An Ant Colony Optimization algorithm for solving the fixed destination multi-depot multiple traveling salesman problem with non-random parameters

Cite as: AIP Conference Proceedings **1862**, 030123 (2017); https://doi.org/10.1063/1.4991227
Published Online: 10 July 2017

T. Ramadhani, G. F. Hertono and B. D. Handari

An Ant Colony Optimization Algorithm for Solving the Fixed Destination Multi-depot Multiple Traveling Salesman Problem with Non-random Parameters

T. Ramadhani, G. F. Hertono, and B. D. Handari[a)]

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

[a)]Corresponding author:bevina@sci.ui.ac.id

Abstract. The Multiple Traveling Salesman Problem (MTSP) is the extension of the Traveling Salesman Problem (TSP) in which the shortest routes of m salesmen all of which start and finish in a single city (depot) will be determined. If there is more than one depot and salesmen start from and return to the same depot, then the problem is called Fixed Destination Multi-depot Multiple Traveling Salesman Problem (MMTSP). In this paper, MMTSP will be solved using the Ant Colony Optimization (ACO) algorithm. ACO is a metaheuristic optimization algorithm which is derived from the behavior of ants in finding the shortest route(s) from the anthill to a form of nourishment. In solving the MMTSP, the algorithm is observed with respect to different chosen cities as depots and non-randomly three parameters of MMTSP*: m, K, L*, those represents the number of salesmen, the fewest cities that must be visited by a salesman, and the most number of cities that can be visited by a salesman, respectively. The implementation is observed with four dataset from TSPLIB. The results show that the different chosen cities as depots and the three parameters of MMTSP, in which m is the most important parameter, affect the solution.

INTRODUCTION

TSP is a problem to find the most efficient means of visiting every node and then returning to the starting node [1]. Many works have made concerning TSP both in an exact or metaheuristics way, in which, Branch and Bound [2], Genetic Algorithm [3], Ant Colony Optimization [4] and also Simulated Annealing [5].

The extension of TSP is MTSP. The problem is best described as finding a set of tours for m salesmen who all start and finish in a single city (depot) [6]. All of which start and many works have been done concerning MTSP both in an exact or metaheuristic way, in which, Lagrangian Relaxation [7], Branch and Bound [8], Particle Swarm Optimization [9], Simulated Annealing [10], Genetic Algorithm [11], and Ant Colony Optimization [12].

MMTSP has higher complexity since the salesmen depart from multiple depots instead of the same depot. There are two types of MMTSP: fixed destination MMTSP, in which every salesman has to end his route at his starting point, and non-fixed destination MMTSP otherwise. [13]. Unlike the two previous problems, few works have been made on the MMTSP before, so that is the reason why we investigate the problem. Previous works have been made before about Integer Linear Programming [14], Branch and Cut [15], Genetic Algorithm [16], Firefly Algorithm [17] and Ant Colony Optimization [13]. In this paper we will investigate the fixed destination MMTSP, using Ant Colony Optimization (ACO) based on [13] work.

Ghafurian and Javadian's previous work concluded that ACO is efficient in solving the MMTSP compared with solution produced with Lingo 8.0. They generate the problem, cities as depots, and the MMTSP parameters in randomly for four different size problems i.e. $n \in \{10,20,30,40\}$. In our work, we will further examine the analysis

International Symposium on Current Progress in Mathematics and Sciences 2016 (ISCPMS 2016)
AIP Conf. Proc. 1862, 030123-1–030123-7; doi: 10.1063/1.4991227
Published by AIP Publishing. 978-0-7354-1536-2/$30.00

on choosing a non-random parameters of MMTSP and choosing different cities as depots. Since there was no benchmark test for the MMTSP, for convenience we use several problems with the dataset taken from TSPLIB.

FIXED DESTINATION MMTSP

When multiple salesmen leave several starting cities (depots) then return to the starting city to create a tour where every city is visited only by a salesman, the problem is considered to be an MMTSP [13]. The objective of MMTSP is to find the shortest path done by m salesmen. Illustration of MMTSP given in Fig. 1.

