Journal of Engineering Science and Technology
Vol. 19, No. 6 (2024) 2225 - 2249

DYNAMIC PARAMETER ADJUSTMENT IN ANT COLONY OPTIMIZATION FOR ENERGY EFFICIENCY IN WIRELESS SENSOR NETWORK

EKHLAS KADHUM HAMZA*,
LUTFI ABDULKADHIM MOHAMMED, AHMED MUDHEHER HASAN

Control and Systems Engineering Department, University of Technology, Iraq
*Corresponding Author: 100374@Uotechnology.Edu.Iq

Abstract

The Internet of Things (IoT) is an innovative technology. It revolutionized the world of technology today, IoT connects numerous sensors and devices to the Internet to perform various functions Every day. These sensors have a finite battery life and must operate sustainably and efficiently in IoT networks. Due to these connections, the IoT uses a lot of energy. Research into energy-efficient and energy-saving techniques is thus essential. In this paper, an enhancement was added to the Ant Colony Optimization (ACO) by dynamically adjusting alpha (α) and beta (β)parameters during the iteration of ACO. This approach is designed to balance exploration and exploitation by modifying how much pheromone trails (alpha) and heuristic information (beta) influence the ants' decision-making process. This enhancement improves ACO performance when selecting the optimal route to save energy in Wireless Sensor Networks (WSNs). A comparison between traditional ACO and proposed ACO is conducted, also the results are compared with the outputs of the Dijkstra algorithm for energy efficiency. The complexity time and total energy consumption of the proposed method are computed and show that the performance of the proposed ACO is more efficient and stable than the traditional ACO algorithm. The enhanced ACO reduces energy consumption by 14.29% in the 20-sensor topology, 7.41% in the 50-sensor topology, and 26.92% in the 100-sensor topology. The Enhanced ACO achieves similar energy consumption results compared to the Dijkstra algorithm but takes more time.

Keywords: Ant Colony Optimization, Dijkstra algorithm, Energy consumption Internet of things, Wireless Sensor Networks.

1. Introduction

In recent years, IoT technologies have advanced technology for a wide range of uses with a wide range of applications being developed through the IoT. these technologies can overcome all the challenges in fields like processing capacity, security, and data mobility, along with the growth related to other technologies to maintain its predicted path. So, the IoT is described as a network of regular objects such as smartphones, Internet TVs, actuators, and sensors that are intelligently connected to make it possible for objects to communicate with one another and with people in new ways [1-6], as shown in Fig. 1 [7].

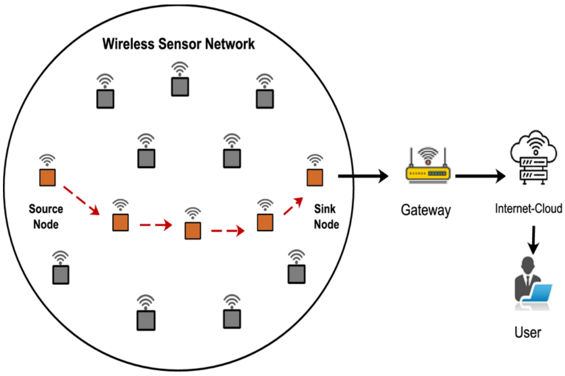


Fig. 1. The architecture of WSN for IoT applications.

WSNs have a major role in IoT technologies. All Technologies are moving towards designing wireless sensor nodes distinguished by fast CPUs and low-power radio connectivity to contribute to WSNs and IoT applications [8, 9]. WSN comprises numerous minor devices called “sensor nodes”. These nodes can monitor, sense, and collect data from the surroundings. The gathered data will be processed and directed to the target node (sink node) using routing protocols [4, 10].

A certain amount of energy will be consumed during these processes. These nodes have batteries and energy. If the sensor battery dies and the power supply is not replaced quickly, it becomes disabled. This will hinder the transfer of the data that has been gathered. Consequently, one major problem in real-world remote sensing applications has been to extend the lifespan of the entire WSN [10-14].

WSNs are used in many fields such as in greenhouses to monitor crucial environmental factors for plant growth, such as for a smart irrigation system, a WSN can be used to monitor the field and control watering status [15, 16]. A Nemours of researchers and academics have conducted several studies to optimize overall network energy usage, as shown in Section 2.

This paper adds enhancements to the ACO by dynamically adjusting alpha (α) and beta (β) parameters during the iteration of ACO. This approach balances exploration and exploitation by modifying how much pheromone trails (alpha) and heuristic information (beta) influence the ants' decision-making process. The proposed ACO algorithm is applied for WSNs with 20, 50, and 100 nodes respectively, and the results of the simulations are expressed in section IV. The outcomes prove that the proposed ACO algorithm shows performance better than the ACO in terms of energy consumption. While the traditional ACO's stochastic nature introduces variability in performance the proposed one is more stable.

