西南交通大学学报

第 54 卷 第 3 期
2019 六 3 月

JOURNAL OF SOUTHWEST JIAOTONG UNIVERSITY

Vol.54 No.3
June 2019

ISSN - 0258-2724

DOI：10.35741/issn.0258-2724.54.3.16

Research article

SOLVING MULTIPLE TRAVELING SALESMAN PROBLEM BY MEERKAT SWARM OPTIMIZATION ALGORITHM

Belal Al-Khateeb [a], Mohammed Yousif [a]
[a]Department of Computer Science, University of Anbar, Ramadi, Anbar, Iraq, belal@computer-college.org, mohammed.alyousif1991@gmail.com

Abstract
Multiple Traveling Salesman Problem (MTSP) is one of various real-life applications, MTSP is the extension of the Traveling Salesman Problem (TSP). TSP focuses on searching of minimum or shortest path (traveling distance) to visit all cities by salesman, while the primary goal of MTSP is to find shortest path for m paths by n salesmen with minimized total cost. Wherever, total cost means the sum of distances of all salesmen. In this work, we proposed metaheuristic algorithm is called Meerkat Swarm Optimization (MSO) algorithm for solving MTSP and guarantee good quality solution in reasonable time for real-life problems. MSO is a metaheuristic optimization algorithm that is derived from the behavior of Meerkat in finding the shortest path. The implementation is done using many dataset from TSPLIB95. The results demonstrate that MSO in most results is better than another results that compared in average cost that means the MSO superior to other results of MTSP.

Keywords: Optimization, Multiple Traveling Salesman Problem, Meerkat Swarm Optimization Algorithm, NP-Hard Problems, Metaheuristic Algorithms.

摘要: 多旅行商問題（MTSP）是各種實際應用之一，MTSP 是旅行商問題（TSP）的延伸。 TSP 專注於搜索銷售人員訪問所有城市的最小或最短路徑（行進距離），而 MTSP 的主要目標是找到 n 個銷售人員以最小化的總成本找到 m 個路徑的最短路徑。 無論何處，總成本是指所有銷售人員的距離總和。 在這項工作中，我們提出了元啟發式算法，稱為 Meerkat Swarm Optimization（MSO）算法，用於求解 MTSP 並在合理的時間內保證良好的質量解決現實問題。 MSO 是一種元啟發式優化算法，它源於 Meerkat 在尋找最短路徑時的行為。 使用 TSPLIB95 中的許多數據集完成實現。 結果表明，大多數結果中的 MSO 優於平均成本比較的另一結果，這意味著 MSO 優於 MTSP 的其他結果。

关键词: 優化，多旅行商問題，Meerkat 群優化算法，NP-Hard 問題，元啟發式算法。

I. INTRODUCTION

TSP is typical optimization problems, TSP is defined as path searching problem for a salesman to visit all cities just once, begin the tour and finish in same city or depot and the main goal or primary objective of TSP is to minimize cost of salesman [1][2][3]

MTSP is improved or extension of TSP that are well-known for many real life problems. In MTSP, cities are divided into m salesmen by assigning the cities to a different salesman. In this work, MTSP has been studied, metaheuristic algorithm has been proposed to solve MTSP and find the shortest path and the results of the algorithm have been checked using dataset [4]. The rest of the paper is organized as follows: The literature review of MTSP are mentioned in Section II. MTSP together with its areas and former studies are mentioned in Section III. MSO metaheuristic algorithm that is proposed for the solution of MTSP is presented in Section IV. The results of MTSP showed in Section V. Finally, conclusions future works are presented in Section VI.

II. Literature Review

Xiaobin Wang et.al [5], proposed a method brand new that relies on the knowledge of Graph Theory to resolve a heterogeneity of multiple depots MTSP and open ways. An easy model (SModel) is introduced to implement the multiple depots MTSP and open paths by remodeling a complicate graph into a simplified graph with a tiny number of edges. Throughout the operation of generating easy model (SModel), high weighted edges are removed preferentially as possible. Since mdop is enforced supported on the SModel, it is secure that the ultimate result is superior. By the experimental results, it’s shown that the new solution is efficient.

