$

where P_{dec} is the probability of generating the best solution and n is the number of cities each salesman must visit.

Crossover is one of the operators in the genetic algorithm. Crossover is a process of creating a new individual (heredity) through a combination of randomly selected genetic material from two or more parental chromosomes [8]. If the offspring obtained are better than the parents, then the offspring will replace the parents. The crossover method used in this paper is 2-point crossover.

The pheromone before and after the update become 2 parental chromosomes. Parent 1 is a chromosome consisting of pheromones prior to the update. Parent 2 is a chromosome consisting of updated pheromones. Apply 2-point crossover between two parents. Two child pheromones will be formed by crossover. Use both pheromones to form route. If the result is better, then the pheromone will be used in the next iteration.

3. Computational results

K-Means and crossover ACO algorithm are applied and tested on several datasets from TSP problems of TSPLIB, namely Att48, Pr76, and Rat99. The parameters are as follows.

$\tau_0=10^{-6}$, $\rho=0.1$, $\alpha= 0.1$, $\beta=2$, $q_0=0.9$, $Q=1$, $P_{dec}=0.8$

In each problem, the number of salesmen (m) used are 2, 3, 4, and 8. The number of ants is 10. The termination condition of ACO is when the iteration reaches a maximum of 100 iterations.

First, we applied the algorithms with random selection of one city as depot. Experiment is conducted on each dataset and the number of salesmen; one city is selected as a depot randomly. Each experiment was done 100 times.

Based on table 1, the total value of travel distance tends to increase as the number of salesmen increases. However, in Pr76 with 4 salesmen can result a smaller total distance compared to 3 salesmen.

Second, we used structured manner in depot selection. The selection of depot in a structured manner is done by setting the entire city into a candidate depot on each data and the number of salesmen. After travel distance of each depot are calculated, then select depot that has the minimum

Table 1. Results based on random selection of depot

Dataset	m	The City Number as Depot	Total Distance		Time
			Best Solution	Average Solution	
Att48	2	20	36,968.25038	38,330.25272	29.54
	3	41	42,617.51461	43,502.50718	30.32
	4	10	51,938.59454	52,344.46546	30.94
	8	39	57,261.39839	57,466.66417	40.13
Pr76	2	17	128,827.6784	132,175.171	51.73
	3	51	131,967.1461	135,052.5815	56.52
	4	32	129,033.9794	130,653.541	59.36
	8	17	183,976.0696	184,673.7198	69.78
Rat99	2	62	1,342.6971	1,386.669514	86.17
	3	82	1,575.725198	1,594.096002	89.46
	4	80	1,652.964704	1,671.281702	91.28
	8	85	2,180.872964	2,205.084698	97.52

Table 2. Results based on selection of depot in structured manner

Dataset	m	The City Number as Depot	Total Distance		Time
			Best Solution	Average Solution	
Att48	2	25	34,606.19681	35,177.61043	28.87
	3	21	38,621.8996	39,294.67312	29.38
	4	15	42,832.45028	43,338.07541	30.13
	8	25	54,038.42506	73,370.18086	38.24
Pr76	2	33	119,770.2327	122,627.5992	51.67
	3	28	118,901.9492	119,814.5166	54.92
	4	28	123,403.8674	124,286.7924	58.43
	8	19	163,934.8089	164,562.1419	69.65
Rat99	2	49	1,280.153809	1,296.433068	85.22
	3	57	1,454.478577	1,470.35254	87.46
	4	59	1,491.957	1,501.668395	90.78
	8	48	1,844.908092	1,850.738142	97.64

distance. The city depot will be called the MTSP depot. Furthermore, experiment is repeated 100 times using the MTSP depot for each dataset and the number of salesmen.

Table 2 is the result of K-Means and crossover ACO implementation with the selection of depots in the structured manner. It appears that in the same dataset with different salesmen numbers, the city that selected as depot can be different. The total value of travel distance tends to increase as the number of salesmen increase, except in Pr76 with 3 salesmen. The total distance travelled in Pr76 with 3 salesmen is smaller than 2 salesmen.

