WSN Data Gathering Using TSP-Modified RNNs and Horse Herd Algorithm

Haider Abdulkarim
College of Communications Engineering, University of Technology, Baghdad, Iraq
haider.a.abdulkarim@uotechnology.edu.iq (corresponding author)

Marwa K. Farhan
Scholarships and Culture Relations Directorate, Ministry of Higher Education and Scientific Research, Baghdad, Iraq
marwa.farhan@scrdiraq.gov.iq

Mustafa Ghanim
College of Communications Engineering, University of Technology, Baghdad, Iraq
mustafa.g.rzooki@uotechnology.edu.iq

Ayman N. Muhi
College of Communications Engineering, University of Technology, Baghdad, Iraq
aymen.n.muhi@uotechnology.edu.iq

Mohaimen Q. Algburi
College of Communications Engineering, University of Technology, Baghdad, Iraq
mohaimen.q.khalaf@uotechnology.edu.iq

Received: 2 May 2025 | Revised: 15 May 2025 | Accepted: 22 May 2025

 DOI: https://doi.org/10.48084/etasr.11864

ABSTRACT

Efficient data-gathering algorithms ensure efficient operation of Wireless Sensor Networks (WSNs) and prevent data loss. This study proposes the Traveling Salesman Problem (TSP) algorithm to gather WSN data. Two variants of the TSP are proposed and implemented, Recurrent Neural Network (RNN) and Horse Herd Optizations (HHO), using the remaining node energy as a second factor in selecting the shortest data-gathering path in addition to node distance. The simulation results show that TSP based on RNN and the weighted sum of both distance and energy outperforms the classic TSP algorithm, shortening the overall path and maximizing the WSN lifetime.

Keywords-traveling salesman problem; wireless sensor network; data gathering; recurrent neural network; horse herd optimization

I. INTRODUCTION

In recent years, Wireless Sensor Networks (WSNs) have been identified as an important technology, with great prospects for a wide range of applications from the scientific field to home automation. One of the most common deployment patterns is the formation of ad hoc WSNs, where nodes are deployed without being associated with any central base station to transmit WSN data traffic. The use of ad hoc WSNs prevents the need for a base station to be fixed as a sink throughout network operation and can reduce deployment costs [1]. Most WSNs know in advance the location of the deployed nodes. However, sensor nodes are often deployed in inaccessible or harsh environments, which limits typical manual deployment. In military fields or environmental applications, nodes are often carried by the wind or animals and, hence, they are not placed accurately. Ideally, the nodes should be distributed across the entire area to be monitored.

Sensor nodes are deployed into a vast sensing field without the need for a previous infrastructure to form WSNs. Sensor nodes aim to collect data at regular intervals, convert them into digital data, and then transmit a signal to the base node or the sink. Sensor nodes must establish a network and identify their neighbors before monitoring the environment. Energy use occurs when the data is uploaded to the Mobile Collector (MC) and the sensing process.

There are two categories of sensor networks: heterogeneous and homogeneous. In homogeneous networks, all nodes have the same energy and capabilities, and such networks fall within flat or hierarchical topologies. The sensors near the static sink in the flat topology use more energy than those near the network's edge. Sensor nodes are distributed to monitor various areas in certain applications, and the network can be cut off. The shortcomings of flat topology can be addressed by utilizing hierarchical topology or clusters. In such applications, the cluster heads are in the top layer, and the collecting nodes make up the lower layer [2]. Data are collected from the lower layers and forwarded to the sink by the cluster head, which can serve as a hub for activity [3]. The cluster head uses more energy than the other nodes because it collects and forwards data from the lower nodes. However, it is possible to reduce energy consumption by dynamically rotating sensor nodes. Heterogeneous networks are characterized by a large number of basic nodes with minimal resources and a small number of resource-rich nodes. Nodes with high resources can become clusters, whereas nodes with limited resources are restricted to simple nodes with limited communication capabilities. Mobile data gathering uses one or more MCs. The MC is a device with a strong transceiver and large battery capacity that uses short-range communications to collect data. MC travels through the sensing field, collecting data while in motion and occasionally pausing to obtain data from sensors [4].