L. Kota et.al [6], proposed the general model of the technical inspection and maintenance systems are shown in the first part, wherever the solution to this problem is a crucial question. A mathematical model of the system’s object skilled assignment is projected with the constraints typical of the system, like experts’ capacity minimum and maximum and constraints on maximum and daily tours of the experts. In the second part, the improved evolutionary programming algorithm is described that solves the assignment, regarding the constraints introducing penalty functions in the algorithm. In the last part, the convergence of the algorithm and therefore the run times and few examination of the parallelization is presented.

T. Ramadhani et.al [7], solved MTSP using the Ant Colony Optimization (ACO) algorithm. In solving the MTSP, Ant Colony Optimization (ACO) is implemented with respect to completely different chosen cities as depots. There are three parameters of MTSP that are considered in the implementation, those are the number of salesmen, the fewest cities that must be visited by a salesman and the most variety of cities which will be visited by a salesman frequently. The implementation is done using four datasets from TSPLIB. The results show that the various chosen cities are as depots and it was found that the number of salesmen is the most important parameter, which have an effect on the solution.

A. Steven et.al [8], performed cluster of any nodes traversed, permitting MTSP to be simplified to an MTSP (multiple traveling salesman problem) or a TSP for every cluster. The clustering algorithms that may be used are agglomerative clustering and K-means clustering, whereas ant colony optimization is used when determining the shortest route for every cluster. The solution to MTSP is calculated from the total of the shortest routes for these clusters. They used data samples from TSPLIB for our implementation. The results of the simulation show that agglomerative clustering ACO algorithms take longer to reason than K-means clustering ACO and standalone ACO algorithms; on the other hand, they yield superior results than the other two. These results also are compared with results obtained from previous researches.

Al-Khateeb [9], solved the Multiple Traveling Salesman Problem (MTSP) by Particle Swarm Optimization (PSO). The work focused on finding the best acceleration factor at which many selected values are tested for these factors. The obtained results demonstrated that the factors are problem dependent.

Kin-Ming Lo et.al [3], proposed a novel effective Genetic Algorithm with Local Operators (GAL) to solved Multiple Traveling Salesman Problem (MTSP) and generate higher quality solution in reasonable time. Wherever, used two local operators, Branch and Bound (BaB) and Cross Elimination (CE), according to the results the GAL have been successfully deployed to generate higher quality results. They algorithm had made improvement in the search ability and speed.

III. The Multiple Traveling Salesman Problem

MTSP is grown or developed from TSP. MTSP is completely different from TSP, as in MTSP there are m salesmen, each depot during a given cluster or group of n cities is divided or split into m tours by distribution each one of these depots to a distinct salesman. The target is to seek out the minimum value of cost of the tours in total. The value will be referred as time or distance [3].

The MTSP is outlined on a graph G = (V, A), where A represents the set of edges and V referred the set of vertices. Let C = (Cij) be the cost matrix defined on the group of A. If Cij = Cji then the cost matrix is symmetric, otherwise it is asymmetric. If the cost matrix satisfies Cij ≤ Cik + Ckj for ∀i, j, k, then the matrix C satisfies the triangle inequality [10][11].

Assignment based mathematical model is one of the most proposed models for MTSP, therefore tree based mathematical model and a three-index row-based model have been common used [12].

The three-index row-based model for the MTSP is as follows: Let n be the number of cities to be visited, and m be the number of salesmen (we assume n ≥ 3m+1). Then the variable xij is defined as follows [1][12]:

$$x_{ij} = \begin{cases} 1, & if\ edge(i,j) is\ used\ in\ tour, \\ 0, & otherwise. \end{cases}$$

Goal function:

minimize $\sum_{(i,j)\in A} c_{ij} x_{ij}$ (1)

Constraints:

$$\sum_{j=2}^{n} x_{1j} = m \quad (2)$$

$$\sum_{j=2}^{n} x_{j1} = m \quad (3)$$

$$\sum_{i=1}^{n} x_{ij} = 1, j=2,\dots,n \quad (4)$$

$$\sum_{j=1}^{n} x_{ij} = 1,\ i = 2,\ \dots,\ n \quad (5)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1,\ \forall S\ V - 1,\ S \neq 0 \quad (6)$$

$$x_{ij} = 0 \vee 1,\ (i,j) \in A \quad (7)$$

In this model, constraints (4), (5) and (7) satisfy the assignment problem constraints. Constraints (2) and (3) ensure the comeback of each salesman to his starting point. Constraint (6) is used to prevent sub-tours [13]. The models and solutions that are used for multi depot are also can be used for MTSP [12] [14].

