Kütahya Dumlupınar University
Institute of Graduate Studies

Journal of Scientific
Reports-A
E-ISSN: 2687-6167
Number 47, June 2021

RESEARCH ARTICLE

An APPLICATION of A MODIFIED CAMEL TRAVELING BEHAVIOR ALGORITHM for TRAVELING SALESMAN PROBLEM

Mehmet Fatih DEMİRAL[1]

[1]Burdur Mehmet Akif Ersoy University, Faculty of Engineering and Architecture, Department of Industrial Engineering, mfdemiral@mehmetakif.edu.tr, ORCID:0000-0003-0742-0633

Received Date:22.03.2021 *Accepted Date:26.08.2021*

ABSTRACT

Camel Traveling Behavior Algorithm (CA) is a fairly new algorithm developed in 2016 by Mohammed Khalid Ibrahim and Ramzy Salim Ali. Scientists have put forward a few publications on CA. CA was applied to continuous optimization problems and engineering problems in the literature. It has been shown that CA has comparable performance with Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). Besides, a modified camel algorithm (MCA) has been implemented in the field of engineering and was showed that it has competitive performance with Cuckoo Search (CS), PSO, and CA. In this work, an application of MCA has been done in the traveling salesman problem. A set of classical datasets which have cities scale ranged from 51 to 150 was used in the application. The results show that the MCA is superior to Simulated Annealing (SA), Tabu Search (TS), GA, and CA for 60% of all datasets. Also, it was given that a detailed analysis presents the number of best, worst, average solutions, standard deviation, and the average CPU time concerning meta-heuristics. The metrics stress that MCA demonstrates a performance rate over 50% in finding optimal solutions. Finally, MCA solves the discrete problem in reasonable times in comparison to other algorithms for all datasets.

Keywords: *Modified Camel Algorithm, Meta-heuristic Algorithms, Traveling Salesman Problem*

1. INTRODUCTION

Combinatorial optimization is a popular research area in the last decades. Besides, solving combinatorial problems with meta-heuristic approaches and comparing them in popular problems are also interesting research of field [1-3]. Meta-heuristics are nature-inspired algorithms that simulate natural phenomena and put forth solutions to mathematical problems. In general, meta-heuristics are investigated in classical and modern meta-heuristics [4, 5]. Classical meta-heuristics are simulated annealing (SA), tabu search (TS), genetic algorithm (GA), particle swarm optimization (PSO), and ant colony optimization (ACO). Artificial Bee Colony algorithm (ABC), Black Hole algorithm (BH), Sine-Cosine algorithm (SCA), Lion Optimization algorithm (LOA), Water-Wave optimization (WWA), Artificial Atom algorithm (A^3), and Physarum-Energy optimization algorithm (PEO) are popular examples of modern meta-heuristics [6-12].

JSR A
Journal of Scientific Reports

Modern meta-heuristics generally start with an initial population and evolving this population with the algorithm mechanism in every iteration. Algorithms mostly have two features in their structures: intensification and diversification. Diversification means that the algorithm searches new areas and has the chance of finding many solutions in the solution space. On the other hand, intensification helps the algorithm to focus on optimal solutions and escape from local solutions. To converge optimality efficiently, meta-heuristics may hybridize with other algorithms or be improved with new mechanisms. Nature-inspired meta-heuristics have been often implemented in business administration, industrial engineering, computer engineering, technology, and other fields of science [13-15].

Camel algorithm mimics the traveling behavior of camels in the desert under several factors. The algorithm explores solution space at a certain level of endurance, supply, and temperature. Even if CA has several parameters, it is a simple and assertive approach to many optimization problems [16-18]. MCA is an approach that was proposed for engineering applications in previous research. It was successfully applied to the anti-jamming smart antenna optimization and the optimization of the proportional-integral-derivative (PID) controller parameters. It is concluded that MCA is a good candidate for online optimization systems [17].

In this study, the MCA is applied to solve the traveling salesman problem (TSP). The traveling salesman problem is a popular test problem for evaluating the efficiency and effectiveness of the optimization algorithms. The discrete problem aims to find an optimal tour, visiting each city exactly once and returning to the starting city where the total length of the tour is optimized [19].

