International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Instrumentation Mesure Métrologie
Vol. 23, No. 4, August, 2024, pp. 307-316
Journal homepage: http://iieta.org/journals/i2m

Optimizing Energy Efficiency in Wireless Sensor Networks Using Dijkstra's Algorithm

Lutfi AbdulKadhim Mohammed*, Ahmed Mudheher Hasan, Ekhlas Kadhum Hamza

Control and Systems Engineering Department, University of Technology, Baghdad 100600, Iraq

Corresponding Author Email: cse.22.16@grad.uotechnology.edu.iq

https://doi.org/10.18280/i2m.230406

Received: 2 July 2024
Revised: 8 August 2024
Accepted: 15 August 2024
Available online: 23 August 2024

Keywords:
Wireless Sensor Networks (WSNs), energy efficiency, Dijkstra's algorithm, routing protocols, optimization

ABSTRACT

A Wireless Sensor Network (WSN) is a network of special systems containing separate sensors working together to collect data about a particular phenomenon without the need for human intervention. Because of the limited battery life of sensor nodes, energy efficiency becomes a significant issue in Wireless Sensor Networks (WSNs). This paper employs Dijkstra's algorithm to optimize energy-efficient routing in WSNs. Traditional shortest path finding algorithm in networks, Dijkstra's Algorithm is adjusted to minimize energy consumption by selecting routes that balance the energy load among nodes. This work employs the algorithm to consider dynamic aspects and energy metrics associated with WSNs. Using simulations, this proposed algorithm is compared against the Ant Colony Optimization algorithm (ACO) in terms of energy consumption, run time, network lifetime, node death rate, and data transmission success rate. The results show that Dijkstra's algorithm reduced overall power usage and extended network lifetime. This research emphasizes how graph-based algorithms can improve on-energy usage in WSNs thus providing a promising approach towards sustainable and long-lived sensor network deployments. Further optimization techniques and practical implementation scenarios will be explored in future work.

1. INTRODUCTION

Wireless Sensor Networks (WSNs) are an essential component of modern technology since they are used in many fields, including smart cities, healthcare, military surveillance, and environmental monitoring. WSNs consist of many tiny sensor nodes put in place around the area being monitored to collect data from there. A major problem with WSNs involves power efficiency because sensor nodes typically operate on battery sources with limited lifetimes. For network longevity and continuous reliable operation, effective energy management is vital [1-4].

The efficiency of energy in WSNs is mainly determined by the routing protocols that are employed for data transmission between nodes and the base station [5-7]. LEACH (Low Energy Adaptive Clustering Hierarchy), which is one of the traditional routing protocols, utilizes clustering techniques to distribute energy demands among nodes [8]. Nevertheless, although they may be helpful at some point, these protocols usually fall short in minimizing overall network power consumption especially when it comes to wide-scale deployment with different node densities.

Regarding the WSN energy efficiency issue, Dijkstra's algorithm presents a potential solution. It is a famous shortest-path algorithm for graph-based networks primarily designed for routing purposes in communication networks and can be applied to finding energy-efficient paths that share energy consumption equally among sensor nodes [9]. By introducing energy metrics into the cost function of the algorithm, it becomes possible to identify powerful saving routes and thus improve network lifetime.

In previous studies, different strategies for energy-efficient routing in WSNs are explored such as nature-inspired algorithms like the Intelligent Water Drops (IWD) algorithm and optimization techniques, for example, Ant Colony Optimization (ACO) [10, 11]. Algorithmic adaptations offer great possibilities for improving energy efficiency as illustrated by these works. Moreover, recently hybrid routing protocols that integrate proactive and reactive routing strategies have emerged as well as the use of mobile sinks to dynamically adjust the data collection points and balance energy consumption [12, 13].

While optimization methods like ACO present heuristic solutions to complex problems, the Dijkstra algorithm provides a deterministic and certain optimal solution for shortest-path routing problems. Dijkstra's algorithm always finds the shortest path without the randomness inherent in heuristic methods like ACO ensuring predictable and reliable outcomes. It also has lower computational overheads than other algorithms; thus, faster decision-making regarding route selection and reduced processing time, both of which are necessary for time-sensitive applications.

This study investigates the energy efficiency of a modified Dijkstra algorithm in WSNs. It suggests modifying the conventional method to take energy measures into account rather than distance metrics. The results demonstrate that the Dijkstra algorithm performs better than ACO in terms of computing time and energy usage. When it comes to scalability, the Dijkstra algorithm gets better as the network gets bigger, as the number of nodes increases from 50 to 200.

In this paper, Section 2 is dedicated to the related work on energy-efficient routing in WSNs. The Dijkstra algorithm overview is in Section 3. The proposed Dijkstra algorithm method for energy consumption is discussed in Section 4. Section 5 contains the results of the simulations. Lastly, Section 6 concludes by drawing implications from the findings and possible areas for further research.

2. RELATED WORKS

Moussa and El Belrhiti El Alaoui [14] present an energy-efficient cluster-based routing protocol for incorporating unequal clustering and improved ACO techniques. With such implementation, the protocol balances energy consumption among nodes especially those closer to the base station that handle larger data traffic. Also, by enhancing the ACO techniques, it optimizes routing paths, thereby reducing the overall energy consumption. Their method splits the network into dissimilar clusters considering the distance to the sink node, remaining energy, number of neighbors, and certain backward relaying nodes in the preceding round to effectively balance the load across Cluster Heads (CHs). Additionally, they propose a batch-based clustering approach that eliminates the need for configuration control overhead and enables the network to operate for multiple rounds. An improved ACO-based routing technique is used for effective and consistent routing from CHs to the sink node. The modification implies that the heuristic function is formulated considering the energy of the next node, the distance from the current sensor node to the next sensor node, and the number of preferred probable relay nodes.

Subramani et al. [15] propose a reliable and energy-efficient routing method utilizing cluster head selection, clustering, and a reliable routing algorithm with failure identification. The nodes are grouped in clusters, and then the ACO is applied for an intelligent routing process for effective data delivery. The cluster heads are chosen according to limited mobility, remaining energy, and distance from other member nodes. Since the cluster heads are designed to complete the routing tasks, they are rotated regularly. After the nodes are grouped into clusters, an intelligent routing procedure using ACO is suggested for efficient data transfer. The main findings are a decrease in energy consumption, data delivery services, and communication reliability.

Moussa et al. [16] introduce an ACO-based routing protocol specifically developed for WSN-based forest fire detection. The proposed protocol assures communication reliability in addition to high-quality communication routes for energy efficiency. Reliable transmission is necessary for critical events in delay-intolerant applications (like forest fire detection) so that choices can be made and actions may be taken promptly. According to the simulation results, their approach performs significantly better in terms of network lifetime and response time (30.55%) than both Load Balanced Cluster-based Routing using ACO and Enhanced Ant-based QoS-aware routing protocol for Heterogeneous WSNs.