IV. Meerkat Swarm Optimization for MTSP

A. Meerkat Swarm Optimization

Meerkat also called (Suricata suricata) are social animals. Meerkats live a life, which is 80%, based on teamwork. Meerkats live in community groups called mobs. Every mob has its own leaders (named alpha) and its own territory. The Meerkat mob leaves their burrows searching for food every morning, the mob is divided into two sub-groups, one for foraging while other stays as a babysitter for pups in burrows. The MSO algorithm simulates the cooperative behavior of meerkats during the search for food as shown in algorithm 1.

Algorithm1: Meerkat Swarm Optimization.
Initialize Mob of Meerkats **n** members. Calculate the fitness of each search agent. Alpha=the best search agent. t=1. **While** (t < Max number of iterations) Divide Mob into two subgroups, Foraging group and Babysitter group. Update Hungry rate and Position of Foraging group. Update Hungry rate and Position of Babysitter group. Decrease Hungry rate for Foraging group with rate. Merge the two groups into Mob and decrease Hungry rate for all with rate. Calculate the fitness of each search agent. Select best one in Mob as Alpha. t=t+1. **End while** Return Alpha.

B. Initialization

The important stage in the operations of solution is the initialization process which provides the algorithm needs as well as the data of the problem and submit it. The preparedness phase consists of number of stages.

The first phase is the process of reading the problem database information. The form of problem information is graphic points. Each city has two points one of them on the x-axis and the other on the y-axis. After reading dataset, the cities are distributed to groups, each salesperson in the same group selects cities called depot to start and finish, the groups don't have the same number of crows. By these points (x-axis and y-axis), distances between each city and other cities can be calculated. This is done through the following equation:

$$\textbf{distance (i, j)} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

(1)

Where x_i and y_i represent a point on the x-axis and y-axis, respectively, for the city i. In addition, x_j and y_j represent a point on the x-axis and y-axis, respectively, for the city j. Equation (1) is repeatedly used until the distances between all cities are calculated. The output of this stage is two-dimensional array that contains values, which represent the distances between cities. In this stage, the number of cities that salesmen have visited is known, as well.

After knowing the number of cities of the problem and number of cities of all groups, as well as the distances between each city, it becomes possible to calculate the value of the initial Hr and Rate for all meerkat. It should be noted that in this step Hr value is between (0, 1) and Rate value is between (0, last calculated rate). This procedure mimics the situation in real meerkat. This case represents the first movements of meerkat to search for the source of the food.

When selecting any path, meerkats can receive quantities of food. These routes do not necessarily lead to the feed source. Therefore, the food during this case is a guide that works on the ways that are taken by crows and not necessarily the food path.

C. MSO Solution Construction

A solution can be constructed using hungry rate, fitness and position. After the initialization step, Meerkats divided into two mob, after that, MSO algorithm starts to work. Meerkats start to move from the beginning node that had chosen in the initialization stage. Mrows move from one node (start depot) to another until reaching to the target node (finish depot).

After determining the destination depot or node (city) and salesman traveling there to, the position value is updated according to hungry rate and current position. After that, the hungry rate is updated. When the Meerkat in all mobs complete all their roads, the cost (fitness) is calculated. This helps the Meerkats attempt to get away from the roads.

The following algorithm represents the MSO algorithm for MTSP.

Algorithm 2: Meerkat Swarm Optimization Algorithm for MTSP

Initialization
No_tours
No_Cluster
INPUTE: Get the dataset information (points)
-Set initial hungry and position for every salesman (Meerkat).
Number of salesmen (Merkats) = *No_Cluster* * No_tours.
-Set the number of cities for each salesman in every *Cluster*. Each salesman will get the same number of cities in same *Cluster*.
-Compute the distances between cities in every Cluster according to equation (no.1).
-Set start and finish city for each Cluster**.**
-Max number of iterations.
Solution
While (t < Max number of iterations)
-Calculate the cost for every Cluster
-divided the Cluster into two mobs
- Update the position of the salesman in first mob and second mob.
- merge the two mobs in all cluster into once tour.
- Calculate the cost for each tour.
End while.
Best Tour= the best tour from all Cluster.
OUTPUT: Best Tour.