The organization of the paper is as follows: Section 2 introduces recent related works on the IoT, WSNs, and energy efficiency. Section 3 describes the problem modelling and case study. Section 4 presents the proposed methods for improving energy efficiency in WSNs using the enhanced ACO. Section 5 details the simulation settings and outcomes. Section 6 shows the conclusion and future works.

2. Related Works

Kaur and Mahajan [17] suggested A hybrid Ant Colony Optimization with Particle Swarm Optimization (ACOPSO)-based clustering protocol for energy efficiency in WSNs. The proposed protocol divides the WSN into several clusters and for each cluster, cluster heads (CHs) are selected. After that, the sensing data is collected directly from each CH by tree-based data aggregation using short-distance connections. The proposed ACOPSO protocol selects the shortest path from the sink node to every CH.

To test and evaluate the proposed method's performance with the current technique General Self-Organized Tree-Based Energy-Balance Routing Protocol (GSTEB) using the following metrics: network lifetime, throughput, stability period, and residual energy, A MATLAB simulation consists of 100 sensor nodes allocated randomly in a 100×100 area with one sink node is used. The outcomes show that the proposed hybrid protocol noticeably improves the WSN lifespan.

Abderrahim et al. [18] proposed an energy-conserving algorithm using the Dijkstra algorithm. The Dijkstra algorithm is applied to WSN after it has been segmented into several clusters, so each node fits into the nearby CH according to the distance to the CH. According to their distance from the source node CH is in charge of informing the nodes in its cluster to be sleeping or active. The proposed path selection algorithm considers the algorithm that selects a list of appropriate paths to transmit data from the source node to the CH with the lowest energy cost.

Abed et al. [19] suggested a model consisting of sensor nodes, one macro base station, and several micro base stations. The micro base station transmits the data that it collects from sensor nodes to the macro base station. The total WSN’s energy consumption is reduced using the suggested Restart Artificial Bee Colony (RABC) algorithm. The suggested model consists of three stages for optimization problems. These stages are used to select the optimal uplink route, find the shortest travel path, and for mutual interference reduction. The simulation outcomes found optimal and near-optimal solutions by the RABC method.

Jain and Agrawal [20] enhanced the Low Energy Adaptive Clustering Hierarchy (LEACH) protocol to solve the sensor’s battery limitation and WSN lifetime problem. LEACH protocol is a categorized protocol that utilizes the CH selection algorithm and cache node selection for data transmission from the source node to the destination node. Based on comparing energy with a threshold the CHs are selected. Also, based on the minimum distance the transmitted data from CH to the cache node is completed. The replacement of cache node memory is organized based on priority. They evaluate the performance of the proposed method based on several standard parameters such as throughput, average residual energy, and live or dead nodes in the network.

Prajapat et al. [21] proposed for cognitive radio sensor networks (CRSN) a neighbour finding algorithm and two greedy k-hop clustering schemes K-hop Spectrum Aware Clustering for Bi-channel connectivity with Edge Contraction (k-SACB-EC), and k-hop Spectrum Aware Clustering for Bi-channel connectivity Without Edge Contraction (k-SACB-WEC). The main objective of this study is to increase network lifetime and attain bi-channel connectivity. Several parameters are considered such as nodes' primary users (PUs) appearance probability of channels, residual energy, channel quality, spectrum awareness, and robustness on PUs' arrival.

Simulation is used to evaluate the suggested method concerning the network lifespan, number of clusters, network stability, and re-clustering frequency. The k-SACB-WEC generates a number of clusters less than Network Stability Aware Clustering (NSAC), k-SACB-EC, Prolong Stable Election Protocol (PSEP), Cognitive LEACH (CogLEACH), and Spectrum Aware Clustering with Weighted Channel Metric (SAC-WCM) by 40%. Moreover, the k-SACB-WEC attains as a minimum around 100% greater number of rounds before the first node died than the related methods concerning network stability.

Abdulzahra et al. [22] proposed a Bacterial foraging optimization routing protocol (BFORP) as a routing protocol for energy-saving to explore the problem of the WSNs' lifespan. By reprocessing the data that visits the source node frequently to the sink it can reduce the routing of extreme messages that may cause energy waste. In the proposed method, the lowest traffic load, the shortest path to the sink, and the maximum residual energy are considered to select nodes for the sending routes. The simulation outcomes proved the efficiency of the proposed protocol in reducing energy consumption and the end-to-end delay.

Mustafa and Hamza [23] improved the Grey Wolf method with Particle Swarm Optimization and Tabu Search Techniques (GW-IPSO-TS) to enhance the CHs selection according to the CH probabilistic reasoning of sensor nodes, and also enhance the routing path of each CH to the base station. The suggested model delivered the optimal routing paths and enhanced the overall WSNs' lifetime, the packet loss rate, and end-to-end delay.

