Reducing the time needed to solve a traveling salesman problem by clustering with a Hierarchy-based algorithm

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

Article Info

Article history:

Received Sep 9, 2022
Revised Mar 6, 2023
Accepted Mar 10, 2023

Keywords:

Balanced iterative reducing and clustering using hierarchies
Metaheuristics
Swarm intelligence
Traveling salesman problem
Whale optimization

ABSTRACT

In this study, we compare a cluster-based whale optimization algorithm (WOA) with an uncombined method to find a more optimized solution for a traveling salesman problem (TSP). The main goal is to reduce the time of solving a TSP. First, we solve the TSP with the Whale optimization algorithm, later we solve it with the combined method of solving TSP which uses the clustering method, called BIRCH (balanced iterative reducing and clustering using hierarchies). Birch builds a clustering feature (CF) tree and then applies one of the clustering methods (for ex. K-means) to cluster data. Experiments performed on three datasets show that the convergence time improves by using the combined algorithm.

Corresponding Author:

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

1. INTRODUCTION

Meta-heuristics are methods for finding the search agents to approximate the exact solutions, and swarm-based algorithms use agents like Bees, Krills, and Whales. These algorithms are flexible so they can solve many problems like traveling salesman problem (TSP) much faster. Therefore researchers have found many meta-heuristics to solve difficult optimization problems. For the TSP, many advanced methods have been used to solve it in a shorter time. The cost function of TSP is minimized, which means finding the shortest path is the best answer. In our previous article (2022) [1], a meta-heuristic continuous optimization algorithm called the whale optimization algorithm (WOA) was combined with K-means to solve the TSP problem, but in this study, WOA is combined with the balanced iterative reducing and clustering using hierarchies (BIRCH) algorithm to find the shortest path. This new TSP solver is called WOA-BIRCH. In TSP, a traveling salesman wishes to visit exactly once each of a list of m cities (where the cost of traveling from city i to city j is C_{ij}) and then returns to the home city with minimum cost [2].

The metaheuristic algorithms have different characteristics because of their inspirations from natural or special biological behaviors. These algorithms need comprehensive tests from the benchmark functions to evaluate their performance. The WOA algorithm has shown good results compared with other meta-heuristics so it is being used in different fields of engineering such as [3]: optimizing the placement of capacitors, Making feature selection techniques, solving the economic dispatch problem, enhancing the performance of photovoltaic power systems, efficiently balancing energy production with the load demand [3], finding proper coefficients, cost minimization, and feature selection based on whale optimization algorithm (FSWOA) [4] with the aim to reduce the dimensionality of medical data [5] by the selection of a reduced feature set.

In this study, after some explanation about the BIRCH algorithm, once the problem solves with WOA, then we recommend the combined BIRCH algorithm as another solution for the same problem. Within the whale algorithm, we have three mathematical phases: encircling prey, spiral bubble-net feeding, and searching for prey [6]. In this algorithm, agents have a unique method to hunt which is called bubble-net feeding [7] as we have explained in our previous research. In section 2, some theories about the BIRCH algorithm, and in section 3 the research method with the pseudocode and some equations have been provided. In section 4 Experimental tests and in section 5 Results have been discussed in the form of some tables and figures, and in section 6, The research is finalized.

2. COMPREHENSIVE THEORIES

In this section, we want to explain some theories about the BIRCH algorithm with some of the equations which help us to understand the steps of the algorithm in the next sections. Clustering is a technique for grouping data into different subsets to find some information from data that were hidden. This method divides objects into subsets, based on the similarity(closeness) within the clusters and dissimilarity in the outer clusters. The issues with clustering algorithms can be summarized as scalability and the execution time for big data. Here, we want to explain the BIRCH algorithm [8] which is an integrated hierarchical clustering algorithm. Birch is a multi-stage method in the field of hierarchical clustering techniques that solves two hierarchical clustering difficulties, which are the scalability and rollback problems. Hierarchical clustering (HC) tends to merge the nearest pairs or divides the farthest pairs we call them agglomerative HC or divisive HC approaches respectively. BIRCH defines as a hierarchy-based technique used in data mining for solving real-life problems. When we insert the data into BIRCH, a clustering feature (CF) will be generated, such as each node showing a cluster, intermediate nodes representing superclusters, and the leaf nodes demonstrating the present clusters. The CF values summarize information about subclusters instead of storing all points [9]. The Branching (Br) parameter indicates the maximum number of children. When a new cluster is built, if it ends up with a greater Branching factor, the parent will divide.