There are two other types of clustered sensor networks: single-hop and multi-hop. In a single-hop network, sensor nodes communicate with the cluster head by single-hopping. The MC needs to cover every sensor node in the field's transmission range to save the most energy possible. It is easier for the MC to collect data packets in a single hop. The MC's route through the sensing field could be preset or random. However, its mobility reduces the energy usage of the network. In a single-hop relay, each sensor speaks directly to the sink [5]. However, this might not be possible in wide geographic areas due to the high transmission power required. In WSNs, multi-hop routing refers to sensors that act as relays for other sensor nodes. Data packets are routed through multiple hop relays between sensors and sent to the data sink. The multi-hop data packet forwarding process uses more energy.

For MCs, to implement data gathering in a non-redundant fashion, it is important to avoid revisiting a single node more than once. One of the most important algorithms that describe a loop-free tour is the Traveling Salesman Problem (TSP) [6]. This problem was defined by T. Kirkman and W. R. Hamilton in the 1800s, defined as a salesman who must travel between N cities. As long as he visits each location once along his journey and ends up where he started, he does not care what order he goes in. Every city has road, rail, or air transportation connections to nearby cities or nodes, and there are weights (or costs) associated with each link connecting them. Cost indicates how difficult it is to navigate this edge of the graph. The salesman aims to minimize his travel expenses while also minimizing the distance he travels [7-9].

This study proposes a TSP data-gathering model for an MC WSN and validates it through simulations. The TSP implementation examined two different algorithms: Recurrent Neural Network (RNN) and Horse Herd Optimization (HHO). This combination leverages the strengths of both methods: the structured optimization capabilities of TSP algorithms (such as heuristics or exact methods) and the sequential learning and pattern recognition abilities of RNNs. Hybrid models are highly adaptable to dynamic large-scale WSN structures.

II. METHOD

A. Data Gathering in WSN

In a WSN, data gathering is the process of compiling readings from sensor nodes, regularly directing them toward a sink, and guiding them in the direction of the base station. Due to the limited energy source of the WSN nodes and the power consumption of each node, efficient data-gathering algorithms need to be designed and implemented. These algorithms should extend the maximum lifetime of a WSN while keeping the total path of collected packets (sensor reading) at a minimum [5, 10]. However, few efficient data-gathering algorithms have been investigated in this context. This study presents a novel method to collect WSN data using the TSP. The proposed data-gathering TSP is implemented using an RNN and the horse herd algorithm, as shown in Figure 1.

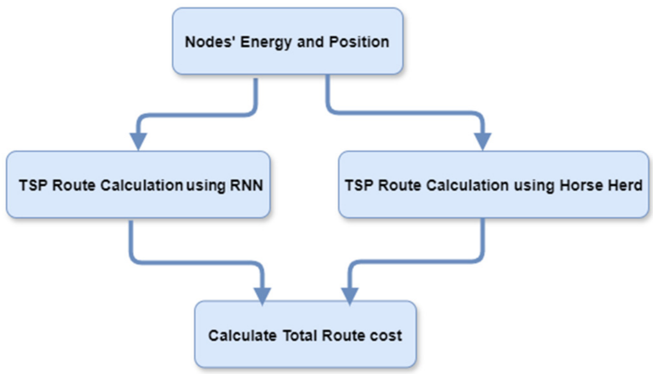

Fig. 1. Flowchart of the proposed method.

B. Recurrent Neural Network (RNN)

1) Structure

For data with a sequential structure, RNNs are frequently implemented for solving different problems [11-13]. For example, time series data are inherently ordered according to time. RNNs differ from traditional Multi-Layer Perceptron (MLP) networks in two aspects: They consider historical input data, and they share parameters and weights in all networks. In addition, RNNs differ from Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs) in that RNNs do not have fixed input and output weights (after learning) and strongly rely on the previous and the current input. Hence, they are widely used for sequential data analysis, such as speech recognition [14].