Adel et al. [17] propose a Dijkstra-based routing algorithm and a Trust-Based clustering. The proposed method starts by grouping the sensor nodes into clusters to build an organized network architecture. The members of each cluster go through a three-phase process to evaluate their level of trust: overall, indirect, and direct. By choosing the most dependable node to serve as the cluster head, these trust metrics allow for dependable data aggregation and routing. The trust metrics ensure reliable data clustering and efficient routing using the Dijkstra algorithm. The well-known LEACH algorithm and their algorithm are compared. Critical performance parameters including energy consumption and packet rate delivery throughout the network's lifetime are the main focus of the evaluation. The outcomes of the simulations show how effective their method is.

Wu et al. [18] suggest a method based on the K-means algorithm for clustering, the ACO to compute the optimal route, and the whale algorithm and a backpropagation neural network to extend the coverage area of a WSN. The experimental results show that the WSN's coverage area has grown, its lifetime has been prolonged, and its nodes' overall energy consumption has been efficiently decreased.

Kaur and Mahajan [19] propose Energy-efficient clustering and tree-based routing protocol based on hybrid ant colony optimization and particle swarm optimization for energy efficiency in WSNs. The proposed protocol divides the WSN into several clusters, and cluster heads are selected for each cluster. The proposed protocol selects the shortest path between the sink node and each cluster head. A MATLAB simulation consists of 100 sensor nodes distributed randomly in a 100*100 area with one base station is used. The outcomes show that the proposed hybrid protocol expands the WSN lifespan.

However, an important issue in deploying WSNs is extending the network's lifetime through efficient energy use. Several optimization techniques have been investigated in current literature, some combine routing protocols, clustering techniques, and different optimization methods like ACO, PSO, and hybrid algorithms. Frequently these approaches are associated with high computational costs. In such scenarios where resources are limited like those found in WSNs, this complexity may cause longer processing time and higher energy consumption. Deploying complicated optimization tools on sensor nodes with limited resources is challenging because it might require more processing power and memory.

3. THE DIJKSTRA ALGORITHM OVERVIEW

Applying Dijkstra's method [9] to every vertex in the graph resolves the single-source shortest path (SSSP) problem in graph theory. It is used for directed graphs with non-negative weights. In the Dijkstra algorithm, two types of vertices are distinguished: (1) Solved and (2) Unsolved vertices. This starts by marking the source vertex as solved then follows by examining all additional edges (by unresolved vertices) linked to the source vertex which helps determine the shortest path to the destination. After the algorithm identifies an edge as the shortest among others, it adds the fitting vertex to the list of solved vertices. The algorithm continues until all nodes are visited one by one. The benefit of this algorithm is that it does not need to consider every edge. This is very helpful especially if some edges are heavily weighted [20].

The Dijkstra algorithm's time and space complexity are $O((E+V)\log V)$ and $O(V+E)$, respectively, where V and are the number of vertices and edges. It is implemented using a priority queue and an adjacency list. As a result, it works well for finding the shortest paths in networks with a large number of edges and nodes. The ACO algorithm has an $O(k \cdot n \cdot V)$ time complexity and an $O(V^2)$ space complexity, where k denotes the number of ants and n is the number of iterations [21, 22].

In many areas including Geographic Information Systems (GIS), Dijkstra's algorithm is used as an effective tool in solving various problems such as locating the shortest path between two given points in road networks [23]. In Routing Protocols of computer networks, Dijkstra's algorithm is fundamental in the development of routing protocols such as Open Shortest Path First (OSPF) used in Internet Protocol (IP) networks [24]. Concerning the robot navigation field in robotics, Dijkstra's algorithm has a unique role in planning robotic paths through environments with obstacles [25].

4. THE PROPOSED METHODOLOGY

This section outlines the steps involved and demonstrates how Dijkstra's algorithm is changed to choose a path that reduces energy consumption while still satisfying WSNs constraints. When using Dijkstra's algorithm on WSNs however, some changes have to be made to factor in power usage. This means taking into account the cost of energy instead of distance costs and as well tracking the level of energy in sensor nodes to avoid nodes depleting.

4.1 Energy consumption model

The cost function for the model is defined as the minimum energy cost of selected routes to enhance the energy utilization of sensor nodes and enhance the lifespan of the entire WSN. In calculating total energy consumption, consideration is made of the energy usage per sensor node. Eqs. (1)-(3) can be used to determine the amount of energy used by a single node during transmission [1].

$$E_{tx} = \left(E_{elec} + E_{amp} \times d^2\right) \times L \quad (1)$$

where, E_{elec} is the energy (measured in Joules/bit) that the transmitter or receiver circuitry uses. E_{amp} (measured in Joules/bit/m^2) is the amount of energy needed to transmit a bit over the air, L is the number of bits sent, and d is the distance between nodes. The energy consumed by a node during reception can be calculated using Eq. (2).

$$E_{rx} = E_{elec} \times L \quad (2)$$

So, the energy consumed by a single node is:

$$E_{node} = E_{tx} + E_{rx} \quad (3)$$

The cost function of the Dijkstra is defined by the total energy consumption (E_{total}) of the selected path and given by Eq. (4), where n is the number of nodes in the path and i is the i^{th} node.

$$E_{total} = \sum_{i=1}^{n} E_{node}(i) \quad (4)$$

4.2 Assumptions and parameters

- Each node has equal energy capacity and initial energy level.
- The experiments are conducted using two area settings (500×500 m^2) and (1000×1000 m^2).
- The nodes are distributed randomly.
- The Sink node is located in the center of the area.
- The value of E_{elec} is often between 50 to 150 nJ/bit. A common value used in many studies is 50 nJ/bit=50×10^{-9} J/bit. While the value of E_{amp} is 100 pJ/bit/m^2 = 100×10^{-12} J/bit/m^2.

4.3 The proposed algorithm steps

1. Initialization: set random positions and initial energy levels for the nodes, and also create a network graph.
2. Path Selection: apply the modified Dijkstra algorithm to select the best path for each data transfer request based on energy consumption.
3. Energy Update: after each transfer, update nodes' energy levels.
4. Recalculation: recalculate paths frequently in case important nodes become depleted energy-wise.

5. EXPERIMENTAL RESULTS

This section uses the ACO and enhanced Dijkstra algorithms to show the experimental results of the four topologies (50, 100, 150, and 200). The area size is 500×500 m^2 for the 50, 100, 150, and 200 topologies and 1000×1000 m^2 for 200 sensors. The energy levels are randomly initialized from 0 to the initial battery capacity of 3×2.85Ah, the distance range is 100 meters, and the packet size is 128 bytes. The sensor nodes are distributed randomly, the sink node is placed in the center of the area. Python is used to implement and evaluate the recommended approach efficiently. Energy usage and execution time are used to assess the outcomes. The following laptop characteristics are used to accomplish the suggested method:

- Processor: Intel (R) Core (TM) i7-11800H, 11th generation, 2.30 GHz; RAM: 16.0 GB, of which 15.7 GB are useable.
- Windows 11 Pro 64-bit version.

5.1 The results of the Dijkstra and ACO algorithms

The suggested Dijkstra algorithm outperforms ACO in terms of time complexity and energy cost. The parameters of the ACO are shown in Table 1.