In this paper, we adopted a mathematical model MMTSP from [14]. Let a complete graph $G=(V,A)$ represents n cities in which V is a set of nodes, while a set of edges constitutes A. Define $C=[c_{ij}]$ as a cost matrix for each arc $(i,j)\in A$. Let V be decomposed such that $V=V'\cup D$, where a depot set D consists of d first cities of V and $V'=\{d+1,d+2,\dots,n\}$. For each depot, there are m_k salesmen such that m is the total of all salesman. For each salesman, u_i is the number of cities that have been visited up to the i-th city. Define L as the maximum number of cities that a salesman can visit and K as the minimum number of cities a salesman must visit. The mathematical model for MMTSP is given by the following [13]:

$$\text{Min} \quad \sum_{k\in D}\sum_{j\in V'}\left(c_{kj}x_{kjk}+c_{jk}x_{jkk}\right)+\sum_{k\in D}\sum_{i\in V'}\sum_{j\in V'}c_{ij}x_{ijk} \tag{1}$$

This is subject to:

$$\sum_{j\in V'}x_{kjk}=m_k\ ,\ k\in D \tag{2}$$

$$\sum_{k\in D}x_{kjk}+\sum_{k\in D}\sum_{i\in V'}x_{ijk}=1\ ,\ j\in V' \tag{3}$$

$$x_{kjk}+\sum_{i\in V'}x_{ijk}-x_{jkk}-\sum_{i\in V'}x_{jik}=0\ ,\ k\in D\ ,\ j\in V' \tag{4}$$

$$\sum_{j\in V'}x_{kjk}-\sum_{j\in V'}x_{jkk}=0\ ,\ k\in D \tag{5}$$

$$u_i+(L-2)\sum_{k\in D}x_{kik}-\sum_{k\in D}x_{ikk}\le L-1\ ,\ i\in V' \tag{6}$$

$$u_i+\sum_{k\in D}x_{kik}+(2-K)\sum_{j\in V'}x_{ikk}\ge 2\ ,\ i\in V' \tag{7}$$

$$\sum_{k\in D}x_{kik}+\sum_{k\in D}x_{ikk}\le 1\ ,\ i\in V' \tag{8}$$

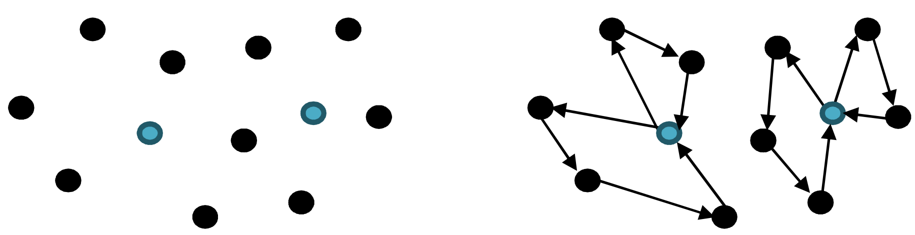
$$u_i-u_j+L\sum_{k\in D}x_{ijk}+(L-2)\sum_{k\in D}x_{jik}\le L-1\ ,\ i\ne j\ ,\ i,j\in V' \tag{9}$$

$$x_{ijk}\in\{0,1\} \tag{10}$$

Details about the constraints, see [13].

ANT COLONY OPTIMIZATION

Metaheuristics is a new generation of the heuristic algorithm. This method has been widely developed and used respect for increasing the complexity of combinatorial problem. One of many metaheuristic algorithm is ACO which is reminiscent of the behavior of ants in finding the shortest route(s) from the anthill to a form of nourishment. ACO algorithm was initially developed by [18] to solve TSP by considering the salesmen as artificial ants. In the real



(a) (b)

FIGURE 1. (a) The example of MMTSP with 12 cities and 2 depots. (b) the route formed from 12 cities

world, the ants choose the path to the food source randomly. The movement of ants is also influenced by the intensity of pheromones contained on the path that will be passed through. To solve MMTSP using ACO algorithm, a salesman is represented by an artificial ant who will travel until it returns to the depot [13]. The process of finding MMTSP solution by using ACO algorithm consists of three main processes: parameter initialization, transition probability, and pheromone update.