Based on the results in table 1 and table 2 it can be seen the total travel distance in table 2, has the better total distance. So, the selection of the depot plays a role in generating a minimum distance. Also, for different number of salesmen with same dataset, different city can be selected as depot that produce the minimum distance.

We compared best solution results using K-Means and crossover ACO in table 2 and best solution using K-Means and ACO. The experiment using K-Means and ACO was conducted 100 times with

Table 3. Comparison of K-Means & Crossover ACO and K-Means & ACO

Dataset	m	Best Solution	
		K-Means & Crossover ACO	K-Means & ACO
Att48	2	34,606.19681	40,955.23469
	3	38,621.8996	45,988.16961
	4	42,832.45028	47,884.10975
	8	54,038.42506	61,559.95431
Pr76	2	119,770.2327	134,185.6134
	3	118,901.9492	141,565.357
	4	123,403.8674	149,802.3226
	8	163,934.8089	189,448.711
Rat99	2	1,280.153809	1,641.21491
	3	1,454.478577	1,649.071721
	4	1,491.957	1,806.315615
	8	1,844.908092	2,647.6236

termination criterion 100 iteration for each dataset and the number of salesmen. The depot used by each dataset and the number of salesmen is the same as the depot used in table 2. The comparison result can be seen in table 3. Based on table 3, it can be seen that the result of solving MTSP using K-Means and crossover ACO has smaller value than K-Means and ACO for each case.

4. Conclusions

The K-Means and crossover ACO methods can be implemented to solve MTSP. K-Means is applied to divide the cities that each salesman will visit. Then, the crossover ACO is applied on every cluster from K-Means results to get the optimal travel route from each salesman. In this paper, the algorithms are applied on 3 datasets from TSPLIB, they are Att48, Pr76, and Rat99 with the number of salesmen are 2, 3, 4, and 8. The implementation results show the city selection as depot can affect the total distance of the salesmen tours. The computational time required to complete MTSP using K-Means and crossover ACO will increase as the number of salesmen and number of cities. The resulting MTSP solution using the K-Means and crossover ACO methods has smaller value than the K-Means and ACO methods.

References

[1] Bektas T 2006 *Omega* **34** 209–19
[2] Russel R A 1977 *Oper. Res.* **25** 517–24
[3] Tang L, Liu J, Rong A and Yang Z 2000 *Eur. J. Oper. Res.* **124** 267–82
[4] Junjie P and Dingwei W 2006 *Proc. The First International Conference on Innovatice Computing, Information and Control (ICICIC),* Beijing (Piscataway: IEEE) pp 210–3
[5] Yousefikhoshbakht M and Sedighpour M 2016 *Proc. Romanian Academy A* **13** 295–302
[6] Latah M 2016 *International Journal Engineering Research Technology (IJERT)* **5** 430–4
[7] Dorigo M, Maniezzo V, and Colorni A 1996 *IEEE T. Syst. Man. Cy.* B **26** 29-41
[8] Holland J H 1975 *Adaptation in Natural and Artificial System* (Michigan-University of Michigan Press)
[9] Bishop C M 2006 *Pattern Recognition and Machine Learning* (New York: Springer)
[10] Dorigo M 1992 *Optimization, Learning and Natural Logarithms* (Milan: Dipartimento di Elettronica Politecnico di Milano) Ph.D. Thesis
[11] Stützle T and Hoos H H 1997 *Proc. of 1997 IEEE International Conference on Evolutionary Computation (ICEC)* April 1997 Indianapolis (Piscataway: IEEE) 309–14
[12] Dorigo M and Gambardella L M 1997 *IEEE Trans. Evol. Comput.* **1** 53–66
[13] Dorigo M and Stützel T 2004 *Ant Colony Optimization* (Cambridge: The MIT Press)
[14] Laporte G 1992 *Eur. J. Oper. Res.* **59** 231–47