The rest of the paper is organized as follows: In Sect. 2, the traveling salesman problem is widely discussed. The MCA is given in Sect. 3. In Sect. 4, the experimental results of the performance analysis of the MCA are described, and finally, Sect. 5 contains a conclusion and future expectations of this work.

2. TRAVELING SALESMAN PROBLEM

The traveling salesman problem (TSP) is a widely studied and well-known benchmark problem in the field of engineering and optimization. Many scientists have been working on that problem to solve it optimally in reasonable times [20]. To solve the problem, many approaches have been applied to solve TSP and its variants. A traveling salesman problem is a problem in which a salesman travels all the cities exactly once and returns to the initial city in optimal distance. The salesman can complete its tour in optimal distance, optimal time, optimal budget, and other objectives. TSP is generally investigated in symmetric and asymmetric types in many studies [21-24]. In symmetric type (s-TSP), the cost of an edge $\left(d_{ij} = d_{ji}\right)$ is valid for all points. Otherwise, if $\left(d_{ij} \neq d_{ji}\right)$ for at least one, then the TSP becomes asymmetric TSP (a-TSP). The TSP is also classified as the number of tours or salesmen. Double TSP (d-TSP) and multiple TSP (m-TSP) are the well-known types of traveling salesman problems. Besides, many researchers are still working on the generalizations of TSP, which are the traveling purchaser problem and the vehicle routing problem.

The traveling salesman problem (TSP) is an NP-hard problem, so finding an optimal solution to the problem in a reasonable time with linear programming requires exponential time. If a large size of data is used, the number of solutions becomes n! When n gets large, it will be impossible to find all of the solutions in a polynomial time. Thus, many exact, heuristic, and meta-heuristic methods are used

JSR A
Journal of Scientific Reports

to solve the TSP problem [25, 26]. Especially, classical and modern meta-heuristics are the highly potential solution algorithms for the TSP problem.

To define TSP in a short form, the problem can be described as: *N* is the set of *n* cities, *E* is the set of the edges, and $D_{ij} = (d_{ij})$ is the distance matrix between city *i* and city *j*. $P_{ij} = \{1,2, \ldots, n,1\}$ is the permutation of the constructed tours. 1 represents the first city; n represents the nth city of all the permutations. Then, the model of the problem is briefly given in Eq. 1.

$$Min. \sum_{i=1}^{n-1} (d_{i,i+1}) + d_{n,1} \tag{1}$$

The Euclidean distance is applied to calculate the distance between cities using Eq. 2.

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

3. MODIFIED CAMEL ALGORITHM

The camel algorithm (CA) is one of the exciting meta-heuristic algorithms in the literature. It is based on the camel traveling behavior in the desert under strict conditions. In the MCA, the camels (camel caravan) are searching for the best positions and looking randomly for the food supply [17]. Therefore, they try to survive and live for a long time in the desert. In the camel algorithm, each camel has its initial supply (*S*) at the beginning, and then both temperature (*T*) and journey duration affect camel endurance (*E*). The temperature *T* of a camel varies randomly between T_{min} and T_{max} using Eq. 3.

$$T_d^{i,iter} = (T_{\max} - T_{\min}) * Rand + T_{\min} \tag{3}$$

However, the temperature is the primary factor that affects the camel endurance; the endurance is redefined in the modified camel algorithm using Eq. 4 [17].

$$E_d^{i,iter} = 1 - \left(\frac{T_d^{i,iter} - T_{\min}}{T_{\max} - T_{\min}} \right) \tag{4}$$

In the MCA, there are two options for producing new solutions or updating the new locations. When the camel visibility v is less than a specific probability, the first alternative occurs using Eq. 5.

$$x_d^{i,iter} = x_d^{i,iter-1} + E_d^{i,iter} * \left(x_d^{best} - x_d^{i,iter-1} \right) \tag{5}$$

On the other hand, when the camel visibility v is larger than a specific probability, the second alternative is realized using Eq. 6.

$$x_d^{i,iter} = (x_{\max} - x_{\min}) * Rand + X_{\min} \tag{6}$$

JSR A
Journal of Scientific Reports

In the light of Eq. 3-6, the pseudo-code of the MCA with oasis condition is shown in Fig. 1 [17].