Gunigari and Chitra [13] suggested a reliable and energy-saving hybrid method, based on ACO, Efficient and Reliable ACO Routing protocol (E-RARP), and game theory clustering algorithm (GEC). Communications reliability and high-quality channels of communication are provided by the proposed protocol to enhance energy consumption. every sensor node is considered a team player in WSNs using a GEC, So it can select for itself a beneficial method, defined by the duration of inactive playback time in the active phase, and afterward choose whether to rest or not. The suggested E-RARP-GEC enhanced the WSNS' lifetime; it also consumes a minimum amount of energy when compared with other proposed methods.

However, a major challenge in the deployment of WSNs is to efficiently use energy to extend the network's operating lifetime. The present literature has investigated a variety of optimization strategies, some of which integrate multiple optimization methods, routing protocols, and clustering methods. However, there are still problems to be overcome concerning overall energy efficiency, path validity, scalability, and computing complexity. Considering these attempts, a more effective methodology is still required to get beyond the drawbacks of the present approaches to lower computational overhead, enhance the viability of the solutions, and improve energy efficiency across the network.

3. Problem Modelling and Case Study

The scenario studied in this paper is about energy efficiency in WSNs using dynamic parameter adjustment in ACO. The WSNs comprise 20, 50, and 100 sensor nodes, along with a central sink node.

3.1. Network setup

Node Positions: Sensor nodes are randomly placed to effectively cover the WSN area. three topologies are used for 20, 50, and 100 nodes. Sink node: Strategically situated to receive and process data sent by the sensor nodes. The sensors are distributed to gather data and transmit it to the sink node for monitoring via multi-hop communication.

3.2. Problem statement

The key limitation of WSNs is the battery capacity of the sensor nodes. So, the main challenge is to decrease the energy consumed by WSNs while confirming efficient data transmission from sensor nodes to the sink node.

3.3. Objectives

This paper proposes an enhancement of the ACO algorithm. It evaluates and compares the performance of the proposed ACO with the traditional ACO and the Dijkstra algorithms for routing data in WSNs. It suggests different network topologies (20, 50, and 100 nodes) and discusses their energy consumption and routing efficiency.

The main objective of this work is to balance the exploration and exploitation of the ACO by modifying how much pheromone trails (alpha) and heuristic information (beta) influence the ants' decision-making process to enhance energy consumption by selecting optimal routes to the sink node. This enhancement improves ACO performance when selecting the optimal route to save energy in the WSN.

The objective function for the proposed model is the minimum energy cost of selected routes to improve the energy use of sensor nodes and improve the lifespan of the entire WSN. Equations (1)-(3) can be used to determine the amount of energy used by a single node during transmission [24].

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

where Eelec is the energy (measured in Joules/bit) that the transmitter or receiver circuitry uses. Eamp (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).

$$E_{rx} = E_{elec} \times L \tag{2}$$

So, the energy consumed by a single node is:

$$E_{node} = E_{tx} + E_{rx} \tag{3}$$

The cost function is the total energy consumption of the selected path that is given by Eq. (4).

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

4. Proposed Energy Efficient Methods

The proposed method for energy efficiency in WSNs is based on two algorithms the Dijkstra algorithm and the Enhanced ACO algorithm to select the minimum energy cost path between the source node and the sink node. The two proposed algorithms are illustrated below.

4.1. Ant colony optimization algorithm

ACO is a nature spread optimization algorithm developed in mid-1990 by Macro Dorigo. It uses artificial ants to mark paths with pheromones, intensifying trails on shorter paths to guide exploration toward more efficient routes [25-27].

The main steps of the ACO are illustrated below [12, 23-25]:

Step 1: *Initialization*

a. Pheromone trails initialization: to initialize the pheromone levels, assign values to the edges of the graph.

b. Ant positions initialization: Place artificial ants on source nodes.

Step 2: *Ant Movement*

Ants probabilistically choose their next movement based on pheromones and heuristics.

The probability of k^{th} ant at i^{th} node choosing j^{th} node as the next node expressed in Eq. (5):

$$p_{ij}^{k} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{i \in allowed\ nodes} \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}} \quad (5)$$

where, τ_{ij} denotes the pheromone level on edge $(i \to j)$, η_{ij} represents the heuristic information, and the parameters α and β are used to regulate the impact of pheromone and heuristic data.

Step 3: *Move Ants*

Artificial ants select their next node based on probabilities.

Step 4: *Pheromone Update*

a. Pheromone Evaporation: to simulate evaporation, decrease the levels of pheromone on all edges as shown in Eq. (6):

$$\tau_{ij} = (1 - \rho).\tau_{ij} \quad (6)$$

Where ρ represents the Pheromone evaporation rate and $(0 < \rho < 1)$.

b. Pheromone Deposition: Ants leave a trail of pheromones along the edges they travel, which is based on the quality of the solution.