The new point walks down recursively, always entering the subcluster which is the closest center until the walk reaches the leaf node. For having a balanced tree, the nodes split recursively. When all data are assigned, the leaf's centers will enter another clustering algorithm like k-means. This step merges the neighboring clusters to improve them. The advantages of this algorithm are its high performance in terms of memory, execution time, quality of the clusters, stability, and scalability, parallel and concurrent clustering, and its interactively and dynamically tune performance; so we combine it with WOA to use it as a TSP solver. In fact, BIRCH handles an accumulated section collectively by storing a compact CF tree. The dense regions are actually called subclusters. The algorithm operates in the following four steps [10]: i) scans all data and builds CF tree (loading), ii) condensing the original tree by building a smaller CF tree, iii) applying a clustering algorithm for all leaf entries, and iv) refining the clusters format(optional). For N d-dimensional data [11], the clustering feature, CF, the cluster's centroid, x0, radius, R, and diameter, D, are [12]:

$$CF = (n, LS, SS) \tag{1}$$

$$x_0 = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{LS}{n} \tag{2}$$

$$R = \sqrt{\frac{\sum_{i=1}^{n}(x_i - x_0)^2}{n}} = \sqrt{\frac{nSS - 2LS^2 + nLS}{n^2}} \tag{3}$$

$$D = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2}{n(n-1)}} = \sqrt{\frac{2nSS - 2LS^2}{n(n-1)}} \tag{4}$$

The WOA algorithm that we want to combine with our algorithm is summarized in the research method section in the form of pseudocode in Algorithm 1 and Table 1, the parameters of the WOA algorithm are defined [13]. In these parameters, **a** which is a kind of distance parameter affects the ability of exploration and exploitation as a decreasing number from 2 to 0 and results in fast convergence.

Algorithm 1 Pseudocode of the whale optimization algorithm

```
(1) Generate the initial population for Xi (i=1, 2, ..., n) [13] [14]
(2) Compute the fitness of all the candidate solutions individually
(3) While termination is not met, for i= 1:n do [13] [14]
    (4) Update a, A, C, I, and P when $\overrightarrow{A} = 2\overrightarrow{a}.\overrightarrow{r} - \overrightarrow{a}$ and
    $\overrightarrow{C} = 2.\overrightarrow{r}$ [15]
    (5) for j=1:n do
        (6) If $p < 0.5$ [16] , and $|A| \leq 1$, then
            (7) Compute:
            $\overrightarrow{D} = \overrightarrow{C}.X^*(t) - \overrightarrow{X}(t), and\, \overrightarrow{X}(t+1) = \overrightarrow{X^*}(t) - \overrightarrow{A}.\overrightarrow{D}$ [17]
        (8) else if ( $|A| \geq 1$ )
            (9) Select a random individual ($X_{rand}$ ) [18]
            (10) Compute
            $\overrightarrow{D} = \left|\overrightarrow{C}.\overrightarrow{X_{rand}} - \overrightarrow{X}\right| and, \overrightarrow{X_{t+1}} = \overrightarrow{X_{rand}} - \overrightarrow{A}.\overrightarrow{D}$
        (11) end if
        (12) else if $P \geq 0.5$ then [19]
            (13) Compute
            $\overrightarrow{X}(t+1) = \overrightarrow{D'}.e^{bl}.cos(2\pi l) + \overrightarrow{X^*}(t)$
            [18] (14) end if
    (15) end for
    (16) Check search agents, Compute the fitness for $X_i$, and update X* [20].
(17) end for
(18) end while
```

Table 1. Parameter definition

Parameter	Definition
a	Decreasing number from 2 to 0
A	CoefficientA= [-a, a]
r	Random number[0,1]
X	The whales position
X*	The best solution
P	Random number [0, 1]
b	A constant for logarithmic shape
l	A value in [-1,1]
D	Distance

3. RESEARCH METHOD

The partitioning technique is one of the foremost important parts of cluster analysis, and lots of algorithms are implemented for it like K-means, K-means++, and K-medoids [21]. K-means first selects K initial seeds, then assigns the set of points into K clusters by minimizing the sum of squared error (SSE) [22]. We apply K-means in phase 3 of the BIRCH pseudocode [23].