Parameter Initialization

The process of finding MMTSP solutions begins with parameter initialization based on [13], in which contains of d depot and m salesmen. Those salesmen have to visit a minimum number of cities (K), and at most L cities that are determined as follows:

1. The amount of depot (d) is determined by the number of cities divided by 10, $d = n/10$.
2. The minimum number of cities that salesmen must visit (K) is determined by the interval $2 \leq K \leq (n - d)/d$.
3. The maximum number of cities that salesmen can visit (L) is determined by the interval $(n - d)/m \leq L \leq n - d$.
4. The number of salesman (m) is determined by the interval $d \leq m \leq (n - d)/K$.

By considering the problem, all of the criteria are calculated to the greater integer value (ceiling). The initial positions of salesmen is distributed evenly on each depot. In this paper, we will choose and observe each parameter value at the lower, middle, and upper bounds. For instance with $n = 76$, the interval for K given by $2 \leq K \leq 9$. Firstly, we choose the lower bound value $K = 2$ which will give an interval by $8 \leq m \leq 34$. After that, we choose $m = 8$ and give the interval for L given by $9 \leq L \leq 68$. When we choose K, it will gives interval and three possible value of m and then we choose m which will give an interval and three possible values of L. Therefore, we have 27 combinations: for K = 2 we have Min = 30, Mid = 83, Max = 135, for K = 6 we have Min = 30, Mid = 38, Max = 45 and for K = 9 we have Min = 30, Mid = 30, Max = 30. Hence, we consider 27 observations for each problem.

Transition Probability

The second process in finding MMTSP solution is transition probability. Determination of next visit for each ant (salesman) is done by choosing the cities which have not been visited. As well as [18], we also define the set of $tabu_k$, in which the set of cities that has been visited by the k-th ant so that the city will only be visited by exactly one ant. The ant will choose the next city by using the following equation [13]:

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{n \in allowed_k} [\tau_{in}(t)]^\alpha [\eta_{in}]^\beta} & , \quad j \in allowed_k \,, \\ 0 & , \quad \text{others}, \end{cases} \tag{11}$$

where $allowed_k = \{n - tabu_k\}$ is a set of cities that can be visited by the k-th ant. The $\tau_{ij}(t)$ represents the amount of pheromone on arc (i, j) at t^{th} time. The η_{ij} represents heuristic value on each arc (i, j) which determined by $\eta_{ij} = 1/c_{ij}$. The α and β parameters represent how the ants will consider the influence of pheromone and heuristic value to the next city selections respectively. In order to obtain feasible results, we need to employ some conditions on the algorithm for transition probability taken from [13]. This condition is also intended to manage the salesmen so they can return to the starting depot.

Pheromone Updates

After all cities have been visited and ants have returned to the depot, the next process is updating pheromone value based on the ant tours. It aims to provide information about the route that has been done by the previous ants. The pheromone update is done using the following equation [13]:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij} \tag{12}$$

The $\tau_{ij}(t+1)$ value is the new pheromone value that has been updated and it will be used for the next route selection process. The ρ parameter is coefficient of evaporation which has the value $\rho \in (0,1)$, such that $(1 - \rho)$ is the evaporation rate. The $\Delta\tau_{ij}$ value represents about how much the intensity of pheromone laid by previous ants on arc (i,j) and obtained by the equation:

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{13}$$

where $\Delta\tau_{ij}^{k}$ is the amount of pheromone laid by the k^{th} on arc (i,j). In other words, $\Delta\tau_{ij}$ is the total amount of *pheromone* laid by the ants who have passed arc (i,j) and determined with the following equation:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k} & , \quad \text{if ant } k \text{ travels on arc } (i,j)\,, \\ 0 & , \quad \text{others}\,, \end{cases} \tag{14}$$

where Q is a constant i.e. $Q \in \{1,100,1000\}$[10] and L_k is the k^{th} ant's total distance upon returning to the depot. Based on the explanations before, here is the algorithm to solve MMTSP using ACO:

```
Initialize the cities' coordinates, cities as depot, ACO and MMTSP
parameters.
Distribute salesman evenly on each depot.
while the process has not met the termination criteria, do
     for each depot, do
          for each salesman on depot, do
              Transition probability process until return to depot
          end for
     end for
     Compute the salesmen total distance
     for each arc, do
          Pheromone update
     end for
end while
Return the best solution found
```

EXPERIMENTAL RESULTS

In this section, we will show the experimental result based on the algorithm given in the previous section. We run the implementation in Core™ i5-4200U CPU @1.6GHz processor with 4096MB RAM and coded with MATLAB 2015a. For ACO parameters, we used the same value as the [13] for α, β, ρ, and the initial value for pheromone τ_{ij} and also we choose $Q = 1$. For every problem, selection of depots will be done in two ways which is randomly and with Round Robin scheduling.

The previous section, we can see that the MMTSP parameters are connected to each other and resulting in 27 observations for each problem. For each observation, we run 10 trials and for each trial we run the implementation for 1000 iterations, and then take the best, average, and worst solutions respectively. The implementation results for $K = 2$ is given by the following tables (Table 1 to Table 4). The first row denotes a minimum number of parameters from the interval, and so on.

As can be seen from the experimental results, we only show the result for $K = 2$ since we can see that the best solution is given when m is minimum and the worst solution is given when m is maximum. We are not showing the other K value because when K is in the middle of the interval, there were no specific trends or patterns of solutions so that we cannot take any conclusion that parameter affect the solution. Meanwhile when K is at its maximum, it gives 9 repeated observations with the same values of m and L, but resulting in different solution produced for each observation. For instance, when $K = 9$ is maximum, it gives the same value $m = 8$ for lower, middle, and upper bounds. This value results in repeated values for L for lower bound, middle, and upper bounds. For K maximum we found the interesting part, in which, even for the same MMTSP parameters, we obtained a different solution. These results show that the solution produced for the same MMTSP parameters does not give the same solution.

For the next implementation, we will choose the depot in a different way by considering our problem as MTSP (single depot). Firstly we will use round robin scheduling method to find the set of depots which gives the best solution which is inspired by [19]. We use the minimum value of m since it gives the best solution in our previous implementation. We assign $K = 9$ and $L = 10$ so that the number of cities that every salesman can visit will be evenly distributed. This method will give a set of cities as a candidate of depots for our next implementation. The parameter details for this round robin scheduling method is given in Table 5.

After we obtain a set of cities a candidate for depots by round robin scheduling, we will have a set of cities as depots for every problem. As given by previous implementations, by random depot selection, we obtained the best solution when m is minimum. We select a different way by round robin scheduling in choosing cities as depots in hoping to obtain a better solution. The implementation results with depots produced by round robin scheduling are given in Tables 6 through Table 9.

As we can see from the implementation results given in Tables 6 through Table 9, produce the same characteristic solution given from Table 1 to Table 4. But we can see that the best solution is different from previous implementations where we can conclude that different cities as depots affect the solution. As well as previous implementations, we omit the middle value of K, because it also gives the same characteristics. For the maximum value of K gives the same characteristics as the previous implementation, which means we can conclude that the MMTSP solution produced by ACO is fluctuating in every iteration. We also find these results occur in TSP at [4].