Ant k traverses edge (i→j), and the pheromone level on this edge is updated using Eq. (7):

$$\Delta\tau_{ij}^{k} = \frac{1}{total\ cost\ chosen\ by\ ant\ k} \tag{7}$$

The total cost in this work is the total energy consumption chosen by ant k.

c. Global Update: to update the pheromone levels on edges taking into account all the ant solutions as expressed in Eq. (8), where m is the ant's number.

$$\tau_{ij} = (1-\rho).\tau_{ij} + \sum_{k=1}^{m}\Delta\tau_{ij}^{k} \tag{8}$$

Step 5*: Termination*

Iterate Ants movement, pheromone update, and solution construction till a termination condition happens (e.g., reach the max iteration number). ACO adjusts pheromone levels on edges to bias ant movements toward high concentrations, promoting the exploration of efficient routes while exploiting prior information.

The Enhancements*: Dynamic Alpha and Beta*

To balance between exploration and exploitation, adaptively adjust α and β over iterations. If the iteration index is t and the number of iterations is T. The dynamic parameters adjustment can be defined in Equations (9) and (10), where α_{init} , α_{max}, and β_{init} are predefined constants. Figure 2 shows the flowchart of the proposed enhanced ACO and Algorithm 1 illustrates the proposed enhanced ACO:

$$\alpha(t) = \alpha_{init} + (\alpha_{max} - \alpha_{init}).\frac{t}{T} \tag{9}$$

$$\beta(t) = \beta_{init} - \beta_{init}.\frac{t}{T} \tag{10}$$

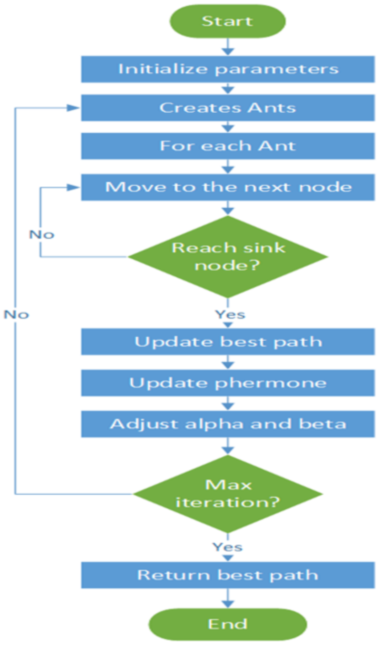


Fig. 2. The flowchart of the proposed ACO.

Algorithm 1 : The Proposed Enhanced ACO

Input: sensor_nodes, source_node_id, num_ants, num_iterations, battery_capacity, transmission_range, alpha = 0.1, beta=0.1, evaporation_rate = 0.5

Output: best path

```
1. CLASS Ant:
   a. METHOD init (source _node_id, nodes, transmission_range):
      - Initialize current_node_id, path, total_distance, total_energy_consumed,
total_hops, nodes, transmission_range
   b. METHOD choose_next_node(pheromone_matrix, alpha, beta):
      - Calculate probabilities for next node based on pheromone_matrix, alpha, beta
      - RETURN next node index or None if no valid node
   c. METHOD move_to_node(next_node_id):
      - IF next_node_id is not None:
         - Move to next_node_id
         - Update total_distance, total_energy_consumed, total_hops
         - Update current_node_id and path
2. CLASS ACO_WSN:
   a. METHOD __init__(positions, source_node_id, num_ants, num_iterations):
      - Initialize nodes, source_node_id, num_ants, num_iterations, transmission_range,
pheromone_matrix, alpha, beta, evaporation_rate
   b. METHOD run():
      - Initialize best_path_id to None, best_fitness to infinity
      - FOR each iteration:
         - Create ants
         - FOR each ant:
            - WHILE ant has not reached destination:
               - Choose and move to next node
            - IF ant reached destination:
               - Calculate fitness
               - Update best_path_id, best_fitness if fitness is better
         - Update pheromones
         - Adjust parameters
      - RETURN best path or empty list if no path found
   c. METHOD update_pheromones(ants):
      - Evaporate pheromones
      - FOR each ant:
         - Update pheromones based on ant's path
   d. METHOD adjust_parameters(iteration):
       - SET iteration_ratio = iteration / self.num_iterations
      - SET self.alpha = self.initial_alpha + (1 - self.initial_alpha) × iteration_ratio
       - SET self.beta = self.initial_beta × (1 - iteration_ratio)
3. Main execution:
   - Initialize ACO_WSN with sensor_nodes, source_node_id, num_ants,
num_iterations
   - Run ACO_WSN and get best path
```

4.2. The Dijkstra algorithm

The Dijkstra algorithm is a widespread algorithm to resolve shortest-path problems using graphs with non-negative edge weights. it is used to obtain the shortest path between two vertices on a graph with minimum distance cost. It was created by computer scientist Edsger W. Dijkstra in 1956 [18, 28, 29]. The algorithm does its computation using a set of visited and a set of unvisited vertices of the graph. It begins at the source vertex and iteratively chooses from the unvisited set a vertex with the minimum distance from the source.