Algorithm: Modified CA

Begin
Step 1: Initialization: Set the min. and max. temperature T_{min} and T_{max}; set the camel caravan size and the dimensions; set the visibility threshold; Initialize the camel caravan using the Eq. (6).
Step 2: Subject the locations to a certain fitness function; determine the current best location; randomly assign a visibility (*v*) for each camel.
Step 3: While (iter < itermax) **do**
for i=1: Camel Caravan size
Compute the temperature T from Eq. (3)
Compute the endurance E from Eq. (4)
If *v* < *visibility threshold* **then**
Update the camel location from Eq. (5)
Else
Update the camel location from Eq. (6)
End If
If (oasis condition occur)
Replenish Endurance
End If
End for
Subject the new locations to the fitness function
If the new best location is better than the older one
The new best is the global best
End If
Assign new visibility for each camel
Step 4: End While
Step 5: Output the best solution
End

Figure 1. Pseudo-code of the MCA.

The neighborhood operators are used to obtain the new solutions from the current solution. A neighborhood operator or the combination of operators may generate new solutions [27, 28]. In this study, four basic neighborhood operators are chosen randomly and applied once in each iteration. The selected operators for the optimization process are swap, insert, reverse, and swap_reverse. As previous papers indicate, the use of a single structure can be divergent and cannot search optimal regions of the search space [28, 29]. Thus, instead of using a single structure, it would hopefully give better results when multiple combinations are used. Then, the neighborhood selection is defined by the minimum of four operators using Eq. 7.

$$NH(x) = \min\left(swap(x), insert(x), reverse(x), swap\_reverse(x)\right) \quad (7)$$

However, the MCA converges at a later stage by using multiple combinations. It will be needed more iteration and computation time to find near-optimal solutions [27-29]. On the other hand, the use of the four investigated structures would give better solutions: swapping, insertion, reversing, and swapping of reversed subsequences than others at 1000-2000 iterations.

JSR A
Journal of Scientific Reports

4. EXPERIMENTAL ANALYSIS

The ten datasets ranged from 51 to 150 cities were selected from the TSPLIB library in the implementation. In this section, all the experiments were run on Intel® Core™ i7 3520-M CPU 2.9 GHz speed with 8 GB RAM by using Matlab. The algorithms which are MCA, CA, GA, TS, and SA are compared to demonstrate the performance of the MCA. All the algorithms were run 10 times independently for optimal parameters and 1000-2000 iterations for each run. There are many parameters for the used meta-heuristics. However, it is mentioned the fundamentals for algorithms. In the SA algorithm, initial temperature (T_0 =40000), cooling rate (r=0.80), and the iteration limit for temperature change (L=10) are sufficient for optimization. In the TS algorithm, the tabu length (L=30) is the adequate parameter. In GA, the crossover rate is 0.80, the mutation rate is 0.02 and the population size is 100. In CA and MCA, the dimension of space (dim=10), min. and max. temperature (T_{min}=0, T_{max} =100), initial endurance (Init_End=1), visibility threshold (Vis=0.5), dying rate (dye_rate=0) are taken as optimal parameters. In CA, initial supply (Init_Supp=1) is available.

JSR A
Journal of Scientific Reports

Table 1. Computational results of algorithms on the medium-scale TSP instances.