3.1. Pseudocode of BIRCH

Here, the five phases of BIRCH are summarized: 1) Search all the data to make a tree as in (1). 2) Rebuild the CF tree with a bigger T (tree size). 3) Apply K-means or K-modes. 4) Cluster refining by making some additional passings over the data for the new reassignments based on the closeness to the centroids. 5) Repeat all the steps to create K number of clusters.

3.2. Equations for the BIRCH algorithm

In the previous equations, the compactness of the clusters defines by *R*, and *D*, such *R* [24] is the average distance from member points to the centroid x0, and *D* is the average pairwise distance within a cluster, and CF entry is a triple, [12] where N is the number of data points, LS (Linear Sum of N), and SS (square sum of N) defines our subclusters. Here is an example of how two CFs can be merged which is called the **additivity theorem** [25]:

$$CF_1 = \langle n_1, LS_1, SS_1 \rangle \tag{5}$$

$$CF_2 = \langle n_2, LS_2, SS_2 \rangle \tag{6}$$

$$CF_1 + CF_2 = \langle n_1 + n_2, LS_1 + LS_2, SS_1 + SS_2 \rangle \tag{7}$$

if we have two centroids for two clusters called x0 and x1, one of the equations of distance measurement that can be used is Euclidean distance [26]:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{8}$$

there exist some other distance measurements like Manhattan and city block. We have two measurements for quality called Q1 (mean of the radius), and Q2 (mean of the diameter). These quality measurements can be used for pre-processing [27]:

$$Q_1 = \frac{\sum_{i=1}^{k} n_i R_i^2}{\sum_{i=1}^{k} n_i} \tag{9}$$

$$Q_2 = \frac{\sum_{i=1}^{k} n_i \left(n_i - 1\right) D_i^2}{\sum_{i=1}^{k} n_i \left(n_i - 1\right)} \tag{10}$$

4. EXPERIMENTAL TESTS

In Table 2 the values of our parameters, and Tables 3 to 6 the result of the experimental tests for our two combined algorithms have shown. We perform experiments on a MacBook with 8 GB of RAM, and 8 core CPU running MATLAB R2019a for both algorithms. This algorithm runs itself until the termination criteria are met. We can use some termination criteria, like a predefined number of iterations (here 20 iterations), sustainability of the results and time limit. The results obtained and shown in the tables are averaged over 20 runs of the proposed model for our datasets which are Ali535, Rat783, and pr1002 from **TSPLIB TESTDATA** [28]. The execution time for both of the algorithms is calculated using a tic-toc command.

5. RESULTS AND DISCUSSIONS

Here, the steps of our algorithm start with setting the parameter. The other parameters like the branching factor can be decided as (11), but these numbers are experimental. For the threshold, we have used the standard deviation of our data model which consists of x and y for the dimensions of our dataset. Again, this formula can change based on experience. The next equations are variance and standard deviation, but we have used (14) by experience.

1: Set the parameters as shown in Table 2:

$$Branching\_Factor = \frac{N}{5} \tag{11}$$

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{\mu})^2 \tag{12}$$

$$S = \sqrt{\sigma}^2 \tag{13}$$

$$S = 5 * \sigma(\sigma(D)) \tag{14}$$

2: Specify K randomly or the below equation for big tours among the cities:

$$k = \sqrt{\frac{N}{10}} \tag{15}$$

which N is the city's length, or we can also divide it by 2 (the standard formula). We use the same technique For the Branching factor.

```
3: Applying the BIRCH algorithm to cluster our data.
4: Applying the WOA algorithm for all of the found clusters i =1:K.
5: Find the location of agents.
6: Sorting by indexing.
7: Joining the clusters by finding the cities that are closer to the centroid of that cluster.
8: Repeat till joining all the clusters.
```

Table 2 mentions some parameters consisting of the initial population, iteration number, and city numbers. In the other tables like Table 3 and Table 4, some statistical values of fitness function (cost function) for unclustered and clustered WOA are mentioned. In Table 5 and Table 6, the execution time for the unclustered and clustered WOA are shown, respectively, and the same statistical calculations are considered for all three datasets. Based on the results, the fitness function is not changed much but based on the time average for Tables 5 and 6, the time has improved for the new method (WOA-BIRCH).