After that, the neighbours of this vertex are visited, and their tentative distances are updated if a shorter path is discovered. This procedure continues until all reachable vertices have been visited or the destination vertex has been reached. Figure 3 shows a weighted graph of vertices, the shortest path from node a to node e is a, b, c, and e with distance=6. Dijkstra's Algorithm is highly perceptive, given that every sub-path of the shortest path is a shortest path. The path (a, b, c, e) is the shortest path from node a to node e, and the sub-path (a, b, c) is as well as the shortest path.

The wireless sensor networks can be represented as weighted graphs to choose the optimal path from a source node to the destination node [18, 28, 29].In this work, the Dijkstra algorithm considers the battery level of each node while selecting the best path with minimum energy cost. It is vital to keep away disabled or low battery-level nodes.

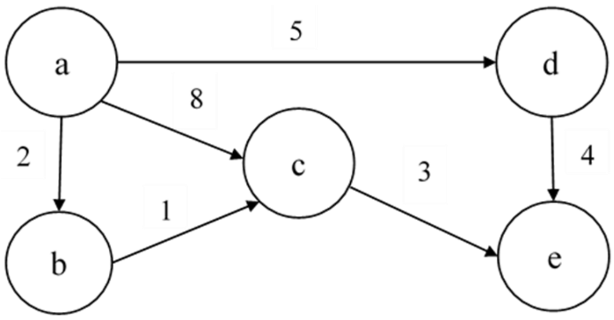


Fig. 3. A weighted graph with 5 nodes.

5. Simulations Setting and Results Analysis

In this section, the simulation setup and the main results conducted by the proposed methodology are expressed. In our previous work we explored the use of Variable Length Genetic algorithm and the Dijkstra algorithm for energy efficiency in WSN [30, 31], in this study the enhance ACO is investigate and the output is compared with the traditional ACO.

5.1. Simulation setup

The simulations aimed to assess the energy efficiency of WSNs using the Dijkstra algorithm, traditional ACO, and improved ACO. The experiments were conducted on WSNs of varying sizes, including 20, 50, and 100 sensor nodes, to analyse performance across different network scales comprehensively. The metrics evaluated were energy consumption, path optimality, and computational time.

Tables 1-3 display the positions of nodes in the WSN topologies of sizes 20, 50, and 100, respectively. The area size is 200×200 for the 20-topology and 500×500 for the 50 and 100 topologies. The energy levels are randomly initialized from 0 to the initial battery capacity of 3×2.85 Ah. The distance range is 100 meters, and the packet size is 128 bytes.

Table 1. The coordinates of 20 Sensor Nodes in the 20-Topology WSN.

Sensor	X	Y	Sensor	X	Y
1	37	117	**12**	58	85
2	116	197	**13**	196	79
3	44	180	**14**	45	52
4	100	187	**15**	48	36
5	89	110	**16**	94	88
6	129	28	**17**	104	160
7	136	31	**18**	103	53
8	20	188	**19**	142	118
9	116	67	**20**	96	70
10	12	168	**Sink Node**	100	100
11	165	52			

Table 2. The coordinates of 50 Sensor Nodes in the 50-Topology WSN.