Figure 2 shows the RNN architecture, where h_t is a vector of hidden node state outputs, and t corresponds to the output time stamp. The matrices W_{hz}, W_{hh}, and W_{xh} correspond to the weights between hidden-output, hidden-hidden (recursive part), and input-hidden nodes, respectively. The vector x_t is the input sequence (at time *t*) and z_t is the output sequence (at time *t*). The RNN model involves three main calculations [15, 16]:

$$\rho = a_t = W_{hh}h_{t-1} + W_{xh}x_t \tag{1}$$

$$h_t = tanh(a_t) \tag{2}$$

$$z_t = softmax(W_{hz}h_t) \tag{3}$$

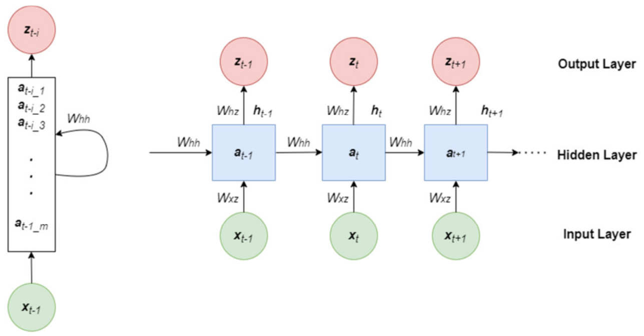

Fig. 2. (a) The recursive nature of an RNN, (b) An RNN model in a timely sequential manner (unfolded).

The size of each matrix and vector is given in Table I.

TABLE I. CORRESPONDING SIZES OF RNN ENTITIES

Vector / Matrix	Size
Hidden-hidden weights matrix W_{hh}	*m×m*
Input-hidden weights matrix W_{xh}	*m×n*
Hidden-output weights matrix W_{hz}	*n×m*
Input vector x_t	*n×1*
Hidden layer states output h_t	*m×1*
Activation function of hidden states a_t	*m×1*
Output layer z_t	*n×1*

2) *Weight Updates*

As mentioned earlier, the RNN is used for sequential data. This data dependency also applies to finding the optimal shortest path to collect data from all nodes in a WSN. Therefore, a modification of the backpropagation training method is used to update the network weights, called Backpropagation Through Time (BPTT). This method unfolds the RNN to build a conventional feedforward neural network to use backpropagation. The loss function $L(z, y)$ is defined as the total cost of traveling between the current search route z_t and the previously found minimum cost paths y_t. The loss function is given by [17]:

$$L(z, y) = \sum_{t=1}^{T} \|z_t - y_t\| \tag{4}$$

where || || denotes the Euclidean distance. The BPTT is carried out at each time point. The derivative of the loss $L(z, y)$ is calculated with respect to each of the weight matrices, W_{hz}, W_{hh}, and W_{xh} using a partial derivative. In BPTT, the loss is calculated over the entire sequence, and the network is unrolled across time steps to form a computational graph. The partial derivative of the loss with respect to a weight is computed by applying the chain rule across all time steps. For weights in the hidden-to-hidden weight matrix, the partial derivative captures how a small change in weight at any time step affects the loss through its influence on the hidden states and subsequent outputs. Mathematically, this involves summing the contributions of the weight to the loss across all time steps, as the same weight is used repeatedly in the sequence. This could be summarized as [18-20]:

$$\frac{\partial L}{\partial W_{hz}} = \sum_{t=1}^{T} \frac{\partial l_t}{\partial z_t} \cdot \frac{\partial z_t}{\partial (softmax)} \cdot h_t \tag{5}$$

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial l_t}{\partial z_t} \cdot \frac{\partial z_t}{\partial (softmax)} \cdot W_{hz} \sum_{k=1}^{t} (W_{hh}^T)^{t-k} \cdot h_k \tag{6}$$

$$\frac{\partial L}{\partial W_{xh}} = \sum_{t=1}^{T} \frac{\partial l_t}{\partial z_t} \cdot \frac{\partial z_t}{\partial (softmax)} \cdot W_{hz} \sum_{k=1}^{t} (W_{hh}^T)^{t-k} \cdot x_k \tag{7}$$

Figure 3 shows a concise flowchart of the RNN algorithm.