TSP	Measure	SA	TS	GA	CA	MCA
eil51	Best	478.97	488.49	478.73	489.7	471.43
(426)	Worst	591.41	577.43	525.4	569.24	495.75
	Avg	541.57	527.14	499.14	514.6	478.79
	Std.	36.85	26.94	18.16	28.77	8.18
	Time	0.2	0.35	46.63	18.29	15.95
	Number	1000	1000	1000	1000	1000
berlin52	Best	8591.18	8466.93	8104.65	8373.08	8357.37
(7542)	Worst	10329.7	9825.44	8883.47	10481	9011.83
	Avg	9415.69	9452.24	8451.47	9194.62	8659.57
	Std.	546.37	413.04	225.23	561.6	213.12
	Time	0.25	0.33	48.62	18.96	16.46
	Number	1000	1000	1000	1000	1000
st70	Best	952.37	920.14	930.31	892.84	898.43
(675)	Worst	1206.55	1212.84	1057.09	1120.32	987.66
	Avg	1064.81	1011.57	1001.24	1021.07	953.59
	Std.	92.9	81.2	45.24	67.11	36.79
	Time	0.49	0.30	69.81	21.38	18.78
	Number	1000	1000	1000	1000	1000
eil76	Best	744.43	745.62	781.92	698.83	676.74
(538)	Worst	897.59	908.12	886.94	815.02	787.58
	Avg	805.4	789.54	827.71	773.3	746.38
	Std.	53.49	47.5	35.87	35.85	31.62
	Time	0.27	0.32	81.84	21.25	19.6
	Number	1000	1000	1000	1000	1000
pr76	Best	144562	138357	128068	147348	146743
(108159)	Worst	162619	148753	149040	178085	162307
	Avg	155394	142024	135946	164720	157814
	Std.	5059.64	3623.74	5544.75	9510.72	4688.28
	Time	0.24	0.33	65.33	22.05	19.12
	Number	1000	1000	1000	1000	1000

Table 1. continued.

TSP	Measure	SA	TS	GA	CA	MCA
rat99	Best	2024.28	1566.22	1521.5	1993.3	1937.49
(1211)	Worst	2214.42	1758.13	1836.3	2399.76	2370.5
	Avg	2142.83	1666.17	1655.32	2222.23	2172.74
	Std.	66.64	63.25	81.33	121.06	146.47
	Time	0.30	0.42	101.5	24.94	21.52
	Number	1000	1000	1000	1000	1000
kroa100	Best	34383.6	35136.1	32865.5	33863.5	29816.7
(21282)	Worst	39483.7	43265.7	37832.2	43562.7	35823
	Avg	37055.2	38186.5	35417.8	37895.5	33721.5
	Std.	1649.85	3080.13	1446.68	2974.65	1746.97

JSR A
Journal of Scientific Reports

	Time	0.64	0.63	191.92	51.01	43.95
	Number	2000	2000	2000	2000	2000
eil101	Best	864.99	862.19	855.68	870.06	843.11
(629)	Worst	1007.91	998.82	970.2	1043.86	943.08
	Avg	938.98	921.97	904.38	939.7	892.74
	Std.	46.19	43.93	33.63	54.01	31.93
	Time	0.77	0.66	196.15	50.7	50.58
	Number	2000	2000	2000	2000	2000
bier127	Best	175583	175733	170334	186581	181965
(118282)	Worst	201548	203137	189736	213526	200309
	Avg	192011	187255	179673	197687	191837
	Std.	7596.82	7756.69	5897.25	9587.58	6177
	Time	0.81	0.78	262.47	57.51	50.45
	Number	2000	2000	2000	2000	2000
kroa150	Best	54469.6	56795.3	59429.8	53625.4	54587.5
(26524)	Worst	66400.4	64183.5	66014.5	65501.6	63608.6
	Avg	60349.3	60694.1	62526.7	58669.9	58616.2
	Std.	3571.35	2803.79	1808.7	3654.78	2691.54
	Time	0.72	0.85	349.21	64.88	57.24
	Number	2000	2000	2000	2000	2000

Table 2. The #of optimal solutions and average CPU time.

Alg.	**Best**	**Worst**	**Avg.**	**Std.**	**Avg. Time**
MCA	4	6	6	5	31.37
CA	2	0	0	0	35.10
GA	4	2	4	3	141.35
TS	0	2	0	2	0.5
SA	0	0	0	0	0.47

Table 1 shows the experimental results and comparison between MCA, CA, GA, TS, and SA. In this table, the results are given as best, worst, average solution, standard deviation, and CPU Time.