Table 1. ACO parameters for the four topologies

Parameters	Topology			
	50	100	150	200
No. of Ants	15	20	20	20
No. of Iterations	30	40	50	50

The optimal paths from source node 16 to the sink node in the 50-sensor topology selected by the Dijkstra algorithm and the ACO are shown in Figures 1 and 2, respectively.

Figures 3 and 4 show the optimal path from sensor 52 to the sink node selected by Dijkstra and the ACO algorithms using the 100-sensor topology. In Figures 5 and 6 the optimal routes from sensor 137 to the sink node selected by the Dijkstra and ACO algorithms using the 150-sensor topology are shown. Figures 7 and 8 show the optimal route from sensor 35 to the sink node in a 200-sensor topology with an area of about 1000×1000 m^2 to increase the challenges of selecting the optimal route to ensure the efficiency of the proposed method.

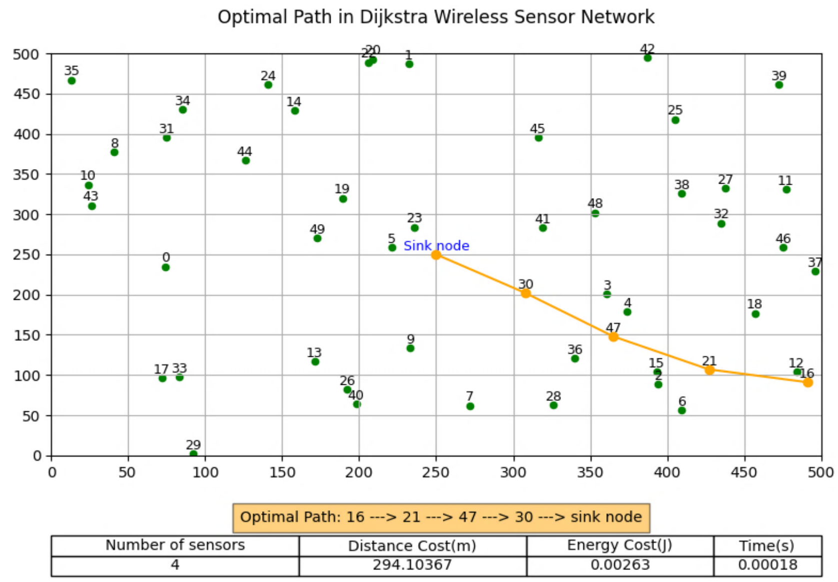


Figure 1. The optimal route from sensor 16 to the sink node in the 50-sensor topology is selected by the Dijkstra algorithm

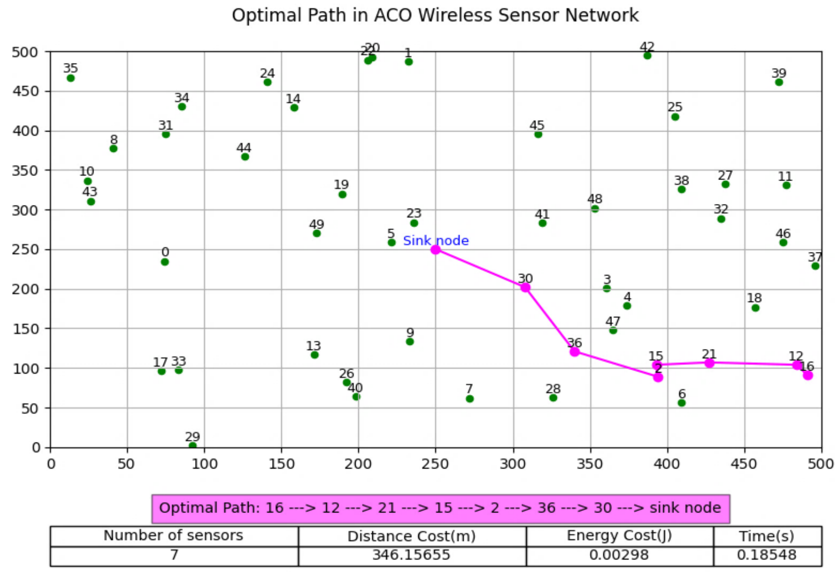


Figure 2. The optimal route from sensor 16 to the sink node in the 50-sensor topology is selected by the ACO algorithm

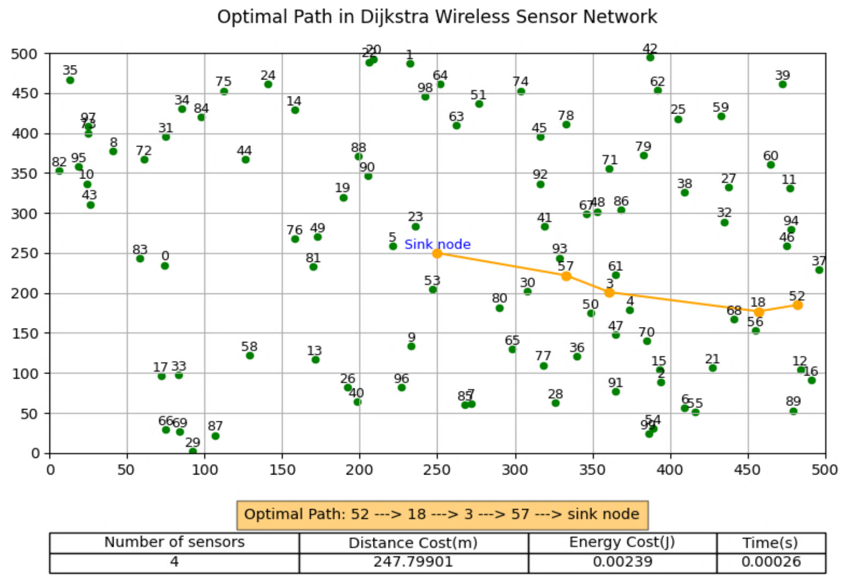
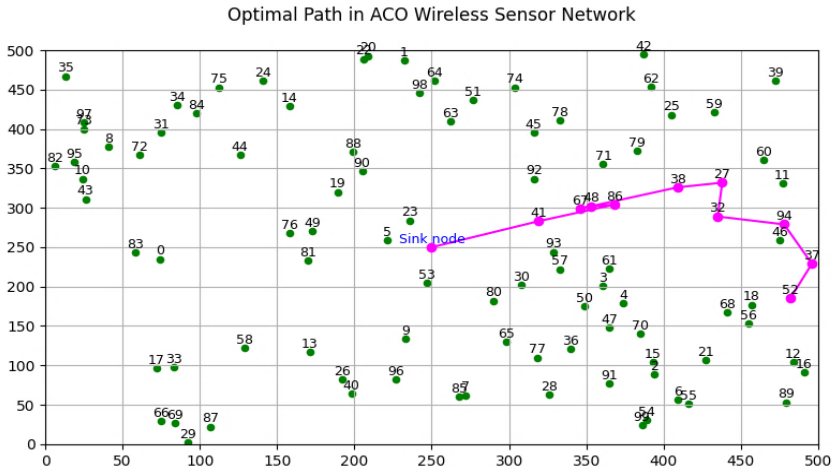