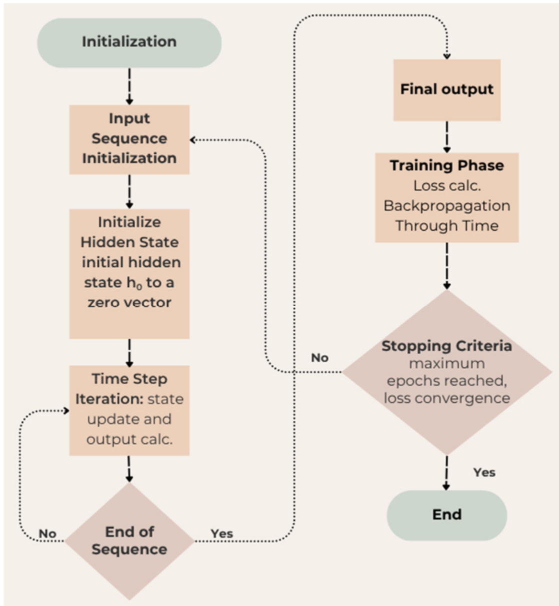

Fig. 3. Flowchart of the RNN.

C. *Proposed Modification for Hybrid Distance-Energy*

In the standard RNN, the forward pass calculates the score h_t vector for the next cites to visit as:

$$h_t = \sum_t^T tanh(W_{xh} * city_{xy_t} + b_h) \tag{8}$$

where b_h is the hidden-layer bias vector. Then the output layer is calculated as:

$$y_t = W_{hy} \times h_t + b_y \tag{9}$$

where b_y is the output-layer bias vector.

This study proposes the following scenario. Each node in the WSN (represented as a candidate city to visit) has a known remaining energy level. The knowledge of each node's energy is assumed to be shared at least within the neighboring nodes. Therefore, an energy vector E is defined at the start of the algorithm. In addition, the data harvesting scenario is assumed to explore the neighboring nodes. The selection of the next node is based on two factors:

1. The nearest node $Node_i$ position (represented as $city_{xy}$), where xy are the xy coordinates.
2. The node $Node_i$ with the least remaining energy, represented by E_i.

The reason for choosing the nearest node is to select the path with the minimum total energy cost. In addition, the selection of the nodes with minimum energy is to ensure that their data are collected as soon as possible before their batteries are totally depleted. Therefore, both the score and output calculations are proposed as follows:

$$h_{t_{Hybrid}} = \sum_t^T \tanh\{(v_1 \times W_{xh} * city_{xy_t} + b_h) + (v_2 \times W_{xh} * E_i + b_h)\} \quad (10)$$

$$y_t = \{(W_{hy} \times h_{t\_Hybrid} + b_y) \quad (11)$$

where $h_{t_{Hybrid}}$ represents a hybrid combination of the importance of both position and energy, determined by the weights v_1 and v_2, according to:

$$\sum v_1 + v_2 = 1 \quad (12)$$

D. Horse Herd Optimization (HHO) Algorithm

Inspired by the social dynamics and movement patterns of horse herds, the relatively new HHO algorithm is a metaheuristic optimization technique applied to complicated optimization problems, especially those involving continuous optimization. The algorithm imitates how a herd of horses travel together, obey a selected leader (alpha horse), and adapt to their surroundings to identify the best places for the horses to graze (best solutions). Horses represent each possible solution to the optimization problem. In addition, a group of horses represents the population in the search space. Based on their level of fitness, certain horses are selected as leaders, while others follow them. The algorithm searches for a balance between exploitation (improving already-found solutions) and exploration (exploring new regions of the solution space) [21-23]. Similar heuristic optimization algorithms have been used in similar problems, such as energy clustering or routing in WSNs [24, 25], but they need to be extensively tested and adapted for the TSP with multi-objective parameters. Here, the positions of the alpha horses in the herd (best solutions), representing the shortest tour path, are used to calculate the fitness value:

$$f_i = \sum_{t=1}^T city_{xy_t} \quad (13)$$

and the leader alpha horse α_{horse} is selected as:

$$\alpha_{horse} = min(f_i) \quad (14)$$

In the proposed method, the leader is selected based on two criteria: route length and remaining energy:

$$f_i = v_1 \times \sum_1^T city_{xy_t} + v_2 \times \sum_1^T E_{i_{meanQuarter}} \quad (15)$$

where $E_{i_{meanQuarter}}$ is the mean of the remaining quarter-energy left in the nodes of the path. The reason for choosing the remaining quarter of energy is that for those WSN nodes with only a quarter of their energy left, their sensor readings should be collected first, before their batteries are depleted. v_1 and v_2 are chosen as in (12). Figure 4 shows a brief flowchart of the RNN algorithm with HHO.

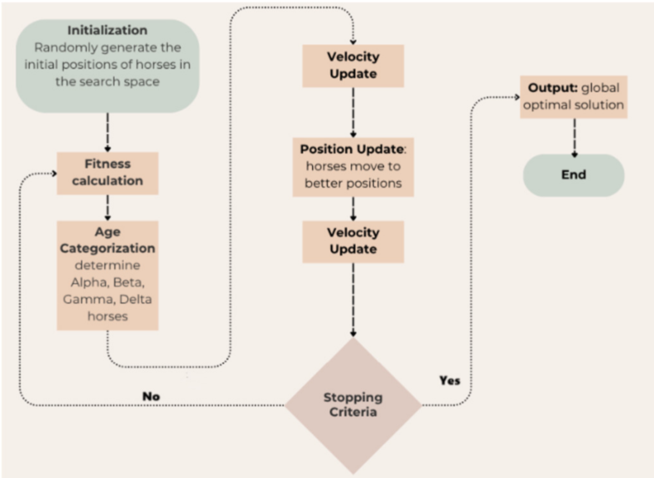

Fig. 4. Flowchart of HHO.

III. RESULTS

This study considers a network of static sensors, where the sensor data should be gathered every n seconds. Each sensor is assumed to have its energy source and knows its geographical location and the remaining energy level. The data-gathering algorithm is carried on by a central entity (e.g. a high-profile sensor with continuous energy and high calculation power). Both the RNN and HHO algorithms were simulated using MATLAB 2022b. The simulation scenarios generated a random node location in each iteration. In addition, the number of nodes changed from 50 to 300 nodes, with a step of 50 nodes each time. The criteria for choosing the path for each algorithm were first considered as only Euclidean distance, and then a combined distance-energy weights (the proposed method scenario). Figures 5 and 6 show the results for TSP using RNN and HHO algorithms.

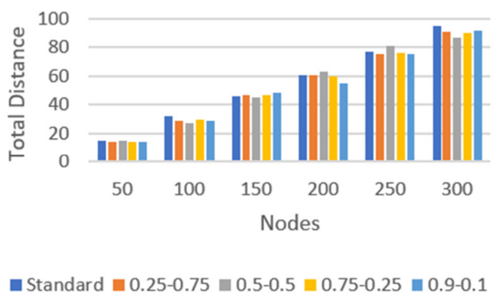

Fig. 5. Total distance using the standard and the modified RNN using distance–energy weights.

In Figure 5, dark blue bars represent the standard route search length, based on TSP, compared to various distance-energy weight pairs (in other colors) using the RNN algorithm.

It is clear that in most cases, the hybrid path selection (based on a weighted sum of both distance and remaining energy) gives a shorter path, compared to the standard TSP-RNN algorithm (where only distance is incorporated in calculations). This can be explained due to the ability of the distance-energy pair to find new solutions (new routes) that are shorter than the distance-only version.