JSR A
Journal of Scientific Reports

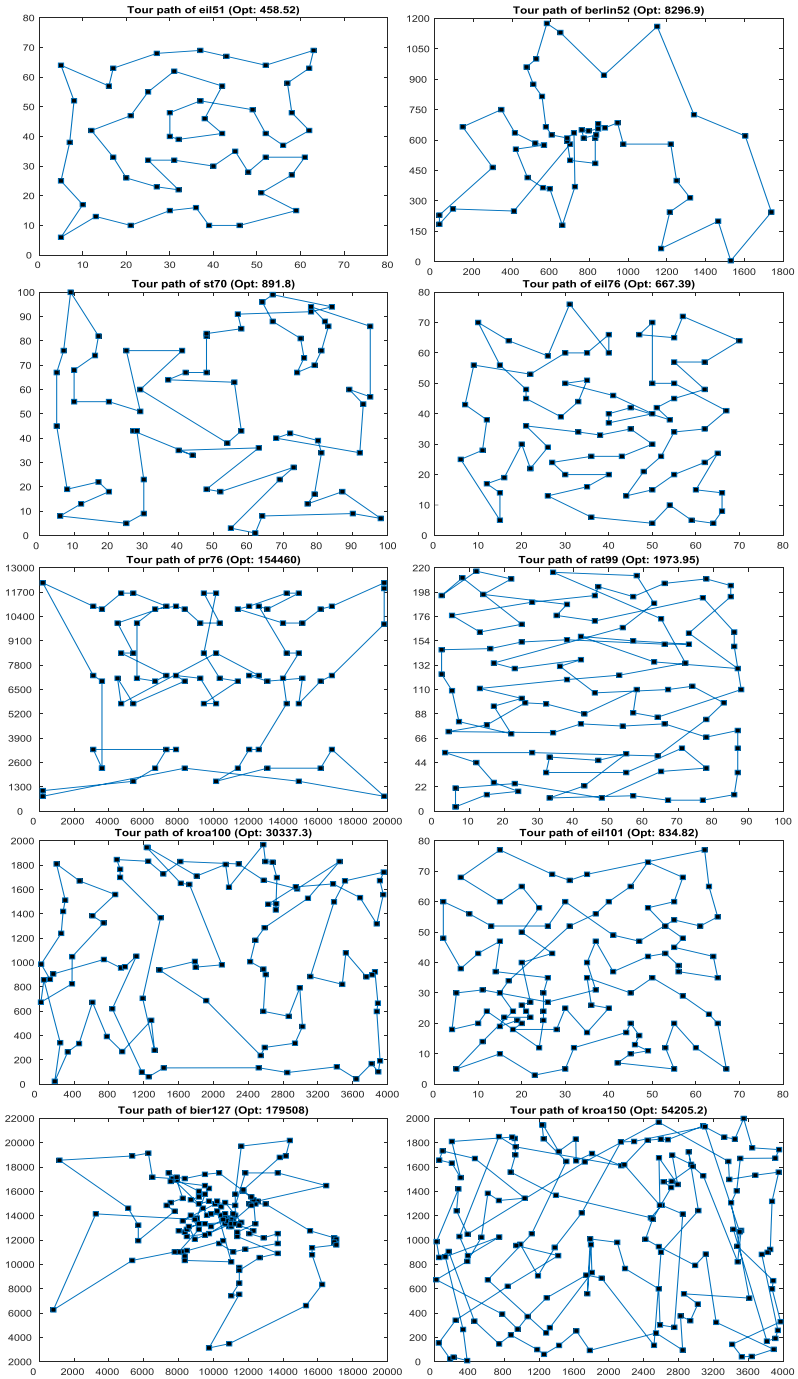

Figure 2. A set of optimal results found by the modified CA (MCA).

