IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 1, March 2022, pp. 41~49
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i1.pp41-49

Solving a traveling salesman problem using meta-heuristics

Anahita Sabagh Nejad, Gabor Fazekas
Department of Informatics, University of Debrecen, Debrecen, Hungary

Article Info

Article history:

Received Mar 25, 2021
Revised Dec 16, 2021
Accepted Dec 30, 2021

Keywords:

Clustering method
Traveling salesman problem meta-heuristics
Whale optimization algorithm

ABSTRACT

In this article, we have introduced an advanced new method of solving a traveling salesman problem (TSP) with the whale optimization algorithm (WOA), and K-means which is a partitioning-based algorithm used in clustering. The whale optimization algorithm first was introduced in 2016 and later used to solve a TSP problem. In the TSP problem, finding the best path, which is the path with the lowest value in the fitness function, has always been difficult and time-consuming. In our algorithm, we want to find the best tour by combining it with K-means which is a clustering method. In other words, we want to divide our problem into smaller parts called clusters, and then we join the clusters based on their distances. To do this, the WOA algorithm, TSP, and K-means must be combined. Separately, the WOA-TSP algorithm which is an unclustered algorithm is also implemented to be compared with the proposed algorithm. The results are shown through some figures and tables, which prove the effectiveness of this new method.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Anahita Sabagh Nejad
Department of Informatics, University of Debrecen
4028 Debrecen, Kassai Street 26, Hungary
Email1: Anahita.sabagh@inf.unideb.hu

1. INTRODUCTION

Solving the traveling salesman problem (TSP) has always been one of the topics of interest to researchers for example solving the vehicle routing problem that was first introduced by Dantzig *et al*. [1], [2]. The problem of determining the travel route is similar to finding the solution from the TSP problem [3]. Since TSP is an NP-hard problem and it takes a protracted time to search out a tour among the cities, the complexity order of this problem becomes exponential which does not have a suitable execution time [4], but an intelligent method requires less computation time and more accuracy [5]. Metaheuristic algorithms are powerful methods for solving many tough optimization problems [6]. In the past three decades, meta-heuristics pose a potential impact on tasks of managing operational problems (e.g. assignment and scheduling) [7].

Swarm intelligence is an optimization technique that looks for the domain by developing a population of individuals. These individuals represent solutions for our problems [8]. The individuals move to the better solution areas. Each iteration forces the individuals to cooperate till reaching the best solution [8]–[10]. The swarm-based algorithms have been proven to be effective in solving nonlinear optimization problems in a large space search domain [11]–[13]. Over the last decades, nature has been a source of inspiration for several meta-heuristics, which have been introduced to solve optimization problems. These meta-heuristics have been tested to solve discrete problems [14]. Some of the nature- inspired algorithms are: particle swarm optimization (PSO) [15], [16], ant colony optimization (ACO) [17]–[19], artificial bee colony (ABC) [20], cuckoo search (CS) algorithm [21], [22], krill herd (KH) [23], and spotted hyena optimizer (SHO) [24].

Exploration and extraction are two basic factors that control a metaheuristic algorithm. Exploration alludes to a metaheuristic algorithm's capacity to search in diverse environment ranges to find the ideal solutions [25]–[27]. On the other hand, exploration is the ability to centralize search inside the optimal range to extract the optimal solution. A metaheuristic algorithm balances these two clashing objectives so that in any metaheuristic calculation or the progressed version, performance is moved forward by controlling the aforementioned parameters [27].

In [28], a new clusterized technique was introduced based on K-means to solve TSP by using the firefly algorithm which we extended that by using the WOA as a new studying. In TSP, with the view to minimize the cost function and hence improve the efficiency, the WOA is combined with the K-means. The same technique is applied to the unclustered algorithm, but based on the characteristics of the TSP problem [29] and the mechanism of the WOA, The maximum scale is determined to have 1323 cities, because the unclustered method is not as good as the clustered method, and we wanted to use the same datasets for both. The figures prove this in the Subsection 4.2. Section 2 provides some detailed information about the TSP, K-means, WOA algorithm, and mathematical models. In section 3, the proposed method is explained, while the results and discussions of the proposed method have been provided in section 4. Finally, the conclusion has been discussed in section 5.