Table 2. The parameter settings

Number of initial population	Iterations	Number of the cities
100	20	535
100	20	783
100	20	1002

Table 3. The fitness function for the unclustered whale optimization algorithm

Dataset	Min	Max	Average	Stdev.s
Ali535	35333.53	36400.23	35903.55	46913.33161
Rat783	1.2895e+05	1.3475e+05	1.3169e+05	1.4941e+03
Pr1002	3.9822e+06	4.1810e+06	4.0951e+06	46913.33

Table 4. The fitness function for the clustered whale with BIRCH algorithm

Dataset	Min	Max	Average	Stdev.s
Ali535	35763.49	36702.4	36197.19	49571.8325
Rat783	1.1233e+05	1.1434e+05	1.1329e+05	642.4314
Pr1002	3.2389e+06	3.4310e+06	3.3316e+06	49571.83

Table 5. The execution time of the unclustered whale optimization algorithm

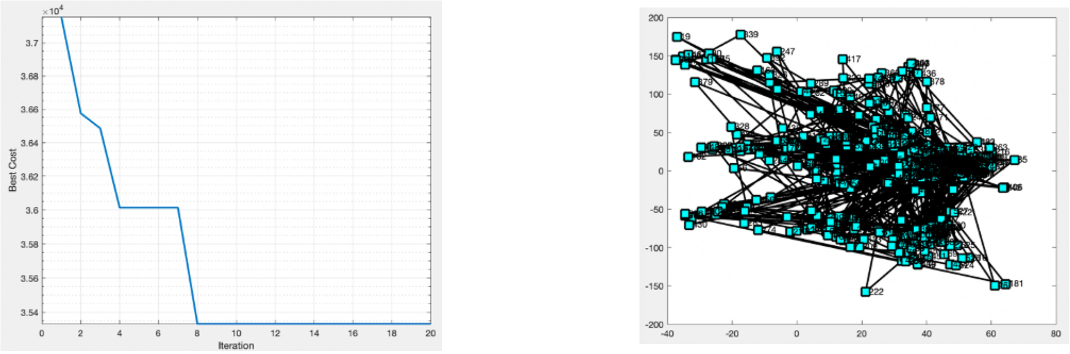
Dataset	Min	Max	Average	Stdev.s
Ali535	8.95	9.13	9.03	0.04
Rat783	12.98	13.21	13.05	0.06
Pr1002	16.19	16.50	16.31	0.07

Table 6. The execution time of the clustered whale optimization with BIRCH algorithm

Dataset	Min	Max	Average	Stdev.s
Ali535	4.11	4.29	4.15	0.04
Rat783	5.15	5.26	5.18	0.02
Pr1002	6.46	6.56	6.48	0.07