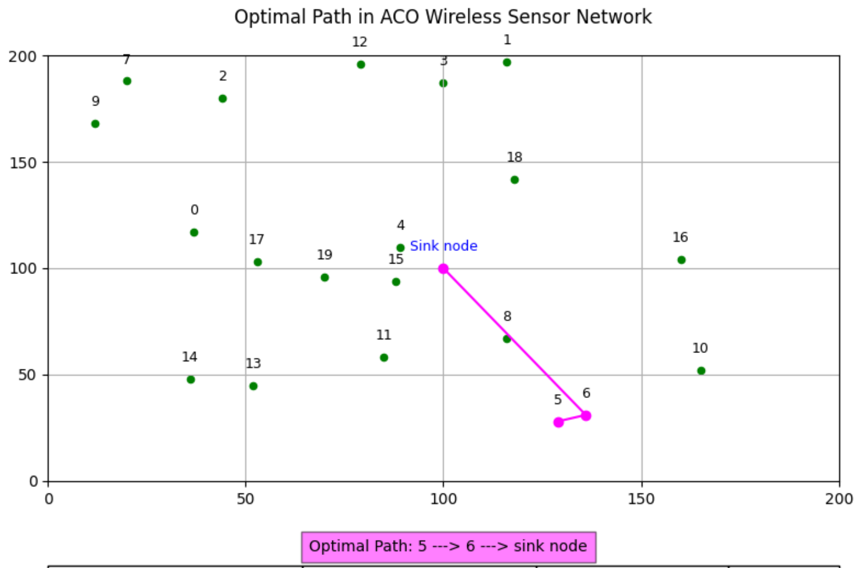
Sensor	X	Y	Sensor	X	Y
0	74	235	**26**	405	418
1	232	487	**27**	192	82
2	394	89	**28**	438	332
3	361	201	**29**	326	63
4	374	179	**30**	92	2
5	221	259	**31**	308	202
6	409	56	**32**	75	396
7	272	62	**33**	435	289
8	41	377	**34**	83	98
9	233	134	**35**	85	430
10	24	337	**36**	13	467
11	477	331	**37**	340	121
12	484	104	**38**	496	229
13	171	117	**39**	409	326
14	158	429	**40**	472	461
15	393	104	**41**	198	64
16	491	91	**42**	319	283
17	72	96	**43**	387	495
18	457	177	**44**	26	311
19	189	320	**45**	126	367
20	209	493	**46**	316	396
21	427	107	**47**	475	259
22	206	489	**48**	365	148
23	236	284	**49**	353	302
24	74	235	**50**	173	270
25	141	461	**Sink Node**	250	250

Table 3. The coordinates of 100 Sensor Nodes in the 100-Topology WSN.

Sensor	X	Y	Sensor	X	Y	Sensor	X	Y
0	74	235	**35**	13	467	**70**	441	167
1	232	487	**36**	340	121	**71**	84	27
2	394	89	**37**	496	229	**72**	385	140
3	361	201	**38**	409	326	**73**	361	356
4	374	179	**39**	472	461	**74**	61	367
5	221	259	**40**	198	64	**75**	25	400
6	409	56	**41**	319	283	**76**	304	453
7	272	62	**42**	387	495	**77**	112	453
8	41	377	**43**	26	311	**78**	158	268
9	233	134	**44**	126	367	**79**	318	110
10	24	337	**45**	316	396	**80**	333	411
11	477	331	**46**	475	259	**81**	383	372
12	484	104	**47**	365	148	**82**	290	182
13	171	117	**48**	353	302	**83**	170	233
14	158	429	**49**	74	235	**84**	6	353
15	393	104	**50**	232	487	**85**	58	243
16	491	91	**51**	173	270	**86**	98	420
17	72	96	**52**	349	175	**87**	268	60
18	457	177	**53**	277	437	**88**	368	304
19	189	320	**54**	482	185	**89**	107	22
20	209	493	**55**	247	205	**90**	199	371
21	427	107	**56**	389	31	**91**	479	53
22	206	489	**57**	416	51	**92**	205	347
23	236	284	**58**	455	153	**93**	365	77
24	141	461	**59**	333	222	**94**	316	337
25	405	418	**60**	129	122	**95**	329	244
26	192	82	**61**	433	421	**96**	478	279
27	438	332	**62**	465	361	**97**	19	358
28	326	63	**63**	365	223	**98**	227	82
29	92	2	**64**	392	454	**99**	25	408
30	308	202	**65**	262	410	**Sink Node**	250	250
31	75	396	**66**	252	461			
32	435	289	**67**	298	130			
33	83	98	**68**	75	29			
34	85	430	**69**	346	299			