2. THE COMPREHENSIVE THEORETICAL BASIS

In the following, we explain some theories in the subsection parts, which are numbered 2.1 to 2.4 for the TSP problem, K-means, WOA, and mathematical models respectively. The mathematical models have 3 types that are explained with a peseodocode. These types are encircling Prey, bubble-Net attacking method, exploration phase or search for the prey.

2.1. The travelling salesman problem (TSP)

In the theory, the problem is explained on a graph G=(V, A), where we have n vertices $V = \{v1,\ v2, \ldots, vn\}$, and a distance matrix $C = (c_{ij})$. We have described **A** as a set of arcs [30] in the (1). In TSP problem, our objective function is the tour with minimum distance. If we want to solve that we need $\frac{(n-1)!}{2}$ [31] comparisons, which makes it impossible to be solved theoretically. We want to improve the time and the complexity of this problem by using the WOA algorithm discovered by Mirjalili, and a data mining technique called K-means. Since TSP belongs to the vehicle routing problem (VRP) and has a minimum cost objective function [32], we apply heuristic algorithms to find the best solutions.

$$A = \{(v_i,\ v_j) \big|\ v_i,\ v_j \in V,\ i \neq j\} \quad (1)$$

2.2. K-means algorithm

Clustering is a method that partitions our data into some groups, so-called clusters [33]–[38]. The steps of data mining start with K clusters, for K centroids. This approach which is iterative starts with a random selection of K objects for K clusters that is the first assignment. As it is explained in the algorithm below, the algorithm should compute the average of the objects for the new assignments of objects to the nearest cluster.

Algorithm: K-means,

```
Inputs:
K: cluster numbers or the initial centroids,
D: a set of n objects,
Output: K clusters,
Method:
(1) arbitrarily choose K objects from D,
(2) repeat,
(3) according to the average of the objects within the cluster, (re)assign the objects to
the closest one
(4) update by calculating the average of the objects for each cluster, and new assignments
(5) until no change happens in the clusters.
```

2.3. The whale optimization algorithm

This algorithm is a type of swarm intelligence algorithm which is inspired by the characteristics of whales [39]. The hunters look for a group of prey to encircle and gradually tighten the ring until they catch that prey [40]. Whales have a special hunting method that is called the bubble-net feeding method which is done by creating a distinctive bubble along a circle [41].

2.4. Mathematical models
Three mathematical models exist that based on them we explain the equations. These models are encircling prey, bubble-net attacking method (spiral bubble net feeding maneuver), and search for the prey. The equations are addressed in the pseudocode of the WOA algorithm. In each model, the D and X parameters obtain new values based on their new positions and distances.

2.4.1. Encircling prey
Whales don't know the best position at the first step, so the target prey becomes the current best candidate solution, but in the next iterations after finding the best agent, the other search agents will try to change their positions to the best candidate solutions. The best search agent is the closest one. The other whales change their positions toward to the best answer. Based on Mirjalili and all, we have the (2) to (5):

$$\overrightarrow{D'} = |\vec{C}.\overrightarrow{X*}(t) - \vec{X}(t)| \tag{2}$$

$$\vec{X}(t+1) = \overrightarrow{X*}(t) - \vec{A}.\vec{D} \tag{3}$$

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \tag{4}$$

$$\vec{C} = 2.\vec{r} \tag{5}$$

D is the distance from (2), but *A* and *C* are coefficients, A=[-a, a] where a is a decreasing number from 2 to 0, and X represents the whale's position, which we use as the agent, r is a number that is selected randomly in [0, 1], *X(t+1)* is the next position for the next whale, which can be a better answer, and t indicates to the iterations. *X*(t)* is the best solution among the whales that we have searched so far, and my change later.