V. Results and Discussion

In this paper, we used data from the dataset called "TSPLIB" in order to solve MTSP, the data is specifically selected from the categories of Pr dataset (Pr76, Pr152, Pr299, Pr439 and Pr1002), where the depots are selected randomly. Table 1, shows the results of Pr76, Pr152, Pr299 and Pr439, the following settings are used: 10 runs each run is with 1000 iterations, the depot for Pr76 are 8, 21 and 34, while the depot for dataset Pr152 are 16, 42 and 69, the dataset Pr299 have 30,83 and 135 as a depot, lastly, the dataset Pr439 selects 44, 121 and 198 as a depot. The reason behind choosing more than one depot for each dataset is to know the best number of depots to be suitable for the dataset.

In order to measure the efficiency and strength of the MSO algorithm, it will be compared with ACO algorithm, Round Robin (RR) algorithm [7] and K-Means Clustering algorithm [2]. The experimental results are shown in tables 1 and 2, respectively. Four dataset and different number of depots are used to evaluate the performance. Since the problem is not the optimal solution was obtained at a specific time and depending on the sources the comparative results were determined number of iterations, number of populations and number of depot.

Table 1.

Results of Pr76, Pr152, Pr299 and Pr439 by ACO and MSO Algorithm.

Name	Iteration	Depot	ACO [7]	RR [7]	MSO
			Ave	Ave	Ave
Pr76	1000	8	150660	162761	**133436**
		21	216050	278538.8	**119927**
		34	275450	402620.7	**91122**
Pr152		16	180995	220938.7	**126147.6**
		42	400250	458573.7	**107693**
		69	533550	748451.2	**57100**
Pr299		30	98432	109876.3	**68414.8**
		83	139420	277650.3	**56972**
		135	280650	525837.5	**40573**
Pr439		44	255580	**231556.9**	236327
		121	366000	537433	**223759**
		198	723209	1279863.4	**177594**

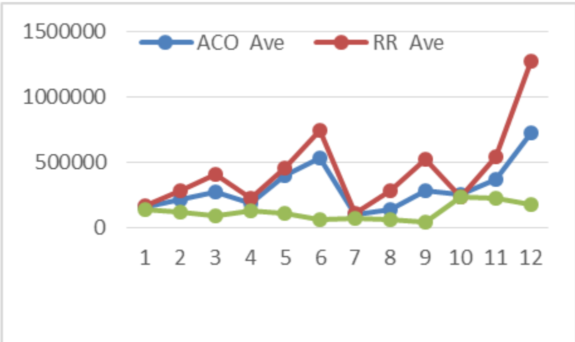

Figure 1: Results of Pr76, Pr152, Pr299 and Pr439 by ACO and MSO Algorithm.

Table 1, shows the best cost, the worst cost and the average cost for MSO and ACO. The obtained results show the superiority of MSO algorithm as MSO gives better results than ACO. This gives a good indication that MSO is better than ACO, which reflects a great success for

MSO in solving the MTSP problem. While, the results of the comparison between RR algorithm and MSO algorithm. The proposed algorithm is very much superior to the RR algorithm, where the comparison is based on 10 runs. The performance of the dataset Pr76, Pr152 and Pr299 was superior in all the results and cases. In Pr439, the algorithm excels in all the results except in the case of 44 depots. This gives indication that MSO is better than RR, which reflects an excellent success for MSO to solving the MTSP problem.

Also, we compared the results of this algorithm with **K-Means Clustering** algorithm in [8]. The experimental results are shown in table 2. Four dataset and different number of depot are designed to measure evaluate this algorithm.

Table 2.
Results of Pr76, Pr152, Pr299 and Pr439 by K-Means Clustering and MSO Algorithm.