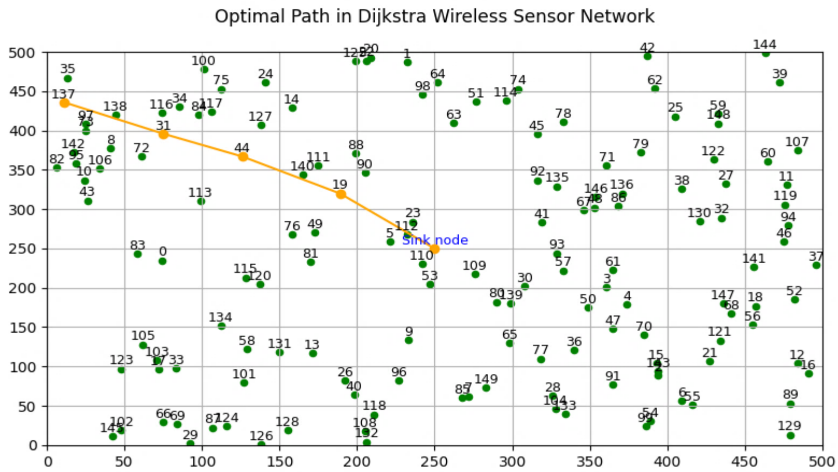
Figure 3. The optimal route from sensor 52 to the sink node in the 100-sensor topology is selected by the Dijkstra algorithm



Optimal Path: 52 ---> 37 ---> 94 ---> 32 ---> 27 ---> 38 ---> 67 ---> 48 ---> 86 ---> 41 ---> sink node

Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
10	437.26728	0.00341	0.36956

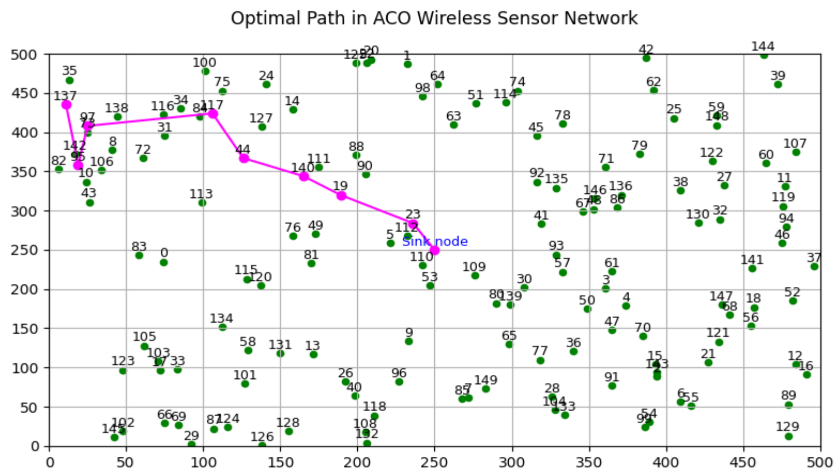
Figure 4. The optimal route from sensor 52 to the sink node in the 100-sensor topology is selected by the ACO algorithm



Optimal Path: 137 ---> 31 ---> 44 ---> 19 ---> sink node

Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
4	305.59000	0.00286	0.00040

Figure 5. The optimal route from sensor 137 to the sink node in the 150-sensor topology is selected by the Dijkstra algorithm



Optimal Path: 137 ---> 95 ---> 97 ---> 117 ---> 44 ---> 140 ---> 19 ---> 23 ---> sink node

Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
8	446.93062	0.00361	0.66404

Figure 6. The optimal route from sensor 137 to the sink node in the 150-sensor topology is selected by the ACO algorithm

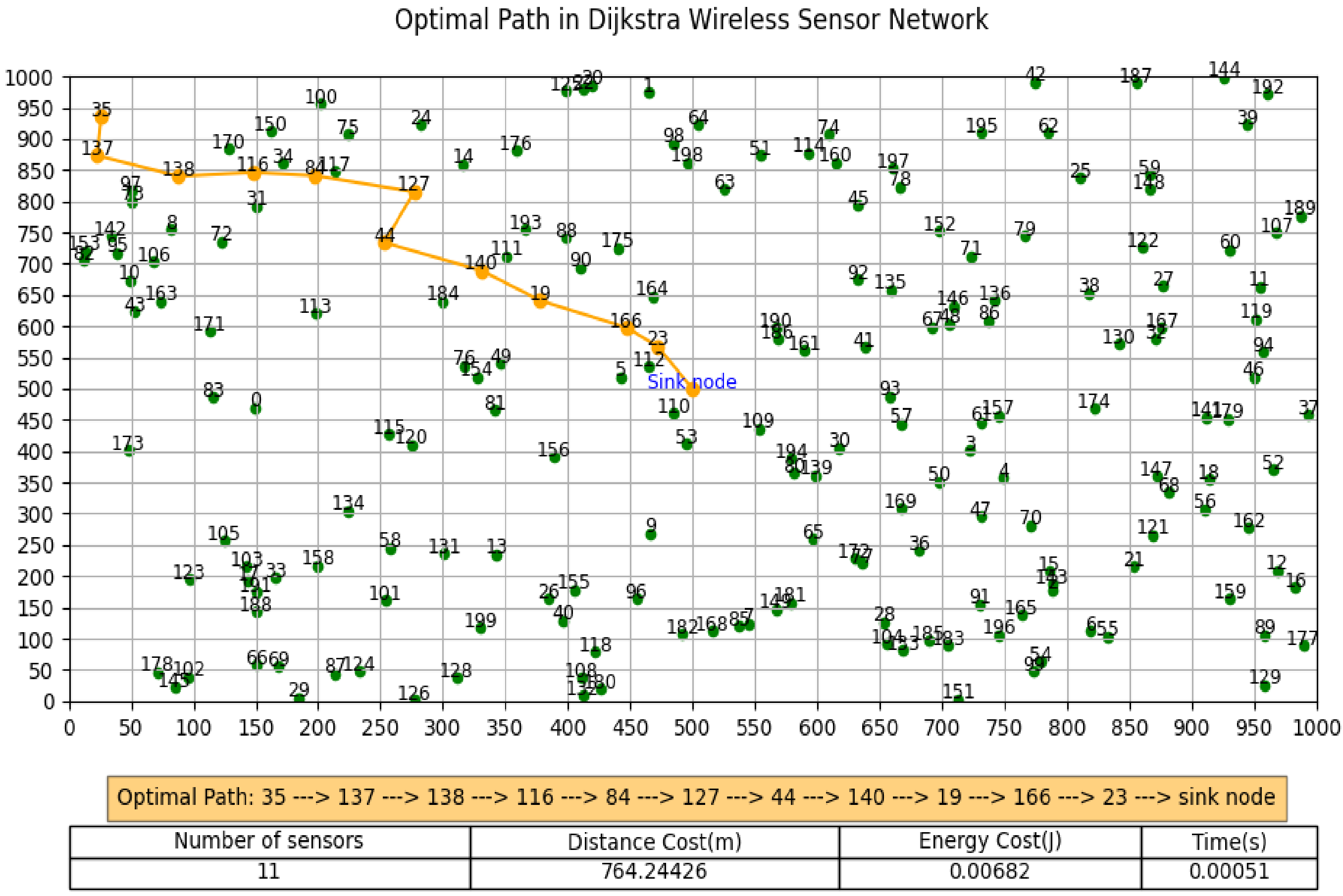


Figure 7. The optimal route from sensor 35 to the sink node in the 200-sensor topology is selected by the Dijkstra algorithm