2.4.2. Bubble-net attacking method
We have these methods; i) shrinking encircling mechanism, and ii) spiral updating position (exploration). In another word, The WOA algorithm adopted these two approaches to update the positions of the whales. The first method is realized by setting a by (4), except for the second approach, that's spiral updating position, we've got the following equations, where D is the distance, b indicates a constant for logarithmic shape, and l stands for a value in this interval [-1,1].

$$\vec{X}(t+1) = \overrightarrow{D'}.e^{bl}.cos(2\pi l) + \overrightarrow{X*}(t) \tag{6}$$

$$\overrightarrow{D'} = \overrightarrow{X*}(t) - \vec{X}(t) \tag{7}$$

Selection between these two approaches is based on the p-value that is a random number in [0, 1]. This is for updating or changing the position. If $p < 0.5$ we apply the (3) for X(t+1), otherwise if $p \geq 0.5$, we apply (6), as it is summarized (8):

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X*}(t) - \vec{A}.\vec{D} & \text{if } p < 0.5) \\ \overrightarrow{D'}.e^{bl}.cos(2\pi l) + \overrightarrow{X*}(t) & \text{if } p \geq 0.5 \end{cases} \tag{8}$$

2.4.3. Exploration phase
If $|A| \geq 1$, the whales select a random whale. This step is called exploration, and they update their distances and positions by (9) and (10). Based on the new $\vec{X}_{rand}$, which is the position of a random whale, the values of $\vec{D}$, and $\vec{X}(t+1)$ change, as it is addressed in step 12 of the pseudocode. The parameter of a changes in an interval from 2 to zero to give the functionality of searching for the prey. This search is global.

$$\vec{D} = |\vec{C}.\vec{X}_{rand} - \vec{X}| \tag{9}$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A}.\vec{D} \tag{10}$$

2.4.4. The pseudocode of the whale optimization algorithm (WOA)
The pseudocode which is defined as follows describes the summarization of the above-mentioned mathematical models for WOA. After defining the population size, as each whale is an agent, we should find

its fitness. In this problem, our fitness is minimized. Before meeting the termination criterion in the proposed method, we update the values, and according to the equations, and conditions which we described earlier, we find the optimal position of the best whale with minimum fitness.

1: Initialize the population X_i (i=1, 2, … n)
2: Calculate the fitness of each search agent
3: X*=the best search agent
4: While (t < maximum number of iterations)
5: for each search agent
6: Update a, A, C, l, and p
7: If1 ($p < 0.5$)
8: if2 ($|A| < 1$)
9: Update the position of the current search agent by the (3)
10: else if2 ($|A| \geq 1$)
11: Select a random search agent (X_{rand})
12: Update the position for the current agent by the (10)
13: else if1($p \geq 0.5$)
14: Update the position of the current search by the (6)
15: Check if any search agent exists beyond the search space and amend it
16: Calculate the fitness of each search agent
17: Update X*

3. THE PROPOSED METHOD AND ALGORITHM

In this work, we are proposing a method that combines K-means with WOA to solve TSP. This new TSP solver finds the number of clusters based on (11), where N is the number of cities [37]. First, we apply K-means to divide our data into K clusters, then we apply the WOA algorithm to find a tour in each cluster which is minimum, and in the final step, we connect our clusters to find the best solution that is the optimal path.

$$K = \sqrt{\frac{N}{2}} \tag{11}$$

We apply the next [28] procedure:

i) We calculate the distances between the centroids of each cluster to find the clusters with minimum distances (the (12) for finding the distance).
ii) We combine the selected clusters in one bigger cluster [37]. Then we connect the tours of these two newfound clusters.
iii) We repeat step 1 until the generation of the minimum tour.

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{12}$$

In this method, each whale is an agent and can be a solution for TSP. The algorithm updates to find the current best solutions. In this algorithm, we consider 4 candidate nodes in step 6. These nodes are around the centroids of that cluster, and just the closer one will be joined to another cluster so that we always select the nearest city. The steps of our algorithm are:

– Step 1: Initialize the number of population as shown in Table 1
– Step 2: Specify K based on the (11)
– Step 3: Applying K-means algorithm
– Step 4: Applying whale optimization algorithm for i=1: K
– Step 5: Find the position of the cities
– Step 6: Sorting by indexing
– Step 7: Find the nodes (cities) in each cluster that are closer to the centroid of that clusters
– Step 8: Join the closest cities to another cluster
– Step 9: Stopping criterion till no cluster remains unjoined

4. RESULTS AND DISCUSSIONS

The platform used for the implementation of this method is matrix laboratory (MATLAB), using Intel CORE i5, and 6 GB RAM. On the last page, our graphical illustrations and tables show the complexity of our algorithm before, and after applying K-means. Our datasets are Eil51, Linhp 318, and rl1323. All data are available in the standard TSP library [42]. The vector a is assigned to have the maximum value of 2 for all of the iterations.