Name	Population	Iteration	Depot	K-Means Clustering [8]	MSO
				Best	Best
Pr76	50	50	8	**126590**	131748
	100	100		**118194**	130669
Pr152	50	50	16	**51494,15**	124205
	100	100		**51489.61**	123255
Pr299	50	50	30	**56162.94**	68126
	100	100		**54946.46**	67356
Pr439	50	50	44	**111857,20**	248964
	100	100		**109148.41**	233870

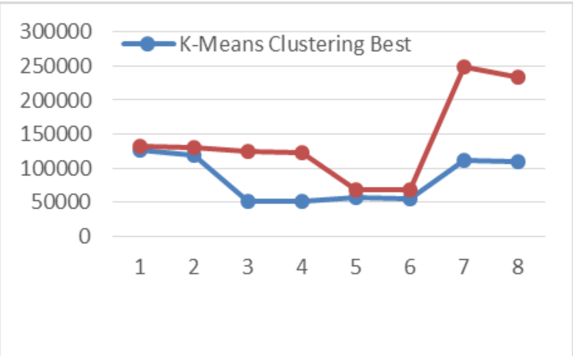

Figure 2: Results of Pr76, Pr152, Pr299 and Pr439 by K-Means Clustering and MSO Algorithm.

The results in table 2 show that the results of MSO algorithm are worse than the **K-Means Clustering** algorithm. This is because the clustering concept in MTSP is very useful and improves the quality of the results.

As well, MSO algorithm is compared with ACO algorithm [5]. The experimental results are shown in table 3. Five datasets and five depots are designed to evaluate the performance.

Table 3.
Results of Pr76, Pr152, Pr226, Pr439 and Pr1002 by ACO and MSO Algorithm.

Name			ACO [5]		MSO	
	Depot	Population	Ave	Time(s)	Ave	Time(s)
Pr76	5	20	180690	51	**138192**	23
Pr152	5	40	136341	128	**136151.3**	47
Pr226	5	50	170877	143	**88646**	60
Pr439	5	100	**165035**	563	256015	180
Pr1002	5	220	387205	2620	**328702**	500

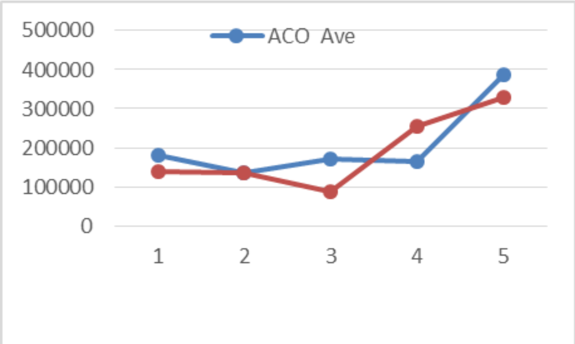

Figure 3: Results of Pr76, Pr152, Pr226, Pr439 and Pr1002 by ACO and MSO Algorithm.

The results in table 3 show that MSO is better than ACO in Pr76, Pr226 and Pr1002 with less execution time. However, in Pr152 and Pr439 by ACO is better than MSO, but MSO still managed to have a less execution time.

Also, MSO algorithm is compared with PSO algorithm [9]. The experimental results are shown in table 4. Five datasets and five depots are selected to evaluate the performance.

Table 4.
Results of Pr76, Pr152, Pr299, Pr439 by PSO and MSO Algorithm.

Problem	Iterations	Depot	PSO [9]	MSO
Pr76	1000	5	291493.6	**138192**
Pr152			440458.4	**137851.3**
Pr299			288484	**75824.5**

Pr439			727180.8	**256015.3**

The results in table 4 show that MSO algorithm is superior to the PSO algorithm, where the comparison is based on 10 runs. This gives indication that MSO is better than PSO, which reflects an excellent success for MSO in solving the MTSP problem.

Finally, results of MSO algorithm is compared with None, BaB, EC and CE+BaB [3]. This experimental results are shown in table 5. Six datasets and different depots (5, 10 and 15) are selected to evaluate the performance.

Table 5.

Results of Pr76, Pr152, Pr226, Pr299, Pr439, Pr1002 by None, BaB, EC, CE+BaB and MSO Algorithm.