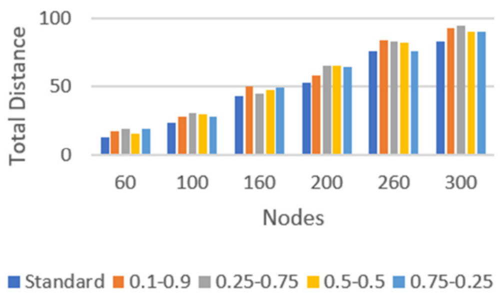

Fig. 6. Total distance using standard and modified HHO using distance-energy weights.

In Figure 6, the standard TSP based on HHO outperforms the weighted sum of the distance-energy approach. This is because the number of generated solutions using HHO will ultimately merge into a solution that is similar to the distance-only version of the TSP horse herd. However, this does not mean that the proposed weighted sum version of the horse herd performs less than the standard one. The proposed algorithm considers the remaining energy as well as the distance, thus preventing the overall WSN network from failing.

Table II shows the total travel cost of the TSP route, using both the standard and modified RNN algorithms. The RNN does not only predict a solution but also estimates confidence in this solution. Therefore, RNN has a heuristic nature. As a result, the weighted sum solution using the modified RNN would achieve the best solution (shortest path), compared to the standard RNN solution (where no remaining energy is considered).

TABLE II. TOTAL ROUTE COST USING STANDARD AND MODIFIED TSP-RNN

Nodes	**Standard distance**	**Proposed weighted (distance-energy) route total cost**				
		0.1-0.9	**0.25-0.75**	**0.5-0.5**	**0.75-0.25**	**0.9-0.1**
50	14.98	14.81	14.08	14.81	14.46	**14.17**
100	31.81	30.88	28.88	27.38	29.26	**29.18**
150	45.98	46.87	46.77	**45.28**	46.99	48.12
200	60.38	60.81	60.94	63.44	60.25	**54.88**
250	77.28	76.80	75.29	81.02	76.46	**75.67**
300	95.07	92.42	91.12	**87.18**	90.38	92.12

Table III compares the total distance costs in two cases, distance metric versus weighted distance-energy metric, using standard and weighted HHO. The results show that the total cost decreases as the energy weight increases. However, the total cost in all weighted cases (using weighted distance-energy) is higher than in the case where only distance is considered. This trade-off is because the TSP using the modified horse herd (with weighted sum) searches for a path that is not necessarily the shortest, but rather a one that gives a minimum cost function in terms of the weighted sum of distance and energy. In addition, the initialization of the horse herd assumes random solutions at the beginning and keeps a record of the best solution (alpha horse) at each iteration. Therefore, the solutions found (while combining both distance and energy) are not necessarily the minimum distance, but they should differ from the shortest path solution (minimum distance cost) because the remaining energy distribution is not correlated with the minimum distance.

TABLE III. TOTAL ROUTE COST, WITH STANDARD AND MODIFIED TSP-HORSE HERD

Nodes	**Standard distance**	**Proposed weighted (distance-energy) route total cost**				
		0.1-0.9	**0.25-0.75**	**0.5-0.5**	**0.75-0.25**	**0.9-0.1**
60	13.43	17.22	18.80	**15.36**	19.29	17.24
100	24.05	**27.78**	30.54	29.58	28.39	30.78
160	43.28	50.54	**44.90**	47.79	49.09	48.75
200	53.35	**58.12**	65.39	65.39	64.91	61.71
260	75.93	84.35	82.88	82.21	**76.45**	83.95
300	82.75	92.82	94.71	90.52	**89.91**	99.41

Table IV compares the best scores of the proposed TSP-RNN hybrid algorithm against the standard Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). It can be inferred that the proposed TSP-RNN weighted sum algorithm outperforms both the standard GA and PSO. Both GA and PSO search for the shortest path, regardless of the remaining energy. In addition, the total iterations of both GA and PSO were found to be larger than the proposed algorithm (approximately 240 more iterations) because of their random nature.