4.1. Tables results before and after applying the clustering method

In these tables, the average of our best tour (fitness) is considered as the benchmark. Table 1 shows some parameters. The initial population and iterations have the same value for both algorithms and are chosen based on some experimental results during the execution of the program, but they can have different values as well. Table 2 represents the results of the fitness function before applying K-means clustering method. We have three datasets, and for each, we have calculated some statistics consisting of the min, max, average, and standard deviation of the fitness function. The fitness average for the first approach has the values of 1176.42, 393928.2, and 8262213. Table 3 indicates the fitness function of the clustered method, which is done by using the K-means approach. The results of the comparison between these two tables prove that the fitness average for the first approach has improved to 489.8785, 82908.2, and 1.19E+06 for the second approach for the same datasets. The statistical values of these two tables show how the fitness function improves when our algorithm combines with K-means, especially for the third dataset. Table 4 shows the execution time of the unclustered approach, which means solving a TSP problem using only the whales and without clustering. The time average has values of 3.765785 (s), 16.63365 (s), and 63.50511 (s), which shows high values especially for the third dataset with 1323 nodes (cities). As in Table 4, even the best value (min) for the third dataset takes 51.669 (s) to run. Table 5 shows the execution time for the clustered approach. Based on this table, time has improved since the time average is reduced from 3.765785 (s), 16.63365 (s), and 63.50511 (s) to 1.2205855(s), 5.707375 (s), and 24.81385 (s) for the clustered method. The minimum value for the third dataset takes 20.345 (s) to run. It confirms that our algorithm is improved more than 50%.

Table 1. The parameter settings

Initial Population	Iterations	Number of the Cities
100	20	51
100	20	318
100	20	1323

Table 2. The fitness value for the unclustered whale optimization algorithm

Dataset	Min	Max	Average	Stdev.s
Eil51	1100.269	1262.614	1176.421	48.53605
Linhp318	376614.9	409489.4	393928.2	8468.606
Rl1323	8174234.322	8740947	8262213	49003.98

Table 3. The fitness value for the clustered whale optimization algorithm

Dataset	Min	Max	Average	Stdev.s
Eil51	454.9	526.76	489.8785	21.2760405
Linhp318	77273	88040	82908.2	3180.155
Rl1323	1.14E+06	1.23E+06	1.19E+06	2.78E+04

Table 4. The execution time for the unclustered whale optimization algorithm

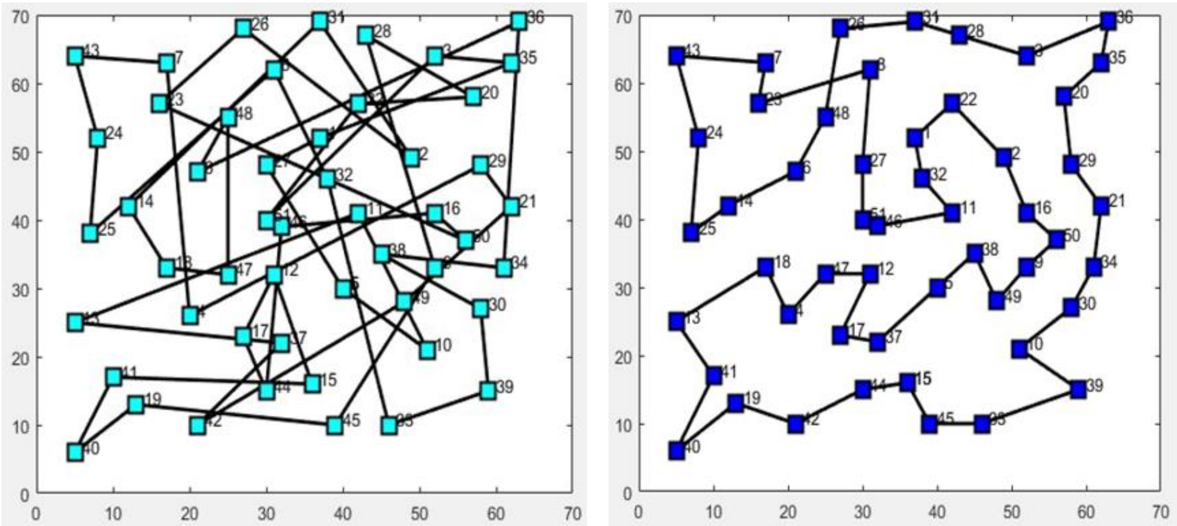
Dataset	Min	Max	Average	Stdev.s
Eil51	3.1421	4.7562	3.765785	o.396958
Linhp318	13.642	20.79	16.63365	1.77977
Rl1323	51.669	79.21	63.50511	6.286887