Number of Salesmen		**5**		**10**		**15**	
Name	**Operator**	**Ave**	**Time**	**Ave**	**Time**	**Ave**	**Time**
Pr76	None[3]	168840	4.20	220259	5.02	272736	5.30
	BaB[3]	167697	6.60	225023	7.66	272145	7.80
	CE[3]	164661	4.90	186317	6.65	226224	6.54
	CE+BaB[3]	166138	6.80	182381	8.29	223927	8.77
	MSO	**138192**	5.70	**130035**	7.00	**119120**	8.25
Pr152	None	138088	16.70	228736	19.10	289744	19.13
	BaB	**131109**	27.00	234944	27.80	304915	28.11
	CE	132395	19.40	136228	22.09	164741	21.27
	CE+BaB	131674	28.00	141993	32.14	164321	29.49
	MSO	137851	22.45	**133005**	29.00	**124857**	30.10
Pr226	None	165893	46.70	247699	51.95	324268	51.12
	BaB	155574	77.00	251155	81.43	340139	81.81
	CE	157120	52.60	172193	58.84	188813	58.74
	CE+BaB	156629	75.60	171338	85.43	188489	82.13
	MSO	**88646**	60.25	**87982**	77.48	**84069**	84.40
Pr299	None	78872	78.00	121429	82.01	163144	83.17
	BaB	77676	120.90	121983	117.67	160755	123.88
	CE	78217	86.90	81323	95.71	87490	101.94
	CE+BaB	77413	123.10	78999	132.31	88526	134.14
	MSO	**75824**	115.48	**73449**	100.67	**72166**	109.84
Pr439	None	152224	184.70	209376	185.17	270185	209.93
	BaB	146436	300.00	207095	283.39	167122	279.41
	CE	148416	215.20	151636	211.26	159609	211.58
	CE+BaB	147389	295.70	**151392**	301.25	**155512**	315.70
	MSO	**256015**	255.47	254471	280.16	244759	300.00
Pr1002	None	375882	793.70	449855	800.00	550605	812.93
	BaB	348712	1154.80	435109	1182.24	535035	1149.55
	CE	349258	864.30	371649	894.75	393283	888.89
	CE+BaB	338580	1224.50	360284	1298.28	383360	1240.62
	MSO	**328702**	900.43	**327288**	1005.48	**325505**	1107.84

The results in table 5 show that MSO algorithm is superior in performance with None, BaB, EC and CE+BaB, where the comparison is based on 10 runs. This gives indication that MSO is better than None, BaB, EC and CE+BaB, which reflects an excellent success for MSO in solving the MTSP problem.

The obtained results in tables 1 thru 5 give a good indication that MSO is a very good algorithm in solving TSP like problems, as it was very good in solving MTSP.

VI. Conclusions and Future Work

The MSO is used to solve MTSP; according to the results in table 1 thru 5, the obtained results demonstrate that MSO is a good algorithm in solving MTSP. In MTSP the obtained solution can be enhanced when the number of salesmen is increased this is due to the increase of the solutions in the search space. Choosing different cities as a depot also affects the MTSP solution quality, as it leads to a better cost. Future work, the success of MSO algorithm to solve the MTSP can be enhanced by using clustering in MTSP, applied it on another problem [15][16][17].

References

[1] M. Yousif and B. Al-Khateeb, "A Novel Metaheuristic Algorithm for Multiple Traveling Salesman Problem," Adv. Res. Dyn. Control Syst., vol. 10, no. 13, pp. 2113–2122, 2018.

[2] D. Lawler, E., Lenstra, J., Kan, A., and Shmoys, THE TRAVELING SALESMAN PROBLEM : A GUIDED TOUR OF COMBINATORIAL OPTIMIZATION. 1985.