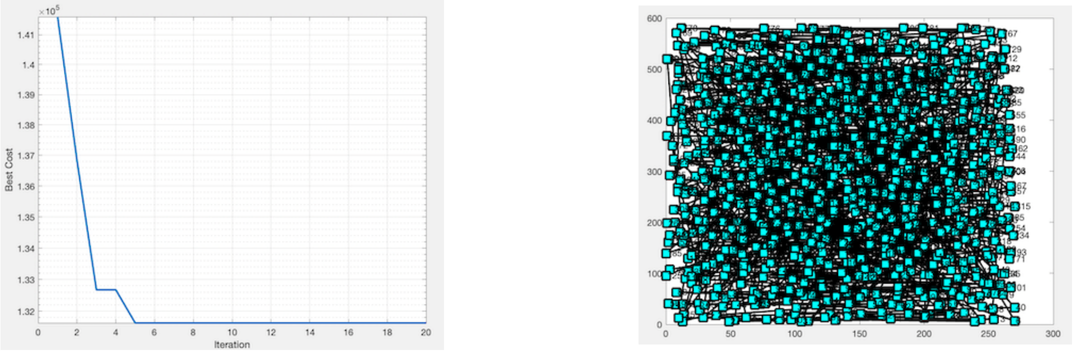
5.1. Figures

Figures 1(a) shows the best cost function of an unclustered approach for the ali535 dataset in 20 iterations, and Figure 1(b) shows the connected cities in an unclustered approach for the same dataset. Figure 2(a) shows the cost function for the Rat783 dataset, and Figure 2(b) shows how these cities are connected. Figure 3(a) shows the cost function of an unclustered approach for the pr1002 datasets during the iterations, and Figure 3(b) shows all the cities which are connected through an unclustered approach for finding the best path. In Figure 4(a) the WOA-BIRCH is applied for clustering with the BIRCH algorithm for the Ali535 dataset, and in Figure 4(b) all the subgraphs (before connections) are ilustrated for the same dataset. In Figure 5(a) the WOA-BIRCH is applied for clustering with the BIRCH algorithm for the Rat783 dataset, and subgraphs are shown in Figure 5(b). This figures is more complicated because of the number of cities (783). In Figure 6(a), the WOA-BIRCH is applied for clustering with the BIRCH algorithm for the Pr1002 dataset, and in Figure 6(b), we have an incomplete graph to show the connections before the clustering with BIRCH.



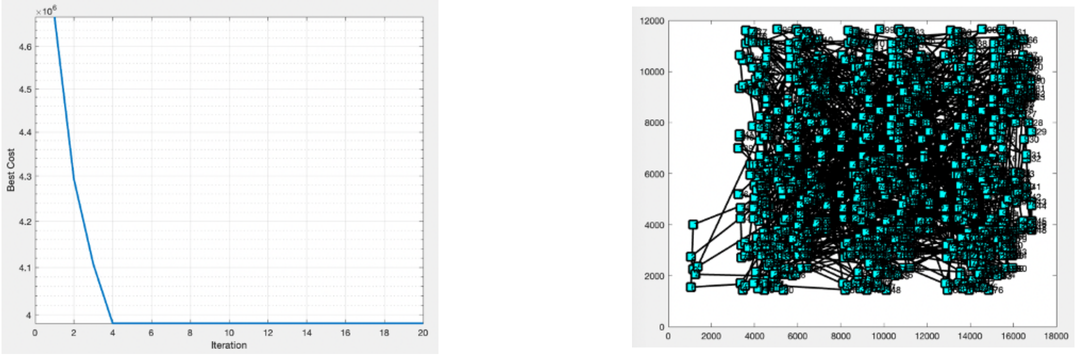
(a) (b)

Figure 1. Effects of selecting the WOA algorithm to solve TSP for ali535, (a) the values of the cost function for ali535 and (b) unclustered approach



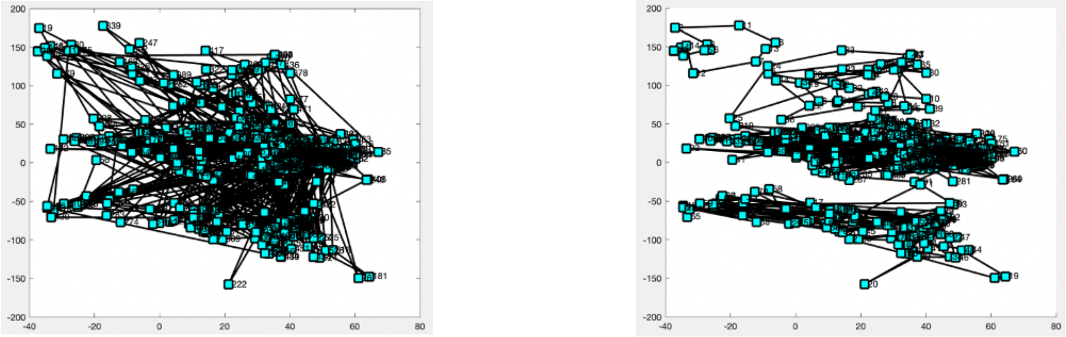
(a) (b)