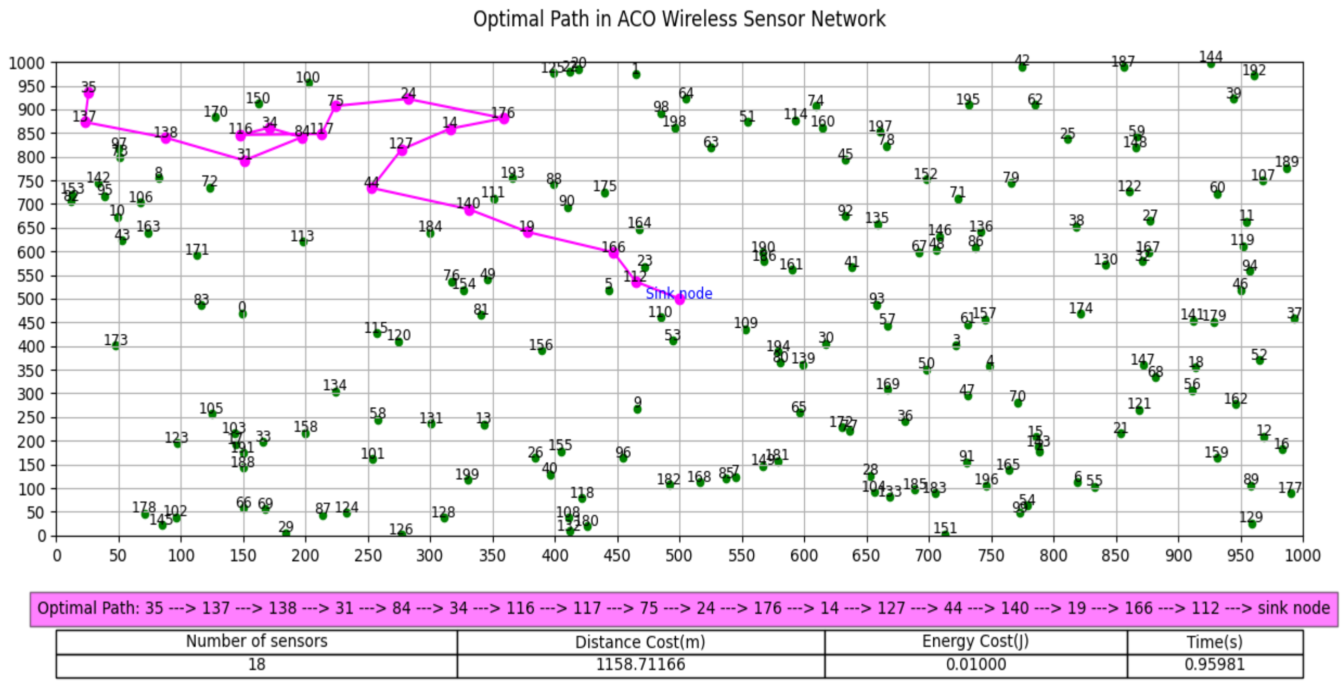


Figure 8. The optimal route from sensor 35 to the sink node in the 200-sensor topology is selected by the ACO algorithm

Table 2 shows the overall performance of the proposed Dijkstra and the ACO algorithms for the three topologies in terms of average energy consumption (joules) and the average run time in seconds. Based on the time and space complexity given in Section 3, Figure 9 compares the two methods in terms of time and space complexity. Figure 10 presents a comparison between Dijkstra's algorithm and the ACO based on three metrics: network lifetime, node death rate, and data transmission success rate for the 200-sensor topology when the network runs until all nodes are depleted. The scatters indicate that Dijkstra's algorithm is superior to ACO in terms of network lifetime, node stability, and data transmission reliability.

Table 2. The comparison between the proposed improved Dijkstra and ACO algorithms

	50		100		150		200	
	Avg. Time	**Avg. Energy**	**Avg. Time**	**Avg. Energy**	**Avg. Time**	**Avg. Energy**	**Avg. Time**	**Avg. Energy**
Dijkstra	7.00E-05	0.0017	0.00022	0.0016	0.00046	0.0015	0.0019	0.0018
ACO	0.11703	0.0023	0.26907	0.0031	0.60685	0.0036	1.15499	0.0036