[3] K. Lo, W. Y. P. Wong, K. L. Yee, and L. S. Mak, "A Genetic Algorithm with New Local Operators for Multiple

Traveling Salesman Problems," Int. J. Comput. Intell. Syst., vol. 11, pp. 692–705, 2018.
[4] T. B. Imdat Kara, "Integer linear programming formulations of multiple salesman problems and its variations," Eur. J. Oper. Res., vol. 174, no. 3, pp. 1449–1459, 2006.
[5] X. Wang, D. Liu, and M. Hou, "A Novel Method for Multiple Depot and Open Paths , Multiple Traveling Salesmen Problem," IEEE 11th Int. Symp. Appl. Mach. Intell. Informatics, pp. 187–192, 2013.
[6] Benjamin Durakovic, Muris Torlak, "Simulation and experimental validation of phase change material and water used as heat storage medium in window applications", J. of Mater. and Environ. Sci., Vol. 8, No. 5, (2017), ISSN: 2028-2508, pp 1837-1846.
[7] 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," Int. Symp. Curr. Prog. Math. Sci. 2016 (ISCPMS 2016), vol. 030123, 2017.
[8] Osamah Ibrahim Khalaf, Ghaida Muttashar Abdulsahib and Muayed Sadik, 2018. A Modified Algorithm for Improving Lifetime WSN. *Journal of Engineering and Applied Sciences, 13: 9277-9282*.
[9] B. Al-Khateeb, "The Selection of Particle Swarm Optimization Learning Factors Values in Solving the Multiple Travelling Salesman Problem," Adv. Res. Dyn. Control Syst., vol. 10, no. 7, pp. 439–445, 2018.
[10] Abd Ghani, M.K, Mohammed, M.A., Arunkumar, N. Mostafa SA, Ibrahim DA, Abdullah MK, Jaber, M.M., Enas Abdulhay, Gustavo Ramirez-Gonzalez, Burhanuddin, M.A.2018. Decision-level fusion scheme for nasopharyngeal carcinoma identification using machine learning techniques. Neural Computing and Applications. https://doi: 10.1007/s00521-018-3882-6.
[11] E. Kivelevitch, K. Cohen, and M. Kumar, "A Market-Based Solution to the Multiple Traveling Salesmen Problem A Market-based Solution to the Multiple Traveling Salesmen Problem," J. Intell. Robot. Syst. Manuscr. No., vol. 72, no. 1, pp. 21–40, 2013.
[12] Mostafa, S.A., Mustapha, A., Mohammed, M.A., Hamed, R.I., Arunkumar, N., Ghani, M.K.A., Jaber, M.M. and Khaleefah, S.H., 2019. Examining multiple feature evaluation and classification methods for improving the diagnosis of Parkinson's disease. Cognitive Systems Research, 54, pp.90-99.
[13] G. Dantzig, R. Fulkerson, and S. Johnson, "Journal of the Operations Research Society of America Solution of a Large-Scale Traveling-Salesman Problem," J. Oper. Res. Soc. Am., vol. 2, no. 4, pp. 393–410, 1954.
[14] F. Nuriyeva and G. Kizilates, "A NEW HEURISTIC ALGORITHM FOR MULTIPLE TRAVELING SALESMAN PROBLEM," TWMS J. Appl. Eng. Math., vol. 7, no. 1, pp. 101–109, 2017.
[15] Y. Mishenin, I. Koblianska, V. Medvid, and Y. Maistrenko, "Sustainable Regional Development Policy Formation: Role of Industrial Ecology and Logistics," *Enterpreneurship and Sustainability Issues*, vol. 6, no. 1, pp. 329–341, 2018.
[16] O. Chkalova, M. Efremova, V. Lezhnin, A. Polukhina, and M. Sheresheva, "Innovative Mechanism for Local Tourism System Management: A Case Study," *Entrepreneurship and Sustainability Issues*, vol. 6, no. 4, pp. 2052–2067, 2019.
[17] A. Shevyakova, E. Munsh, and M. Arystan, "Information Support for the Development of Tourism for the Diversification of the Economy of Kazakhstan," *Insights into Regional Development*, vol. 1, no. 2, pp. 138-154, 2019.