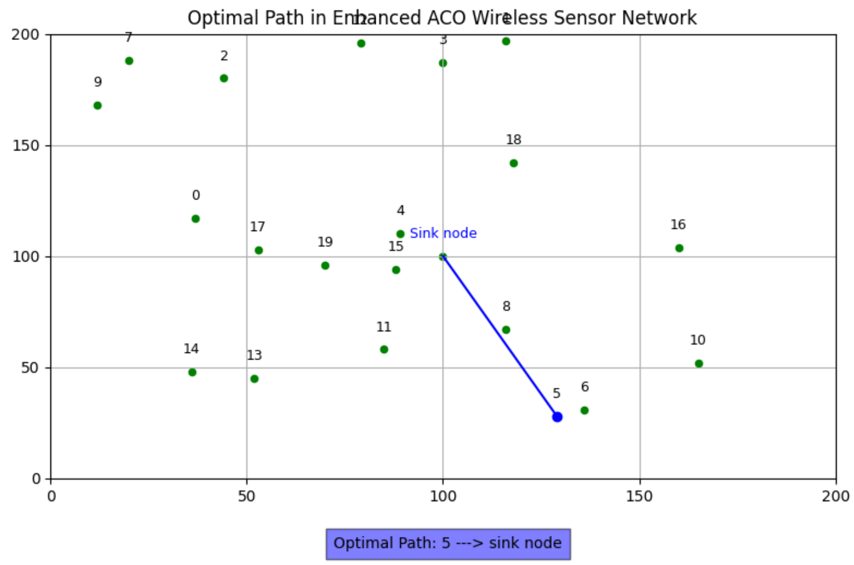
5.2. Simulation results and analysis

The energy efficiency of the enhanced ACO, Dijkstra's algorithm, and traditional ACO in WSNs are compared in this section. The analysis reveals significant differences in energy cost and stability between both the enhanced ACO and the traditional ACO. The result of the three topologies (20,50, and 100) are expressed. Figures 4-6 show the optimal routes from sensor 5 to the sink node selected by the ACO, Enhanced ACO, and the Dijkstra algorithm, and Figs. 7-9 show the selected routes from sensor 9 to the sink node by the three mentioned methods in the 20-sensor topology. Enhanced ACO improves energy utilization by selecting paths with minimal energy consumption.



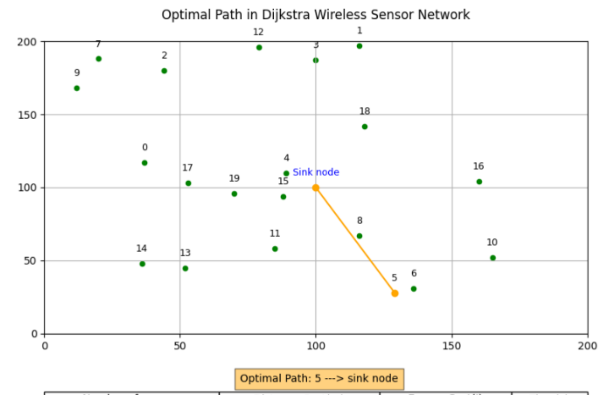
Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
2	85.44250	0.00083	0.06542

Fig. 4. The optimal route selected by the ACO algorithm from sensor 5 to the sink node (20-sensor topology).



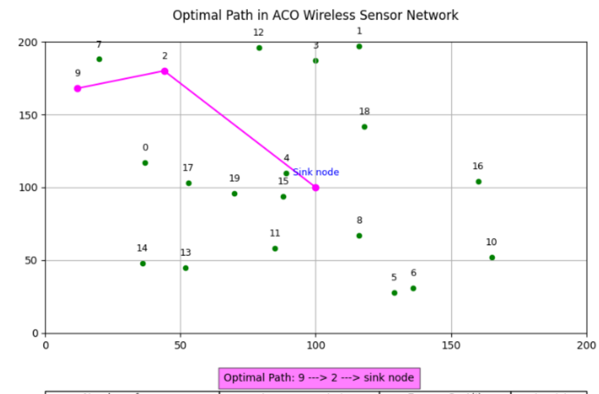
Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
1	77.62087	0.00072	0.04788

Fig. 5. The optimal route selected by the Enhanced ACO algorithm from sensor 5 to the sink node (20-sensor topology).



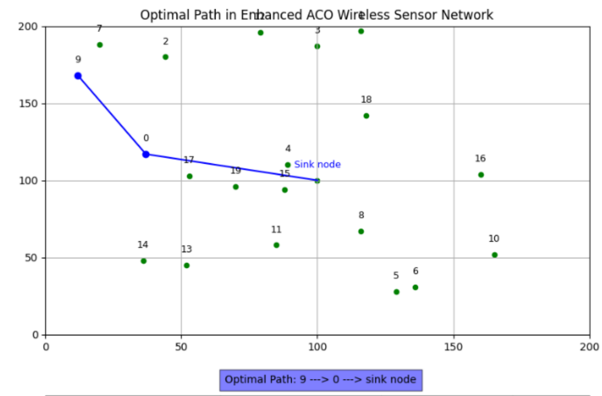
Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
1	77.62087	0.00072	0.00004

Fig. 6. The optimal route selected by Dijkstra's algorithm from sensor 5 to the sink node (20-sensor topology).



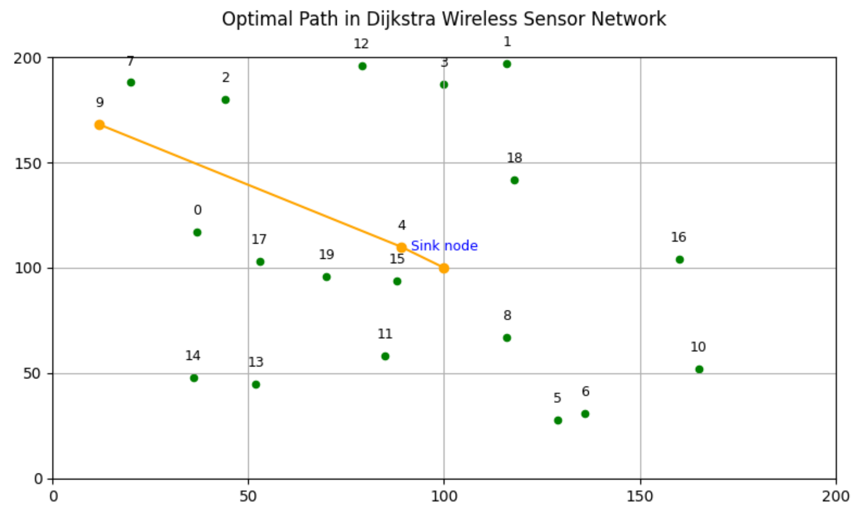
Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
2	131.82846	0.00130	0.06014