TABLE IV. COMPARISON BETWEEN TSP-RNN, MODIFIED TSP-RNN, GA, AND PSO TOTAL COSTS

Nodes	**Standard RNN**	**Best proposed total cost**	**Genetic**	**PSO**
50	14.98	**14.17**	15.61	14.23
100	31.81	**29.18**	32.02	30.25
150	45.98	**45.28**	46.32	45.35
200	60.38	**54.88**	62.31	58.87
250	77.28	**75.67**	78.14	77.04
300	95.07	**87.18**	97.34	89.84

Table V compares the performance of the proposed TSP-HHO against GA and PSO. The standard HHO offered the least cost. The proposed hybrid TSP performs better than both GA and PSO when the number of nodes exceeds 100.

TABLE V. COMPARISON BETWEEN TSP-HHO, MODIFIED TSP-HHO, GA, AND PSO TOTAL COSTS

Nodes	**Standard HHO**	**Best proposed total cost**	**GA**	**PSO**
60	**13.43**	15.36	14.86	15.25
100	**24.05**	27.78	26.58	26.48
160	**43.28**	44.90	46.27	45.73
200	**53.35**	58.12	60.93	59.64
260	**75.93**	76.45	78.31	77.84
300	**82.75**	89.91	92.04	91.25

IV. CONCLUSIONS

This work proposed a modification to the calculation of the TSP best route in WSN using a weighted sum of both the distance and the remaining energy. The results were compared with the standard methods. Using the RNN with a weighted sum improved the network performance and shortened the optimum path for the TSP, compared to the case considering only distance. On the other hand, the weighted sum did not improve the TSP performance in the case of HHO, but the remaining energy was considered and the best TSP routes were slightly longer than the standard TSP case. The proposed modified algorithms show promising results and could be further investigated to optimize data gathering in WSNs.

Future enhancements could be further investigated, such as hybrid metaheuristic algorithms with TSP. In addition, a dynamic clustering scenario for WSN could be considered with adaptive TSP path planning. Furthermore, energy-aware TSP trajectory optimization could be investigated with mobile sinks.

REFERENCES

[1] P. Kumar and S. R. N. Reddy, "Wireless sensor networks: a review of motes, wireless technologies, routing algorithms and static deployment strategies for agriculture applications," *CSI Transactions on ICT*, vol. 8, no. 3, pp. 331–345, Sep. 2020, https://doi.org/10.1007/s40012-020-00289-1.

[2] M. R. Rubia and M. SivanArulSelvan, "A Survey on Mobile Data Gathering in Wireless Sensor Networks - Bounded Relay," *International Journal of Engineering Trends and Technology*, vol. 7, no. 5, pp. 205–208, Jan. 2014, https://doi.org/10.14445/22315381/IJETT-V7P247.

[3] O. Gurewitz, M. Shifrin, and E. Dvir, "Data Gathering Techniques in WSN: A Cross-Layer View," *Sensors*, vol. 22, no. 7, Mar. 2022, Art. no. 2650, https://doi.org/10.3390/s22072650.

[4] E. J. Duarte-Melo and M. Liu, "Data-gathering wireless sensor networks: organization and capacity," *Computer Networks*, vol. 43, no. 4, pp. 519–537, Nov. 2003, https://doi.org/10.1016/S1389-1286(03)00357-8.

[5] R. Sridhar and N. Guruprasad, "Energy efficient chaotic whale optimization technique for data gathering in wireless sensor network," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 4, Aug. 2020, Art. no. 4176, https://doi.org/10.11591/ijece.v10i4.pp4176-4188.

[6] T. Tawanda, P. Nyamugure, S. Kumar, and E. Munapo, "A Labelling Method for the Travelling Salesman Problem," *Applied Sciences*, vol. 13, no. 11, May 2023, Art. no. 6417, https://doi.org/10.3390/app13116417.

[7] B. Gavish and S. C. Graves, "The Travelling Salesman Problem and Related Problems," Massachusetts Institute of Technology, Operations Research Center, Working Paper, Jul. 1978.