TABLE 1. Implementation result of pr76 by randomly depots selection

m	L	Best	Worst	Average
	9	151,457	160,989	153,810
8	39	150,899	159,999	155,310
	68	**143,432**	157,170	150,660
	4	220,282	233,177	225,620
21	36	201,434	226,978	216,050
	68	208,012	220,556	215,400
	2	270,455	**288,900**	279,201
34	35	263,811	283,711	275,450
	68	266,357	277,346	272,030

TABLE 2. Implementation result of pr152 by randomly depots selection

m	L	Best	Worst	Average
	9	176,948	183,704	181,020
16	73	181,119	192,598	186,640
	136	**163,359**	180,995	172,610
	4	401,256	412,264	407,950
42	70	329,763	447,777	400,250
	136	397,403	439,729	422,890
	2	528,147	**539,853**	533,550
69	69	530,031	538,922	534,330
	136	532,101	536,994	534,640

TABLE 3. Implementation result of pr299 by randomly depots selection

m	L	Best	Worst	Average
	9	108,658	112,709	110,380
30	139	99,665	108,941	102,520
	269	**90,876**	103,274	98,432
	4	224,372	232,922	228,970
83	137	137,155	148,118	142,070
	269	131,240	144,694	139,420
	2	277,118	**283,015**	280,650
135	136	278,530	282,524	280,930
	269	278,247	282,402	131,240

TABLE 4. Implementation result of pr439 by randomly depots selection

m	L	Best	Worst	Average
	9	242,470	251,744	247,470
44	202	260,179	282,743	275,000
	395	**236,113**	267,961	255,580
	4	582,746	595,953	588,530
121	200	347,960	386,173	366,000
	395	351,236	377,362	363,080
	2	717,599	**727,821**	723,209
198	199	719,460	727,732	722,610
	395	716,459	725,643	722,760

TABLE 5. Details of parameter settings in round robin scheduling to determine the depot candidates

Problem	m	K	L	Number of Cities Taken as Depot Candidates
Pr76	8	9	10	8 cities as candidates
Pr152	16	9	10	16 cities as candidates
Pr299	30	9	10	30 cities as candidates
Pr439	44	9	10	44 cities as candidates

TABLE 6. Implementation result of pr76 by selecting the best cities as depots taken from Round Robin Scheduling

m	*L*	Best	Worst	Average
	9	166,068	172,661	169,242.7
8	39	160,437	177,868	167,032.5
	68	**152,557**	175,397	162,761
	4	301,014	310,466	306,407.9
21	36	275,143	282,676	279,027.3
	68	272,089	282,415	278,538.8
	2	393,644	407,770	402,620.7
34	35	401,166	408,456	404,712.3
	68	402,295	**411,418**	415,224.6

TABLE 7. Implementation result of pr152 by selecting the best cities as depots taken from Round Robin Scheduling

m	*L*	Best	Worst	Average
	9	228,258	258,524	247,738.5
16	73	203,825	272,217	230,436
	136	**186,297**	244,418	220,938.7
	4	514,095	544,163	528,724.9
42	70	411,441	487,062	458,573.7
	136	434,128	472,316	454,521
	2	739,546	759,073	746,340
69	69	732,952	757,416	748,451.2
	136	739,291	**763,181**	752,189

TABLE 8. Implementation result of pr299 by selecting the best cities as depots taken from Round Robin Scheduling

m	*L*	Best	Worst	Average
	9	152,070	155,195	153,640.6
30	139	111,586	125,731	120,320.2
	269	**102,452**	120,168	109,876.3
	4	373,323	376,895	375,054.4
83	137	272,434	282,673	278,078.4
	269	269,611	284,208	277,650.3
	2	523,308	**528,797**	525,837.5
135	136	524,209	528,614	526,583.7
	269	524,450	527,741	525,816

TABLE 9. Implementation result of pr439 by selecting the best cities as depots taken from Round Robin Scheduling

m	*L*	Best	Worst	Average
	9	366,588	376,014	371,221.6
44	202	238,632	271,858	255,863.3
	395	**216,701**	245,213	231,556.9
	4	962,099	972,280	966,482.9
121	200	521,265	556,915	537,433
	395	523,035	561,571	542,356.9
	2	1,275,132	1,285,104	1,279,863.4
198	199	1,276,098	1,283,775	1,281,051.3
	395	1,276,008	**1,286,387**	1,280,420.5

CONCLUSIONS

After we implement ACO to solve MMTSP with a non-random parameter and a different selection of depots, we can conclude that both the parameter and depot choice in solving MMTSP is very essential in producing a solution. As we can see from the implementation results, the best solution occurs when the number of salesmen is at a minimum. On the contrary, the worst solution occurs when the number of salesmen is at its maximum. We can also see that implementing the problem with different cities as depots by searching the depot candidates using round robin scheduling did not give a better solution. From this occurrence, we can conclude that choosing different cities as depots also affects the MMTSP solution produced. Hence, both the MMTSP parameter and different cities as depots affect the solution produced.