参考文:

[1] M. Yousif 和 B. Al-Khateeb，“一種用於多旅行商問題的新型元啟發式算法”，Adv。RES。達因。 Control Syst。，vol。 10，不。 13，pp. 2113-2122, 2018。
[2] D. Lawler，E.，Lenstra，J.，Kan，A。和 Shmoys，“旅行銷售員問題：組合優化的指導之旅”。 1985 年。
[3] K. Lo，W。Y. P. Wong，K。L. Yee 和 L. S. Mak，“A Genetic Algorithm with New Local Operators for Multiple Travel Salesman Problems，”Int。 J. Comput。 INTELL。 Syst。，vol。 11，pp.692-705, 2018。
[4] T. B. Imdat Kara，“多個推銷員問題及其變化的整數線性規劃公式”，Eur。 J. Oper。 Res。，vol。 174，沒有。 3，pp.1449-1459, 2006。
[5] X. Wang，D。Liu 和 M. Hou，“一種新的多倉庫和開放路徑的方法，多個旅行推銷員問題”，IEEE 11th Int。 SYMP。申請馬赫。 INTELL。信息學，第 187-192 頁，2013 年。
[6] Benjamin Durakovic，Muris Torlak，“相變材料和水在窗戶應用中用作蓄熱介質的模擬和實驗驗證”，J.of Mater。和環境。 Sci。，Vol。 8，No。5，（2017），ISSN：2028-2508，pp 1837-1846。
[7] T. Ramadhani，G。F. Hertono 和 B. D. Handari，“用於解決具有非隨機參數的固定目標多車場多旅行商問題的蟻群優化算法”，Int。 SYMP。 CURR。 PROG。數學。科學。 2016 年（ISCPMS 2016），第一卷。 030123, 2017。
[8] Osamah Ibrahim Khalaf，Ghaida Muttashar Abdulsahib 和 Muayed Sadik，2018。改進的生命週期 WSN 算法。 Journal of Engineering and Applied Sciences，13：9277-9282。
[9] B. Al-Khateeb，“解決多旅行商問題中粒子群優化學習因素值的選擇”，Adv。RES。達因。 Control Syst。，vol。 10，不。 7，pp. 439-445, 2018。
[10] Abd Ghani，M.K，Mohammed，M.A.，Arunkumar，N。Mostafa SA，Ibrahim DA，Abdullah MK，Jaber，M.M.，Enas Abdulhay，Gustavo Ramirez-Gonzalez，Burhanuddin，M.A.2018。利用機器學習技術識別鼻咽癌的決策級融合方案。神經計算與應用。 https：// doi：10.1007 / s00521-018-3882-6。
[11] E. Kivelevitch，K。Cohen 和 M. Kumar，“基於市場的解決多個旅行推銷員問題的問題基於市場的解決多個旅行商問題的問題”，J。Intell。機器人。 SYST。 Manuscr。不，第一卷 72，不。 1，pp. 21-40, 2013。
[12] Mostafa，S.A.，Mustapha，A.，Mohammed，M.A.，Hamed，R.I.，Arunkumar，N.，Ghani，M.K.A.，Jaber，M.M。和 Khaleefah，S.H.，2019。檢查多功能評估和分類方法，以改善帕金森病的診斷。 Cognitive Systems Research，54，pp.90-99。
[13] G. Dantzig，R。Fulkerson 和 S. Johnson，“美國運作研究學會期刊解決大規模旅行 - 推銷員問題”，J。Oper。 RES。 SOC。 Am。，vol。 2，沒有。 4，pp. 393-410, 1954。
[14] F. Nuriyeva 和 G. Kizilates，“用於多種旅行銷售員問題的新啟發式算法”，TWMS J. Appl。工程。數學，第一卷。 7，不。 1，pp. 101-109, 2017。
[15] Y. Mishenin，I。Koblianska，V。Medvid 和 Y. Maistrenko，“可持續區域發展政策的形成：工業生態和物流的作用”，企業家精神和可持續性問題，第一卷。 6，不。 1，pp. 329-341, 2018。
[16] O. Chkalova，M。Efremova，V。Lezhnin，A。Polukhina 和 M. Sheresheva，“地方旅遊系統管理的創新機制：案例研究”，“企業家精神和可持續性問題”，第一卷。 6，不。 4，pp. 2052-2067, 2019。
[17] A. Shevyakova，E。Munsh 和 M. Arystan，“為哈薩克斯坦經濟多樣化發展旅遊業提供信息支持”，“區域發展見解”，第一卷。 1，不。 2，pp. 138-154, 2019。