As inferred from Table 1, it can be observed that the quality of the MCA solutions is better compared to CA, GA, TS, and SA for 60% of all datasets. Besides, in Table 2, MCA finds 21 optimal, CA finds 2 optimal, GA finds 13 optimal, TS finds 4 optimal and SA finds never optimal solutions among 40 best results. Table 2 summarizes that MCA is superior to CA, GA algorithm, TS, and SA for 53% of all optimal solutions. MCA has low standard deviations among other algorithms. A low standard deviation specifies that the MCA is a more reliable and certain approach to find the optimal results. Lastly, MCA solves the TSP problem in reasonable times in comparison to other algorithms for all datasets.

In general, the experimental analysis shows that the MCA is a reliable and certain approach for solving the traveling salesman problem. This modified meta-heuristic gives better results and reasonable standard deviation as compared to the test meta-heuristics. Figure 2 shows a set of optimal results found by the MCA on the benchmark datasets.

5. CONCLUSION

In recent decades, solving discrete problems via modern meta-heuristics is a popular research area. In this paper, the MCA is applied to the symmetric TSP instances. To evaluate the performance of MCA, it has been tested on ten benchmark test datasets. The experimental results show that the MCA can find better solutions compared to the camel algorithm (CA), genetic algorithm (GA), simulated annealing (SA), and tabu search (TS) for 60% of all datasets and 53% of all optimal solutions. As CPU time is considered, MCA is considerably fast (31.37 secs.) to find optimal solutions. In future works, MCA can be compared with other meta-heuristics to evaluate performance analysis in scheduling, assignment, timetabling, routing, and many other combinatorial problems.

ACKNOWLEDGEMENTS

This research received no specific grants from any funding agency in public, commercial or non-profit sectors.

REFERENCES

[1] Parejo, J.A., Ruiz-Cortés, A., Lozano, S., and Fernandez, P., (2012). Metaheuristic optimization frameworks: a survey and benchmarking. Soft Computing, 16, 527–561. https://doi.org/10.1007/s00500-011-0754-8.

[2] Cárdenas-Montes, M., (2018). Creating hard-to-solve instances of travelling salesman problem. Applied Soft Computing, 71, 268-276. https://doi.org/10.1016/j.asoc.2018.07.010.

[3] Yang, XS., (2010). Nature-inspired metaheuristic algorithms. United Kingdom (Bristol): Luniver Press.

[4] Gogna, A. and Tayal, A., (2013). Metaheuristics: review and application. Journal of Experimental & Theoretical Artificial Intelligence, 25(4), 503–526. http://doi.org/10.1080/0952813X.2013.782347.

[5] Rajpurohit, J., Sharma, T.K., Abraham, A., and Vaishali, (2017). Glossary of metaheuristic algorithms. International Journal of Computer Information Systems and Industrial Management Applications, 9, 181-205.

[6] Karaboga, D. and Basturk, B., (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization, 39(3), 459–471.

[7] Hatamlou, A., (2018). Solving travelling salesman problem using black hole algorithm. Soft Computing, 22(24), 8167-8175. https://doi.org/10.1007/s00500-017-2760-y.

[8] Mirjalili, S., (2016). A sine cosine algorithm for solving optimization problems. Knowledge-Based Systems, 96, 120-133.

[9] Yazdani, M. and Jolai, F., (2016). Lion optimization algorithm (LOA): A nature-inspired metaheuristic algorithm. Journal of Computational Design and Engineering, 3(1), 24-36.

[10] Zheng, Y.J., (2015). Water wave optimization: a new nature-inspired metaheuristic. Computers & Operations Research, 55:1–11.

[11] Yildirim, A.E. and Karci A., (2018). Applications of artificial atom algorithm to small-scale traveling salesman problems. Soft Computing, 22(22), 7619-7631. https://doi.org/10.1007/s00500-017-2735-z.

[12] Feng, X., Liu, Y., Yu, H., and Luo, F. (2019). Physarum-energy optimization algorithm. Soft Computing, 23, 871–888. https://doi.org/10.1007/s00500-017-2796-z.

[13] Dorigo, M., Birattari, M., and Stutzle, T., (2006, Nov). Ant colony optimization. IEEE Computational Intelligence Magazine, 1(4), 28-39.