ACKNOWLEDGMENTS

Our work is fully granted and supported by Cluster Research 2015 Universitas Indonesia.

REFERENCES

1. D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The Traveling Salesman Problem: A Computational Study* (Princeton University Press, Princeton, New Jersey, 2006).
2. R. Wiener, J. Object Technol. **2**, 65–86 (2003).
3. Z. H. Ahmed, Int. J. Biom. Bioinformatics **3**, 96–105 (2010).
4. I. Brezina Jr. and Z. Čičková, Manag. Inf. Syst. **6**, 10–14 (2011).
5. X. Geng, Z. Chen, W. Yang, D. Shi, and K. Zhao, Appl. Soft. Comput. **11,** 3680–3689 (2011).
6. T. Bektas, Omega **34**, 209–219 (2006).

7. S. Yadlapalli, W. A. Malik, S. Darbha, and M. Pachter, Nonlinear Anal. Real World Appl. **10**, 1990–1999 (2009).
8. J. F. Cordeau, M. Iori, G. Laporte, and J. J. S. González, Networks **55**, 46–59 (2010).
9. S. Pang, T. Li, F. Dai, and M. Yu, Appl. Math. Inform. Sci. **7**, 2439–2444 (2013).
10. C. H. Song, K. Lee, and W. D. Lee, "Extended Simulated Annealing for Augmented TSP and Multi-Salesmen TSP," in *Proceedings of The International Joint Conference on Neural Networks*, Portland, 2003 (IEEE, New Jersey, 2003), Vol. 3, pp. 2340–2343.
11. V. Arya, A. Goyal, and V. Jaiswal, Int. J. Appl. Innov. Eng. Manag. **3**, 425–430 (2014).
12. P. Junjie and W. Dingwei, "An Ant Colony Optimization Algorithm for Multiple Traveling Salesmen Problem," in *Proceedings of The First International Conference on Innovative Computing, Information and Control*, Beijing, 2006 (IEEE, New Jersey, 2006), Vol. 1691778, pp. 210–213.
13. S. Ghafurian and N. Javadian, Appl. Soft. Comput, **11**, 1256–1262 (2011).
14. I. Kara and T. Bektas, Eur. J. Oper. Res. **174**, 1449–1458 (2006).
15. E. Benavent and A. Martínez, Ann. Oper. Res. **207**, 7–25 (2013).
16. K. Kardel, F. Adbesh, and N. Javadian, "A New Approach to Solving the Fixed Destination Multi-Depot Multiple Traveling Salesman Problem using Genetic Algorithms," in *Proceedings of The 7th International Conference on Optimization: Techniques and Applications*, Kobe, 2007 (German National Library of Science and Technology (TIB), Hannover, 2007), Vol. 8, pp. 73–79.
17. L. Kota and K. Jarmai, Adv. Logist. Syst. **7**, 95–102 (2013).
18. M. Dorigo, V. Maniezzo, and A. Colorni, IEEE Trans. Syst. Man. Cybern. B, Cybern. **26**, 1–13 (1996).
19. Karina, G. F. Hertono, and B. D. Handari, "Algoritma sweep dan elite ant system untuk menyelesaikan Multiple Traveling Salesman Problem (MTSP)," in *Proceedings of the Seminar Nasional Matematika*, Bandung, 2016 (Jurusan Matematika Fakultas Teknologi Informasi dan Sains Universitas Katolik Parahyangan, Bandung, 2016), Vol. 11, pp. 114–119.