Figure 2. Effects of selecting the WOA algorithm to solve TSP for Rat783, (a) the values of the cost function for Rat783 and (b) unclustered approach



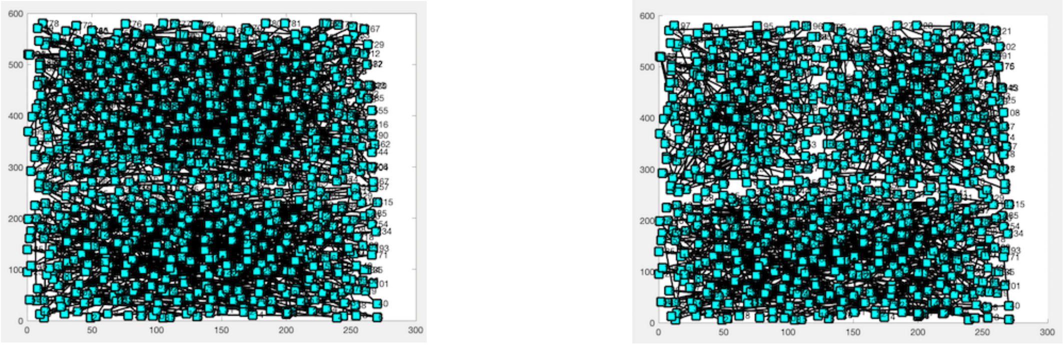
(a) (b)

Figure 3. Effects of selecting the WOA algorithm to solve TSP for pr1002, (a) the values of cost function for pr1002 and (b) unclustered approach



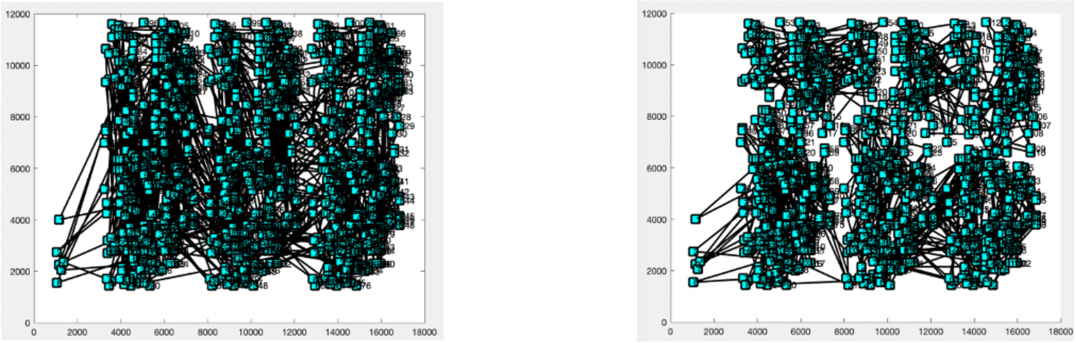
(a) (b)

Figure 4. Effects of selecting the combined BIRCH and WOA algorithm to solve TSP for ali535, (a) connected clusters and (b) the unjoined optimal clusters



(a) (b)

Figure 5. Effects of selecting the combined BIRCH and WOA algorithm to solve TSP for Rat783, (a) connected clusters and (b) the unjoined optimal clusters



(a) (b)

Figure 6. Effects of selecting the combined BIRCH and WOA algorithm to solve TSP for pr1002, (a) connected clusters and (b) the unjoined optimal clusters

6. CONCLUSION

In this paper, our data is clustered by the BIRCH algorithm to speed up the execution time of the WOA algorithm to solve TSP. The observations show that the convergence time is improved. The idea was to cluster our data by the BIRCH algorithm, then solve each part by WOA and finally connect those clusters to complete a tour. Based on the final results, the TSP solver WOA-BIRCH is faster than the whale algorithm for TSP, therefore we can conclude that applying the BIRCH algorithm has empowered our algorithm and made it more applicable. The comparison measurements certify the improvement of our algorithm.