[14] Akça, M.R. (2011). Yapay arı kolonisi algoritması kullanılarak gezgin satıcı probleminin Türkiye'deki il ve ilçe merkezlerine uygulanması, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.

[15] Bektas, T., (2006). The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega, 34(3), 209-219.

[16] Ibrahim, M.K. and Ali, R.S., (2016). Novel optimization algorithm inspired by camel traveling behavior. Iraqi Journal for Electrical and Electronic Engineering, 12(2), 167-177.

[17] Ali, R.S., Alnahwi, F.M., and Abdullah, A.S., (2019). A modified camel travelling behavior algorithm for engineering applications. Australian Journal of Electrical and Electronics Engineering, 16(3), 176-186. https://doi.org/10.1080/1448837X.2019.1640010.

[18] Hassan, K.H, Abdulmuttalib, T.R., and Jasim, B.H., (2021, Feb). Parameters estimation of solar photovoltaic module using camel behavior search algorithm. International Journal of Electrical and Computer Engineering (IJECE), 11(1), 788-793.

[19] Burkard, R.E., Deineko, V.G., van Dal, R., van, J.A.A., Veen, der, and Woeginger, G.J., (1998). Well-solvable special cases of the traveling salesman problem: a survey. Society for Industrial and Applied Mathematics, 40(3), 496-546. https://doi.org/10.1137/S0036144596297514.

[20] Sahana, S.K., (2019). Hybrid optimizer for the travelling salesman problem. Evolutionary Intelligence, 12, 179–188. https://doi.org/10.1007/s12065-019-00208-7.

[21] Nagata, Y. and Soler, D., (2012). A new genetic algorithm for the asymmetric traveling salesman problem. Expert Systems with Applications, 39(10), 8947-8953. https://doi.org/10.1016/j.eswa.2012.02.029.

[22] Osaba, E., Yang, XS., Diaz, F., Lopez-Garcia, P., and Carballedo, R., (2016). An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. Engineering Applications of Artificial Intelligence, 48, 59-71. https://doi.org/10.1016/j.engappai.2015.10.006

[23] Öncan, T., Altınel, İ.K., and Laporte, G., (2009). A comparative analysis of several asymmetric traveling salesman problem formulations. Computers & Operations Research, 36(9), 637-654.https://doi.org/10.1016/j.cor.2007.11.008.

[24] Malik, W., Rathinam, S., and Darbha, S., (2007). An approximation algorithm for a symmetric generalized multiple depot, multiple travelling salesman problem. Operations Research Letters, 35(6), 747-753. https://doi.org/10.1016/j.orl.2007.02.001.

[25] Asih, A.M.S., Sopha, B.M., and Kriptaniadewa, G., (2017). Comparison study of metaheuristics: empirical application of delivery problems. International Journal of Engineering Business Management, 9, 1-12. https://doi.org/10.1177/1847979017743603.

[26] Khanra, A., Maiti, M.K., and Maiti, M., (2015). Profit maximization of tsp through a hybrid algorithm. Computers & Industrial Engineering, 88, 229-236. https://doi.org/10.1016/j.cie.2015.06.018.

[27] Halim, A.H. and Ismail, I., (2019). Combinatorial optimization: comparison of heuristic algorithms in travelling salesman problem. Archives of Computational Methods in Engineering, 26, 367–380. https://doi.org/10.1007/s11831-017-9247-y.

[28] Szeto, W.Y., Yongzhong, W. and Ho, S.C., (2011). An artificial bee colony algorithm for the capacitated vehicle routing problem. European Journal of Operational Research, 215(1), 126-135. https://doi.org/10.1016/j.ejor.2011.06.006.

[29] Demiral, M.F. and Işik, A.H., (2020). Simulated annealing algorithm for a medium-sized tsp data. In: D. J. Hemanth and U. Kose (Eds), Artificial Intelligence and Applied Mathematics in Engineering Problems. ICAIAME 2019. Lecture Notes on Data Engineering and Communications Technologies, 43. Springer, Cham, pp. 457-465. https://doi.org/10.1007/978-3-030-36178-5_35.