[8] H. A. Abdulkarim and I. F. Alshammari, "Comparison of algorithms for solving traveling salesman problem," *International Journal of Engineering and Advanced Technology*, vol. 4, no. 6, pp. 76–79, 2015.

[9] S. Goyal, "A survey on travelling salesman problem," presented at the Midwest Instruction and Computing Symposium, 2010.

[10] S. Soni and M. Shrivastava, "Novel Learning Algorithms for Efficient Mobile Sink Data Collection Using Reinforcement Learning in Wireless Sensor Network," *Wireless Communications and Mobile Computing*, vol. 2018, no. 1, Jan. 2018, Art. no. 7560167, https://doi.org/10.1155/2018/7560167.

[11] G. Chen, "A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation." arXiv, Jan. 14, 2018, https://doi.org/10.48550/arXiv.1610.02583.

[12] M. M. Krishna, Midhunchakkaravarthy, and J. Vankara, "Detection of Sarcasm Using Bi-Directional RNN Based Deep Learning Model in Sentiment Analysis," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 31, no. 2, pp. 352–362, Jul. 2023, https://doi.org/10.37934/araset.31.2.352362.

[13] J. Zhu, Q. Jiang, Y. Shen, C. Qian, F. Xu, and Q. Zhu, "Application of recurrent neural network to mechanical fault diagnosis: a review," *Journal of Mechanical Science and Technology*, vol. 36, no. 2, pp. 527–542, Feb. 2022, https://doi.org/10.1007/s12206-022-0102-1.

[14] F. Wang and D. M. J. Tax, "Survey on the attention based RNN model and its applications in computer vision." arXiv, Jan. 25, 2016, https://doi.org/10.48550/arXiv.1601.06823.

[15] L. Medsker and L. C. Jain, *Recurrent Neural Networks: Design and Applications*. CRC Press, 1999.

[16] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities.," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982, https://doi.org/10.1073/pnas.79.8.2554.

[17] F. M. Salem, "A Basic Recurrent Neural Network Model." arXiv, Dec. 29, 2016, https://doi.org/10.48550/arXiv.1612.09022.

[18] R. M. Schmidt, "Recurrent Neural Networks (RNNs): A gentle Introduction and Overview." arXiv, Nov. 23, 2019, https://arxiv.org/abs/1912.05911v1.

[19] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997, https://doi.org/10.1109/78.650093.

[20] I. Sutskever, "Training Recurrent Neural Networks," Ph.D. dissertation, University of Toronto, Canada, 2013.

[21] F. MiarNaeimi, G. Azizyan, and M. Rashki, "Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems," *Knowledge-Based Systems*, vol. 213, Feb. 2021, Art. no. 106711, https://doi.org/10.1016/j.knosys.2020.106711.

[22] D. Moldovan, "Binary Horse Optimization Algorithm for Feature Selection," *Algorithms*, vol. 15, no. 5, May 2022, Art. no. 156, https://doi.org/10.3390/a15050156.

[23] H. Abdulkarim, M. K. Farhan, and M. Ghanim, "Image Forgery Detection Using Combined 2D DWT-Neural Network," in *2023 16th International Conference on Developments in eSystems Engineering (DeSE)*, Istanbul, Turkiye, Dec. 2023, pp. 276–281, https://doi.org/10.1109/DeSE60595.2023.10468748.

[24] S. Panbude, B. Iyer, A. B. Nandgaonkar, and P. S. Deshpande, "DFPC: Dynamic Fuzzy-based Primary User Aware clustering for Cognitive Radio Wireless Sensor Networks," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12058–12067, Dec. 2023, https://doi.org/10.48084/etasr.6279.

[25] H. Khujamatov, M. Pitchai, A. Shamsiev, A. Mukhamadiyev, and J. Cho, "Clustered Routing Using Chaotic Genetic Algorithm with Grey Wolf Optimization to Enhance Energy Efficiency in Sensor Networks," *Sensors*, vol. 24, no. 13, Jan. 2024, Art. no. 4406, https://doi.org/10.3390/s24134406.