ACKNOWLEDGEMENT

This paper was supported by the University of Debrecen under OpenA/DEENK/104/2023. My deepest appreciation is dedicated to my supervisor, Dr.Gabor Fazekas for the encouragement and advice throughout this project.

REFERENCES

[1] A. S. Nejad and G. Fazekas, "Solving a traveling salesman problem using meta-heuristics," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 1, pp. 41–49, Mar. 2022, doi: 10.11591/ijai.v11.i1.pp41-49.

[2] K. L. Hoffman, M. Padberg, and G. Rinaldi, "Traveling salesman problem," *Encyclopedia of Operations Research and Management Science*, vol. 1, pp. 1573–1578, 2013, doi: 10.1007/978-1-4419-1153-7_1068.

[3] M. Alshayeji, B. Behbehani, and I. Ahmad, "Spark-based parallel processing whale optimization algorithm," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 4, p. e6607, Sep. 2022, doi: 10.1002/cpe.6607.

[4] 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, Mar. 2020, doi: 10.1007/s00521-020-04849-z.

[5] H. F. Eid, "Binary whale optimisation: an effective swarm algorithm for feature selection," *International Journal of Metaheuristics*, vol. 7, no. 1, p. 67, 2018, doi: 10.1504/ijmheur.2018.091880.

[6] W. Z. Sun, J. S. Wang, and X. Wei, "An improved whale optimization algorithm based on different searching paths and perceptual disturbance," *Symmetry*, vol. 10, no. 6, p. 210, Jun. 2018, doi: 10.3390/sym10060210.

[7] R. O. Bawazir and N. S. Cetin, "Comprehensive overview of optimizing PV-DG allocation in power system and solar energy resource potential assessments," *Energy Reports*, vol. 6, pp. 173–208, Nov. 2020, doi: 10.1016/j.egyr.2019.12.010.

[8] G. Ramadevi, S. Yeruva, P. Sravanthi, P. E. Vamsi, and S. J. Prakash, "Analysis and detection of diabetes using data mining techniques – efficiency comparison," *International Journal of Scientific Research in Science and Technology*, vol. 7, no. 4, pp. 73–79, Jul. 2021, doi: 10.32628/cseit217425.

[9] X. Xu, M. Ester, H. P. Kriegel, and J. Sander, "Clustering and knowledge discovery in spatial databases," *Vistas in Astronomy*, vol. 41, no. 3, pp. 397–403, Jan. 1997, doi: 10.1016/S0083-6656(97)00044-5.

[10] N. Shetty and R. Shirwaikar, "A comparative study: BIRCH and clique," *International Journal of Engineering Research Technology (IJERT)*, vol. 2, no. 11, pp. 2019–2023, 2013, doi: 10.17577/IJERTV2IS110610.

[11] X. L. Ma et al., "DHC: distributed, hierarchical clustering in sensor networks," *Journal of Computer Science and Technology*, vol. 26, no. 4, pp. 643–662, Jul. 2011, doi: 10.1007/s11390-011-1165-0.

[12] P. K. Prasad and C. P. Rangan, "Privacy preserving BIRCH algorithm for clustering over vertically partitioned databases," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4165 LNCS, Springer Berlin Heidelberg, 2006, pp. 84–99.

[13] Y. Li et al., "Intradialytic hypotension prediction using covariance matrix-driven whale optimizer with orthogonal structure-assisted extreme learning machine," *Frontiers in Neuroinformatics*, vol. 16, p. 956423, Oct. 2022, doi: 10.3389/fninf.2022.956423.

[14] Q. Jiang, Y. Guo, Z. Yang, Z. Wang, D. Yang, and X. Zhou, "Improving the performance of whale optimization algorithm through OpenCL-based FPGA accelerator," *Complexity*, vol. 2020, pp. 1–15, Dec. 2020, doi: 10.1155/2020/8810759.

[15] S. Mirjalili, J. S. Dong, and A. Lewis, "Nature-inspired optimizers," *Studies in Computational Intelligence*, vol. 811, pp. 7–20, 2020, doi: 10.1007/978-3-030-12127-3.