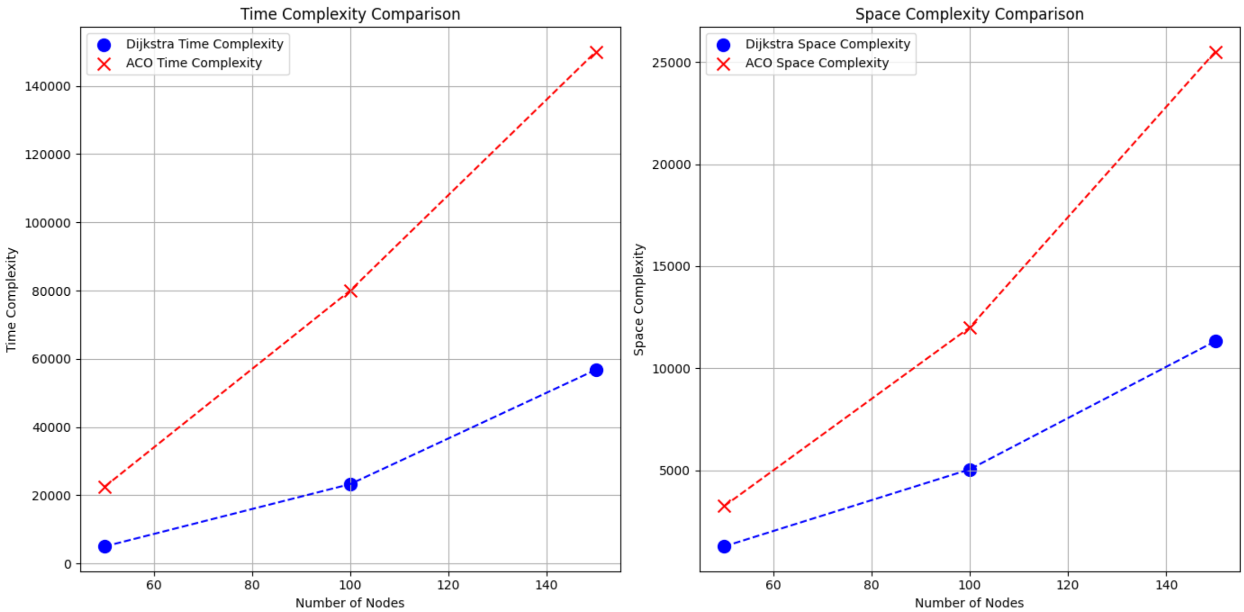


Figure 9. The time and space complexity of the Dijkstra and the ACO algorithms

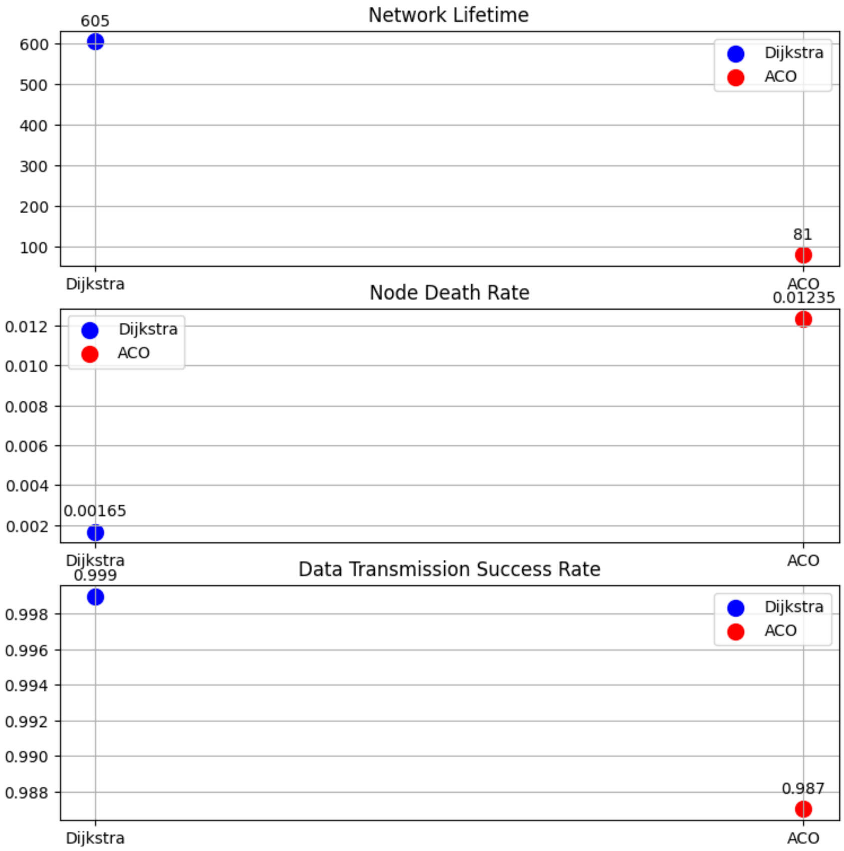


Figure 10. Performance Comparison of Dijkstra's Algorithm and ACO in a 200-sensor WSN

5.2 Results of energy-aware routing in WSN

Add a random initial energy level between 0 and the battery capacity for each node. Modify the plotting to turn nodes with depleted energy (0 energy) with red color. Ensure the proposed algorithms do not select nodes with depleted energy when forming the optimal path.

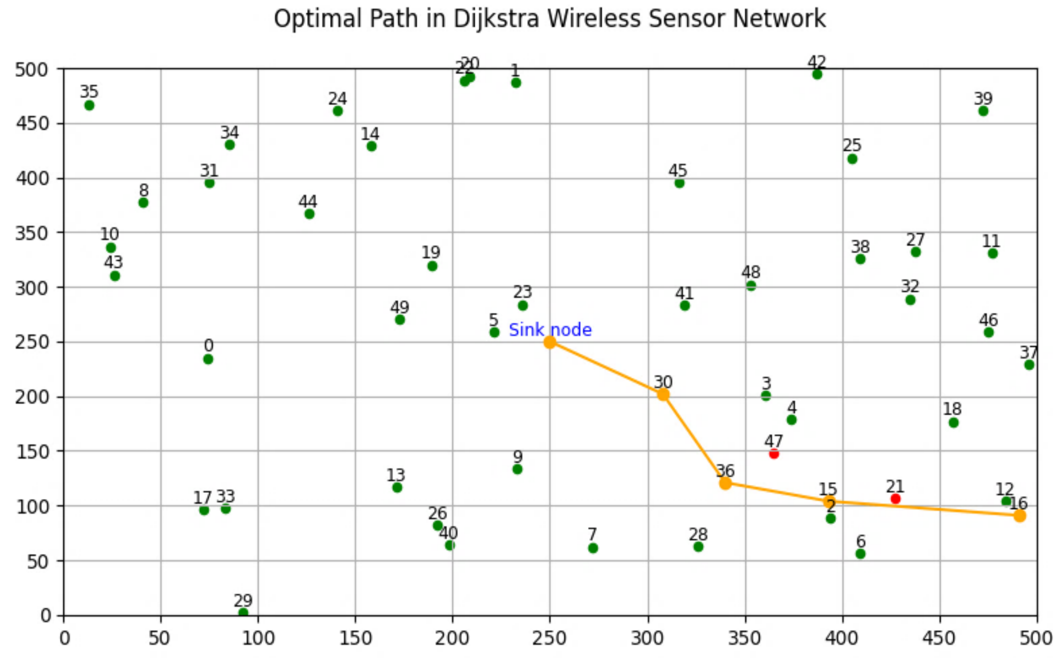
Figures 11-13 display the performance of the proposed

Dijkstra algorithm within the three topologies after the depletion of some nodes.

Figure 11 shows the optimal route from sensor 16 to the sink node after the depletion of sensor 21 and sensor 47 in the 50-sensor topology selected by the Dijkstra algorithm.

Figure 12 displays the optimal route from sensor 52 to the sink node in the 100-sensor topology after the depletion of sensor 18 and sensor 57 selected by the Dijkstra algorithm.

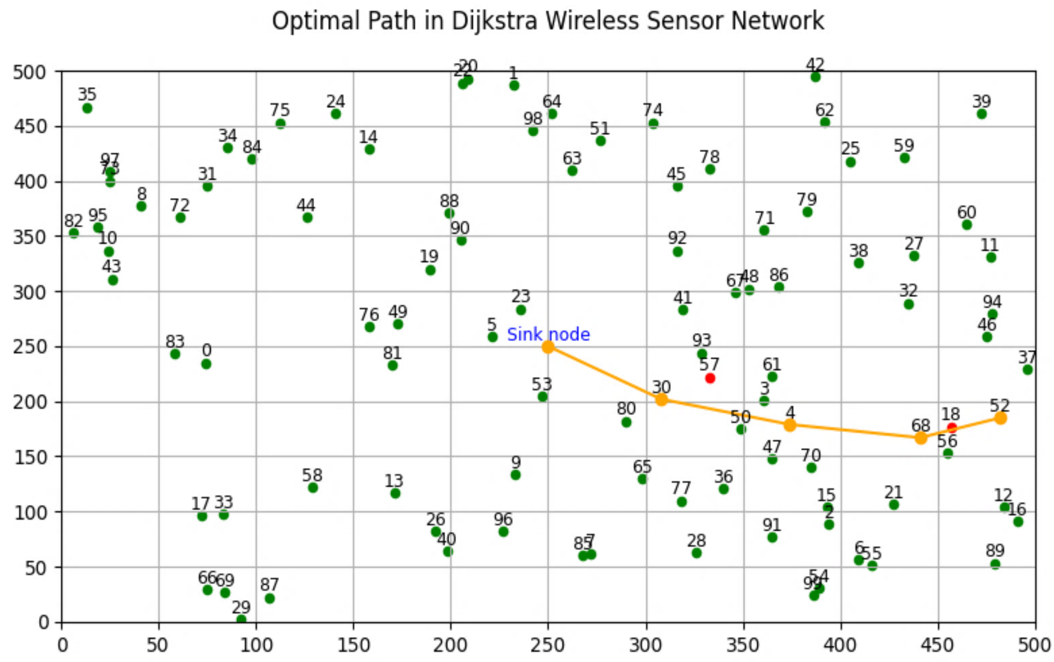
Figure 13 depicts the optimal route from sensor 137 to the sink node in the 150-sensor topology after the running down of sensor 31 and sensor 19 selected by the Dijkstra algorithm.