Table 5. The execution time for the clustered whale optimization algorithm

Dataset	Min	Max	Average	Stdev.s
Eil51	1.0021	1.4997	1.2205855	0.1708681
Linhp318	4.6754	7.953	5.707375	0.979334
Rl1323	20.345	55.258	24.81385	7.647444

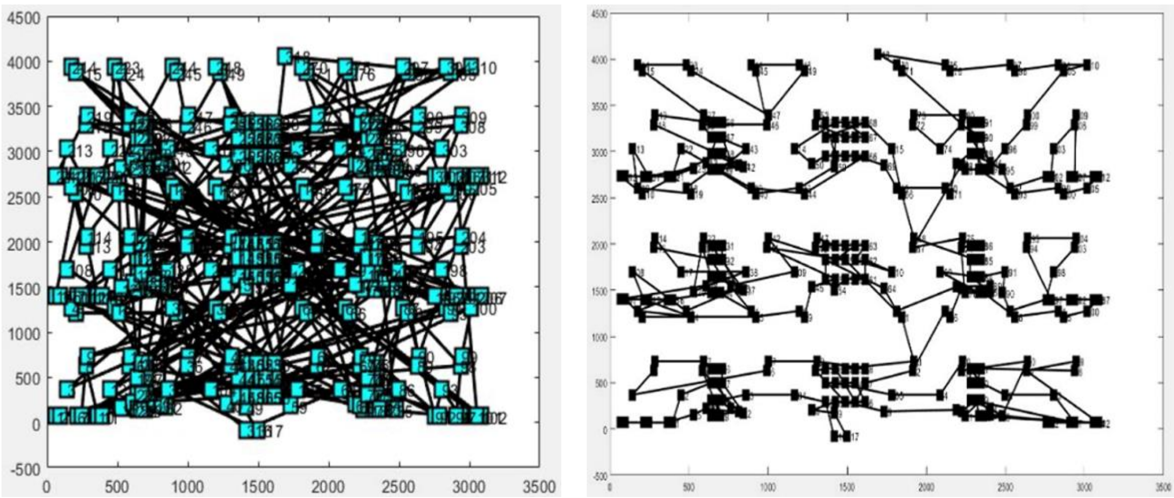
4.2. Figures

In this section, the figures show how this designed method is used to find a more efficient route so that all nodes are met once, and the salesman returns to the starting point. As it is obvious, the figures become more complex as the number of nodes becomes more, so the illustration of all the connections becomes hard as well, therefore the first illustration which is Figure 1 for the Eil51 dataset is a clearer indication of the calculations that have been done. Figure 1(a) shows a graph, which demonstrates the routes of the salesman from the time when our algorithm starts. Figure 1(b) demonstrates the salesman routing but with the help of clustering, so this figure finds the connections for the smallest distances, which are more optimized. In Figure 2 the TSP is solved for Linhp318 such that in (a) unclustered method and (b) clustered approach are presented. Figure 2(a) is more complicated than Figure 2(b). The number of the dataset is 318, but in Figure 2(b), the problem is solved with less complexity, and the reason is using the K-means clustering approach. This approach is helped to find better connections.