[16] X. Lin, P. Zhang, F. Meng, and C. Liu, "A coal seam thickness prediction model based on CPSAC and WOA–LS-SVM: a case study on the ZJ mine in the huainan coalfield," *Energies*, vol. 15, no. 19, p. 7324, Oct. 2022, doi: 10.3390/en15197324.

[17] D. B. Prakash and C. Lakshminarayana, "Optimal siting of capacitors in radial distribution network using whale optimization algorithm," *Alexandria Engineering Journal*, vol. 56, no. 4, pp. 499–509, Dec. 2017, doi: 10.1016/j.aej.2016.10.002.

[18] S. Mostafa Bozorgi and S. Yazdani, "IWOA: an improved whale optimization algorithm for optimization problems," *Journal of Computational Design and Engineering*, vol. 6, no. 3, pp. 243–259, Feb. 2019, doi: 10.1016/j.jcde.2019.02.002.

[19] A. Abraham, S. K. Shandilya, L. Garcia-Hernandez, and M. L. Varela, *Hybrid intelligent systems: 19th International Conference on Hybrid Intelligent Systems (HIS 2019) held in Bhopal, India, December 10-12, 2019, series advances in intelligent systems and computing*, vol. 17. Springer International Publishing, 2020.

[20] Q. Wang, Y. Tian, L. Lin, R. Vanga, and L. Ma, "An improved whale algorithm for setting standard scheduled block time based on the airline f," *Discrete Dynamics in Nature and Society*, vol. 2020, pp. 1–14, Oct. 2020, doi: 10.1155/2020/6049062.

[21] M. da S. Nunes, M. C. Júnior, G. F. Miranda, and B. T. Andrade, "An approach to preprocess and cluster a BRDF database," *Graphical Models*, vol. 119, p. 101123, Jan. 2022, doi: 10.1016/j.gmod.2021.101123.

[22] J. Qi, Y. Yu, L. Wang, J. Liu, and Y. Wang, "An effective and efficient hierarchical K-means clustering algorithm," *International Journal of Distributed Sensor Networks*, vol. 13, no. 8, p. 17, Aug. 2017, doi: 10.1177/1550147717728627.

[23] A. Selamat, N. T. Nguyen, and H. Haron, *Intelligent information and database systems: 5th Asian Conference, ACIIDS 2013, Kuala Lumpur, Malaysia, March 18-20, 2013, Proceedings, Part I*, vol. 7802 LNAI, no. PART 1. Springer Berlin Heidelberg, 2013.

[24] L. Cao, J. Z. Huang, J. Bailey, Y. S. Koh, and J. Luo, *New frontiers in applied data mining: PAKDD 2011 International Workshops, Shenzhen, China, May 24-27, 2011, Revised Selected Papers*, vol. 7104, no. September 2014. Springer Berlin Heidelberg, 2012.

[25] J. Han, M. Kambe, and J. Pie, *Data mining: concepts and techniques.* Elsevier, 2012.

[26] M. Affenzeller, S. Winkler, S. Wagner, and A. Beham, *Genetic algorithms and genetic programming: modern concepts and practical applications.* CRC Press, 2009.

[27] H. Okumura and K. Naito, "Weighted kernel estimators in nonparametric binomial regression," *Journal of Nonparametric Statistics*, vol. 16, no. 1–2, pp. 39–62, Feb. 2004, doi: 10.1080/10485250310001624828.

[28] S. George, "Mp-testdata—the tsplib symmetric traveling salesman problem instances." 2008.

BIOGRAPHIES OF AUTHORS

Dr. Anahita Sabagh Nejad She obtained Bachelor's Degree in Information Technology in 2013. Later, in 2016, she got her MSc in Computer Science from the University of Debrecen. She has been a Ph.D. candidate since 2018. Research interest: artificial intelligence, computational theory, data mining, meta-heuristics, etc. Her researches are in the field of evolutionary algorithms. She can be contacted at Email: anahita.sabagh@inf.unideb.hu

Dr. Gabor Fazekas is an Associate Professor at the University of Debrecen, Informatics Department who has so many experiences in 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.