Optimal Path: 16 ---> 15 ---> 36 ---> 30 ---> sink node

Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
4	316.89619	0.00308	0.00011

Figure 11. The optimal route from sensor 16 to the sink node after the depletion of sensors 21 and 47



Optimal Path: 52 ---> 68 ---> 4 ---> 30 ---> sink node

Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
4	258.02227	0.00217	0.00033

Figure 12. The optimal route from sensor 52 to the sink node after the depletion of sensors 18 and 57

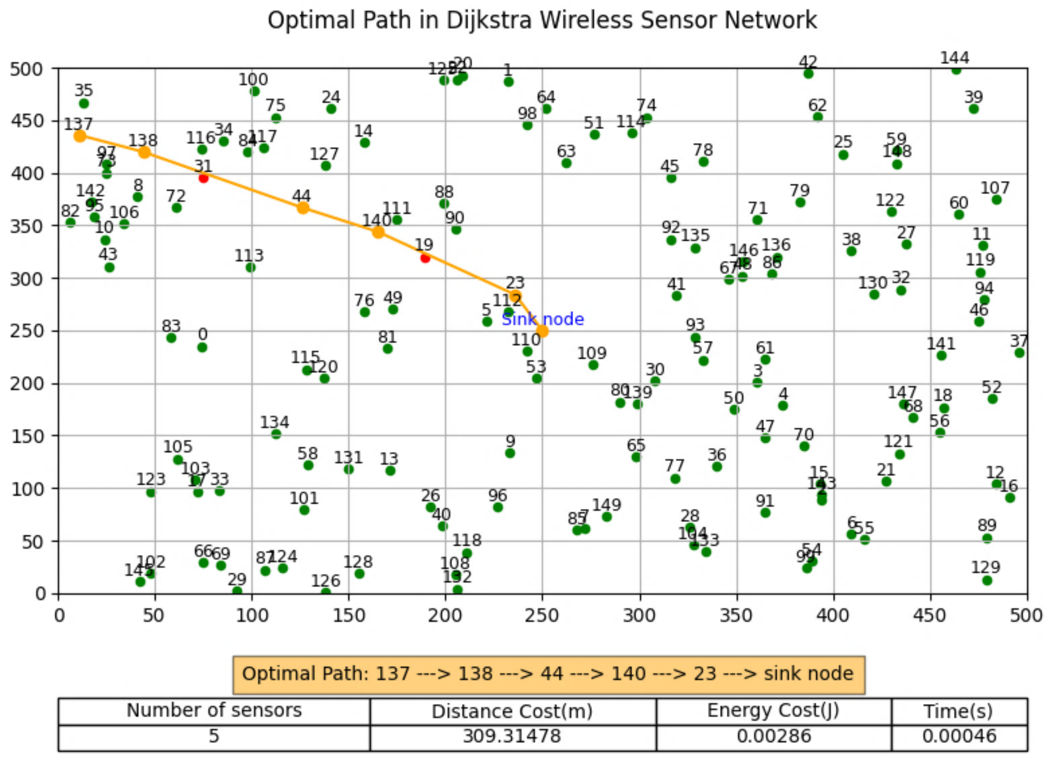


Figure 13. The optimal route from sensor 137 to the sink node after the depletion of sensors 31 and 19

5.3 Result discussion

The performance of Dijkstra's algorithm and ACO is evaluated across the four proposed network topologies: 50, 100, 150, and 200 nodes. The results are measured in terms of run time and energy consumption. As shown in experimental results, the Dijkstra algorithm always outperforms the ACO algorithm in both energy and run-time parameters, even when the number of sensors and area size is increased it still shows better performance. Dijkstra's algorithm is the recommended solution for shortest-path issues in static graphs. The time and space complexity of the two methods explains why Dijkstra’s algorithm often exhibits lower time overhead compared to ACO making it more efficient in applications that need minimum run time. Dijkstra's algorithm produced an average time of 7.00×10^{-5} seconds and an average energy usage of 0.0017 joules for the WSN with 50 nodes. On the other hand, ACO uses more energy (0.0023 joules) and takes longer on average (0.11703 seconds). Dijkstra's method remained effective up to a network size of 100 nodes, with an average execution time of 0.00022 seconds and an average energy usage of 0.0016 units. However, the ACO's performance demonstrated a significant rise in time, amounting to 0.26907 seconds, and energy consumption, up to 0.0031 units Dijkstra's method continued to perform well for a network with 150 nodes, consuming an average of 0.0015 units of energy and 0.00046 seconds on average. The average duration of ACO increased to 0.60685 seconds, and 0.0036 units of energy were consumed. Dijkstra's algorithm demonstrated an average duration of 0.0019 seconds and an average energy consumption of 0.0018 units for the largest network size of 200 nodes. With an average time of 1.15499 seconds and energy consumption of 0.0036 units, ACO's performance continued to degrade. Three important metrics are compared between Dijkstra's Algorithm and ACO: Network Lifetime, Node Death Rate, and Data Transmission Success Rate. When comparing Dijkstra's method to ACO (81) the network lifespan of 605 is much greater. When comparing Dijkstra (0.00165) to ACO (0.01235), the node death rate is much lower. High data transmission success rates are achieved by both methods; Dijkstra marginally outperforms ACO (0.999 vs. 0.987).

6. CONCLUSIONS AND FUTURE WORKS

In this research paper, a comparative study is made on the energy efficiency of the modified Dijkstra’s Algorithm and ACO in Wireless Sensor Networks. The proposed Dijkstra algorithm uses the energy consumption model as a cost function instead of the distance. Based on the evaluation and analyses, it was found that the Dijkstra algorithm results outperform ACO outcomes in terms of energy efficiency, time, network lifetime, node death rate, and data transmission success rate. Dijkstra’s algorithm has the feature of being capable of being efficient in terms of energy consumption as it predicts the shortest path directly through a deterministic approach thus reducing computational overhead and the need for other energy-intensive operations. On the other hand, although it may provide better routing solutions, the ACO algorithm needs more computational resources and repeated processes which lead to high usage of energy. In general, situations where the network topology is stable and finding the precise optimal path is vital are better suited for Dijkstra's method. In more complicated or dynamic contexts where numerous criteria need to be optimized simultaneously the ACO is preferred. As far as future work is concerned, constraints and improvements can be imposed on the ACO algorithm to improve its performance such that it selects the most optimal paths with minimal consumption of energy in WSNs. Also, additional parameters can be checked to prove the proposed method's performance efficiency.