(a) (b)

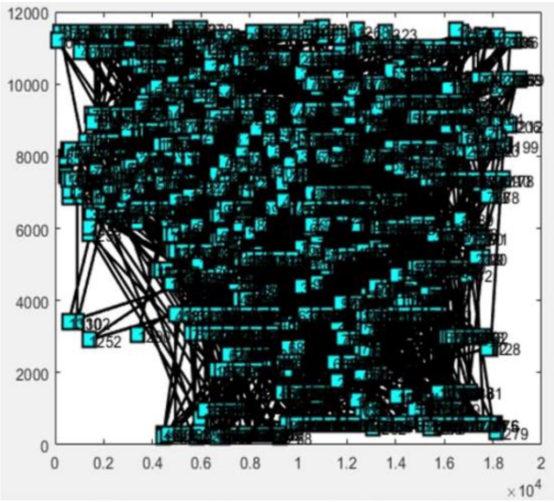
Figure 1. Comparison between different approaches for the Eil51 dataset (a) unclustered approach and (b) clustered approach



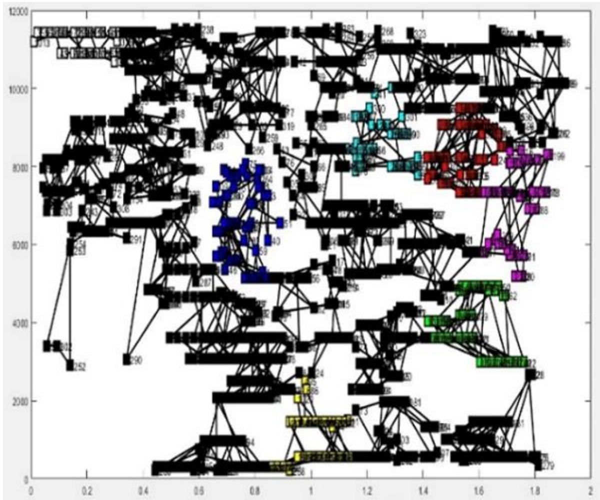
(a) (b)

Figure 2. Comparison between different approaches for the Linhp318 dataset (a) unclustered approach and (b) clustered approach

Figure 3 compares the results in the algorithm performance for the rl1323 dataset. In (a) unclustered method and (b) clustered approach are applied. Figure 3(a) shows more complexity than the previous figures. The reason is the number of the dataset with 1323 nodes, which means the salesman should travel between all these nodes and return to the first node. Solving such a problem with this amount of data practically is impossible but with the help of whales, it became solved with high complexity in Figure 3(a). Figure 3(b) is solved this problem with less complexity because of the clustering approach.



(a)



(b)

Figure 3. Comparison between different approaches for the Rl1323 dataset (a) unclusterd approach and (b) clustered approach

5. CONCLUSION AND FUTURE WORKS

This article introduced two approaches to solve a TSP: an unclustered method, and a combinatorial approach based on WOA, and K-means. The WOA algorithm, and K-means are combined to solve a TSP problem. This approach divides the problem into some clusters and applies WOA for these small clusters. In the end, this combined algorithm joins the nearest clusters. The results generate the optimal solution concerning the iterative best cost. This new clusterized heuristic has proven to dominate the first approach of solving the same problem based on mentioned tables, and figures. Future research will be for solving more high-scale optimization problems. Furthermore, this article is a prime studying for future research on clustering since the WOA algorithm can be hybridized with other clustering algorithms to find more efficient solutions than the proposed method.

ACKNOWLEDGEMENTS

I would like to thank my university for supporting this work, and my supervisor Dr. Gabor Fazekas for the encouragement, and guidance he has provided throughout my research.

REFERENCES

[1] G. B. Dantzig and J. H. Ramser, “The truck dispatching problem,” *Management Science*, vol. 6, no. 1, pp. 80–91, Oct. 1959, doi: 10.1287/mnsc.6.1.80.

[2] N. Alsumairat and M. Alrefaei, “Solving hybrid-vehicle routing problem using modified simulated annealing,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 6, pp. 4922–4931, Dec. 2021, doi: 10.11591/ijece.v11i6.pp4922-4931.

[3] Z. K. A. Baizal, K. M. Lhaksmana, A. A. Rahmawati, M. Kirom, and Z. Mubarok, “Travel route scheduling based on user’s preferences using simulated annealing,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 2, pp. 1275–1287, Apr. 2019, doi: 10.11591/ijece.v9i2.pp1275-1287.

[4] E. Damghanijazi and A. Mazidi, “Meta-heuristic approaches for solving travelling salesman problem,” *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, pp. 18–23, 2017.

[5] K. Loubna, B. Bachir, and Z. Izeddine, “Ant colony optimization for optimal low-pass state variable filter sizing,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 1, pp. 227–235, Feb. 2018, doi: 10.11591/ijece.v8i1.pp227-235.