REFERENCES

[1] Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA. https://doi.org/10.1109/HICSS.2000.926982

[2] Abdulaziz, W.B., Croock, M.S. (2022). Design of smart irrigation system for vegetable farms based on efficient wireless sensor network. Iraqi Journal of Computer, Communication, Control and System Engineering, 22(1): 72-85. https://doi.org/10.33103/uot.ijccce.22.1.8

[3] Hamza, E.K., Salman, K.D., Jaafar, S.N. (2023). Wireless sensor network for robot navigation. In Mobile Robot: Motion Control and Path Planning, Cham: Springer International Publishing, pp. 643-670. https://doi.org/10.1007/978-3-031-26564-8_18

[4] Alhasan, R.A., Hamza, E.K. (2023). A novel CNN model with dimensionality reduction for WSN intrusion detection. Revue d'Intelligence Artificielle, 37(5): 1121-1131. https://doi.org/10.18280/ria.370504

[5] Ahmad, I.A., Al-Nayar, M.M.J., Mahmood, A.M. (2022). Investigation of energy efficient clustering algorithms in WSNs: A review. Mathematical Modelling of Engineering Problems, 9(6): 1693-1703. https://doi.org/10.18280/mmep.090631

[6] Majeed, A.M., Raafat, S.M., Ramadhan, N.M. (2022). A survey of power consumption minimization in WSN using feedback control strategies. Iraqi Journal of Computers, Communications, Control and Systems Engineering, 22(4): 37-47. https://doi.org/10.33103/uot.ijccce.22.4.4

[7] Hamza, E.K., Alhayani, H.H. (2018). Energy consumption analyzing in single hop transmission and multi-hop transmission for using wireless sensor networks. Al-Khwarizmi Engineering Journal, 14(1): 156-163. https://doi.org/10.22153/kej.2018.10.009

[8] Singh, S.P., Sharma, S.C. (2015). A survey on cluster based routing protocols in wireless sensor networks. Procedia Computer Science, 45: 687-695. https://doi.org/10.1016/j.procs.2015.03.133

[9] Dijkstra, E.W. (1959). A note on two problems in connexion with graphs. Numerische Mathematik, 1: 269-271. https://doi.org/10.1007/BF01386390

[10] Deepalakshmi, D., Pushpa, B. (2023). Enhanced intelligent water drop algorithm optimized routing (EIWDR) for quality of service enhancement in Internet of Things-based wireless sensor networks (IWSN). International Journal of Computer Networks and Applications, 10(3): 342-358. https://doi.org/10.22247/ijcna/2023/221889

[11] Bhowmik, T., Banerjee, I. (2018). An improved ACO based energy efficient routing algorithm in WSNs. In 2018 15th IEEE India Council International Conference (INDICON), Coimbatore, India, pp. 1-6. https://doi.org/10.1109/INDICON45594.2018.8987118

[12] Aydin, M.A., Karabekir, B., Zaim, A.H. (2021). Energy efficient clustering-based mobile routing algorithm on WSNs. IEEE Access, 9: 89593-89601. https://doi.org/10.1109/access.2021.3090979

[13] Prusty, A.R., Sethi, S., Nayak, A.K. (2016). A hybrid multi-hop mobility assisted heterogeneous energy efficient cluster routing protocol for wireless ad hoc sensor networks. Journal of High Speed Networks, 22(4): 265-280. https://doi.org/10.3233/jhs-160548

[14] Moussa, N., El Belrhiti El Alaoui, A. (2021). An energy-efficient cluster-based routing protocol using unequal clustering and improved ACO techniques for WSNs. Peer-to-Peer Networking and Applications, 14(3): 1334-1347. https://doi.org/10.1007/s12083-020-01056-4

[15] Subramani, S., Selvi, M., Kumar, S.S., Kannan, A. (2021). Energy efficient clustering routing protocol and ACO algorithm in WSN. In Advances in Computing and Data Sciences: 5th International Conference (ICACDS 2021), Nashik, India, pp. 68-80. https://doi.org/10.1007/978-3-030-81462-5_7

[16] Moussa, N., Nurellari, E., El Belrhiti El Alaoui, A. (2023). A novel energy-efficient and reliable ACO-based routing protocol for WSN-enabled forest fires detection. Journal of Ambient Intelligence and Humanized Computing, 14(9): 11639-11655. https://doi.org/10.1007/s12652-022-03727-x

[17] Adel, S., Samir, R., Samy, I., Ismail, T. (2023). Integrated trust-clustering and Dijkstra routing algorithms for energy-efficient WSNs. In 2023 17th International Conference on Telecommunications (ConTEL), Graz, Austria, pp. 1-6. https://doi.org/10.1109/contel58387.2023.10199013

[18] Wu, L., Dawod, A.Y., Miao, F. (2024). Data transmission in wireless sensor networks based on ant colony optimization technique. Applied Sciences, 14(12): 5273. https://doi.org/10.3390/app14125273

[19] Kaur, S., Mahajan, R. (2018). Hybrid meta-heuristic optimization based energy efficient protocol for wireless sensor networks. Egyptian Informatics Journal, 19(3): 145-150. https://doi.org/10.1016/j.eij.2018.01.002

[20] Karur, K., Sharma, N., Dharmatti, C., Siegel, J.E. (2021). A survey of path planning algorithms for mobile robots. Vehicles, 3(3): 448-468. https://doi.org/10.3390/vehicles3030027

[21] Ammar, A., Bennaceur, H., Châari, I., Koubâa, A., Alajlan, M. (2016). Relaxed Dijkstra and A* with linear complexity for robot path planning problems in large-scale grid environments. Soft Computing, 20: 4149-4171. https://doi.org/10.1007/s00500-015-1750-1

[22] Dorigo, M., Gambardella, L.M. (1997). Ant colonies for the travelling salesman problem. Biosystems, 43(2): 73-81. https://doi.org/10.1016/s0303-2647(97)01708-5.

[23] Khaing, O., Wai, H., Myat, E. (2018). Using Dijkstra's algorithm for public transportation system in Yangon based on GIS. International Journal of Science and Engineering Applications, 7(11): 442-447. https://doi.org/10.7753/ijsea0711.1008

[24] Tadimety, P.R. (2015). OSPF: A network routing protocol. Berkeley: Apress, pp. 13-17. https://doi.org/10.1007/978-1-4842-1410-7

[25] Sundarraj, S., Reddy, R.V.K., Basam, M.B., Lokesh, G.H., Flammini, F., Natarajan, R. (2023). Route planning for an autonomous robotic vehicle employing a weight-controlled particle swarm-optimized Dijkstra algorithm. IEEE Access, 11: 92433-92442. https://doi.org/10.1109/access.2023.3302698