[6] X.-S. Yang, “A new metaheuristic bat-inspired algorithm,” in *Nature Inspired Cooperative Strategies for Optimization (NICSO*

2010), Springer Berlin Heidelberg, 2010, pp. 65–74.
[7] Y. K. Qawqzeh *et al.*, "Applying the big bang-big crunch metaheuristic to large-sized operational problems," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 2484–2502, Jun. 2020, doi: 10.11591/ijece.v10i3.pp2484-2502.
[8] H. M. Hasanien, "Performance improvement of photovoltaic power systems using an optimal control strategy based on whale optimization algorithm," *Electric Power Systems Research*, vol. 157, pp. 168–176, Apr. 2018, doi: 10.1016/j.epsr.2017.12.019.
[9] C. Dhaenens and L. Jourdan, *Metaheuristics for big data*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2016.
[10] A. S. Jaradat and S. B. Hamad, "Community structure detection using firefly algorithm," *International Journal of Applied Metaheuristic Computing*, vol. 9, no. 4, pp. 52–70, Oct. 2018, doi: 10.4018/IJAMC.2018100103.
[11] Z. Beheshti and S. M. Shamsuddin, "A review of population-based meta-heuristic algorithm," *International Journal of Advances in Soft Computing and its Applications*, vol. 5, no. 1, pp. 1–35, 2013.
[12] E. Hazir, E. S. Erdinler, and K. H. Koc, "Optimization of CNC cutting parameters using design of experiment (DOE) and desirability function," *Journal of Forestry Research*, vol. 29, no. 5, pp. 1423–1434, Sep. 2018, doi: 10.1007/s11676-017-0555-8.
[13] M. Song and D. Chen, "An improved knowledge-informed NSGA-II for multi-objective land allocation (MOLA)," *Geo-spatial Information Science*, vol. 21, no. 4, pp. 273–287, Oct. 2018, doi: 10.1080/10095020.2018.1489576.
[14] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.
[15] T. Zeugmann *et al.*, "Particle swarm optimization," in *Encyclopedia of Machine Learning*, Boston, MA: Springer US, 2011, pp. 760–766.
[16] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, pp. 1945–1950, doi: 10.1109/CEC.1999.785511.
[17] M. Dorigo and M. Birattari, "Ant colony optimization," in *Encyclopedia of Machine Learning and Data Mining*, Boston, MA: Springer US, 2017, pp. 56–59.
[18] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 1999, pp. 1470–1477, doi: 10.1109/CEC.1999.782657.
[19] Sourabh Joshi and Sarabjit Kaur, "Ant colony optimization meta-heuristic for solving real travelling salesman problem," in *Emerging Research in Computing, Information, Communication and Applications*, Singapore: Springer Singapore, 2016, pp. 55–63.
[20] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, Oct. 2007, doi: 10.1007/s10898-007-9149-x.
[21] H. Chiroma *et al.*, "Bio-inspired computation: Recent development on the modifications of the cuckoo search algorithm," *Applied Soft Computing*, vol. 61, pp. 149–173, Dec. 2017, doi: 10.1016/j.asoc.2017.07.053.
[22] X.-S. Yang and S. Deb, "Cuckoo search: recent advances and applications," *Neural Computing and Applications*, vol. 24, no. 1, pp. 169–174, Jan. 2014, doi: 10.1007/s00521-013-1367-1.
[23] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, Dec. 2012, doi: 10.1016/j.cnsns.2012.05.010.
[24] S. Ghafori and F. S. Gharehchopogh, "Advances in spotted hyena optimizer: a comprehensive survey," *Archives of Computational Methods in Engineering*, Jul. 2021, doi: 10.1007/s11831-021-09624-4.
[25] N. Rana, M. S. A. Latiff, S. M. Abdulhamid, and H. Chiroma, "Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments," *Neural Computing and Applications*, vol. 32, no. 20, pp. 16245–16277, Oct. 2020, doi: 10.1007/s00521-020-04849-z.
[26] N. Al-Madi, H. Faris, and S. Mirjalili, "Binary multi-verse optimization algorithm for global optimization and discrete problems," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 12, pp. 3445–3465, Dec. 2019, doi: 10.1007/s13042-019-00931-8.
[27] Z. Zhang, C. Huang, K. Dong, and H. Huang, "Birds foraging search: a novel population-based algorithm for global optimization," *Memetic Computing*, vol. 11, no. 3, pp. 221–250, Sep. 2019, doi: 10.1007/s12293-019-00286-1.
[28] A. Jaradat, B. Matalkeh, and W. Diabat, "Solving traveling salesman problem using firefly algorithm and k-means clustering," in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, Apr. 2019, pp. 586–589, doi: 10.1109/JEEIT.2019.8717463.
[29] S. Lin, "Computer solutions of the traveling salesman problem," *Bell System Technical Journal*, vol. 44, no. 10, pp. 2245–2269, Dec. 1965, doi: 10.1002/j.1538-7305.1965.tb04146.x.
[30] C. Rego, D. Gamboa, F. Glover, and C. Osterman, "Traveling salesman problem heuristics: Leading methods, implementations and latest advances," *European Journal of Operational Research*, vol. 211, no. 3, pp. 427–441, Jun. 2011, doi: 10.1016/j.ejor.2010.09.010.
[31] F. Chebihi, M. essaid Riffi, A. Agharghor, S. Cherif Bourki Semlali, and A. Haily, "Improved chicken swarm optimization algorithm to solve the travelling salesman problem," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 3, pp. 1054–1062, Dec. 2018, doi: 10.11591/ijeecs.v12.i3.pp1054-1062.
[32] C. Archetti, D. Feillet, M. Gendreau, and M. Grazia Speranza, "Complexity of the VRP and SDVRP," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 5, pp. 741–750, Aug. 2011, doi: 10.1016/j.trc.2009.12.006.
[33] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data*, Berlin/Heidelberg: Springer-Verlag, 2006, pp. 25–71.
[34] R. Xu and D. WunschII, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005, doi: 10.1109/TNN.2005.845141.
[35] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, Sep. 1999, doi: 10.1145/331499.331504.
[36] P. Kang and S. Cho, "K-means clustering seeds initialization based on centrality, sparsity, and isotropy," in *Intelligent Data Engineering and Automated Learning-IDEAL 2009*, Springer Berlin Heidelberg, 2009, pp. 109–117.
[37] J. Han, M. Kamber, and J. Pei, "Cluster analysis," in *Data Mining*, Elsevier, 2012, pp. 443–495.
[38] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1967, pp. 281–297.
[39] J. Zhang, L. Hong, and Q. Liu, "An improved whale optimization algorithm for the traveling salesman problem," *Symmetry*, vol. 13, no. 1, Art. no. 48, Dec. 2020, doi: 10.3390/sym13010048.
[40] R. Oftadeh, M. J. Mahjoob, and M. Shariatpanahi, "A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search," *Computers and Mathematics with Applications*, vol. 60, no. 7, pp. 2087–2098, Oct. 2010, doi:

10.1016/j.camwa.2010.07.049.
[41] W. A. Watkins and W. E. Schevill, “Aerial observation of feeding behavior in four baleen whales: eubalaena glacialis, balaenoptera borealis, megaptera novaeangliae, and balaenoptera physalus,” *Journal of Mammalogy*, vol. 60, no. 1, pp. 155–163, Feb. 1979, doi: 10.2307/1379766.
[42] “MP-TESTDATA-the TSPLIB symmetric traveling salesman problem instances.” MP-TESTDATA-The TSPLIB Symmetric Traveling Salesman Problem Instances (accessed Feb. 13, 2021).

BIOGRAPHIES OF AUTHORS

Anahita Sabagh Nejad is a Ph.D. Candidate at the University of Debrecen, Faculty of Informatics, with a background in Computer Science. She is interested in meta-heuristics and has experience in data mining, networking, and the theory of languages and machines as well. She can be contacted at email: Anahita.sabagh@inf.unideb.hu.

Dr. Gabor Fazekas is an Associate Professor at University of Debrecen, Faculty of Informatics, Department of Information Technology and its Theoretical Background. He has several experiences in supervising Ph.D. students. He is retired but still supervising Ph.D. students in different fields like computer systems, databases, combinatorial coding, didactic, and many more. He can be contacted at email: Fazekas.gabor@inf.unideb.hu.