Fig. 7. The optimal route selected by the ACO algorithm from sensor 9 to the sink node (20-sensor topology).



Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
2	122.05124	0.00097	0.09988

Fig. 8. The optimal route selected by the Enhanced ACO algorithm from sensor 9 to the sink node (20-sensor topology).

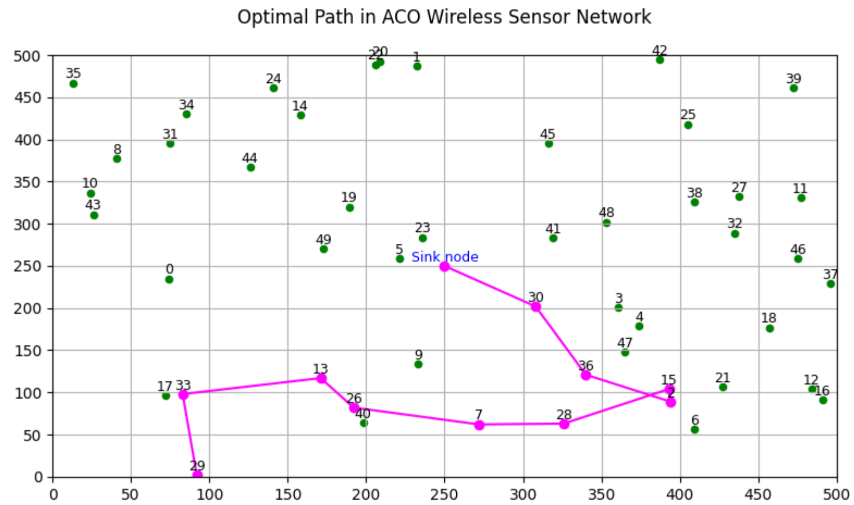


Optimal Path: 9 ---> 4 ---> sink node

Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
2	111.26628	0.00118	0.00007

Fig. 9. The optimal route selected by Dijkstra's algorithm from sensor 9 to the sink node (20-sensor topology).

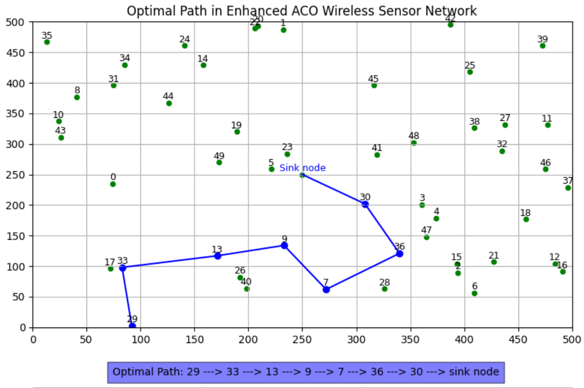
Figures 10-12 show the routes from sensor 29 to the sink node and Figs. 13-15 display the routes from sensor 39 to the sink node. These routes are selected by the three algorithms to show the performance of the proposed method when increasing the number of sensors to 50.



Optimal Path: 29 ---> 33 ---> 13 ---> 26 ---> 7 ---> 28 ---> 15 ---> 2 ---> 36 ---> 30 ---> sink node

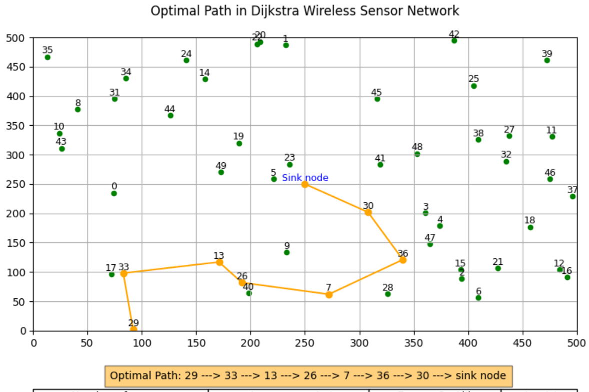
Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
10	682.46685	0.00639	0.18523

Fig. 10. The optimal route selected by the ACO algorithm from sensor 29 to the sink node (50-sensor topology).



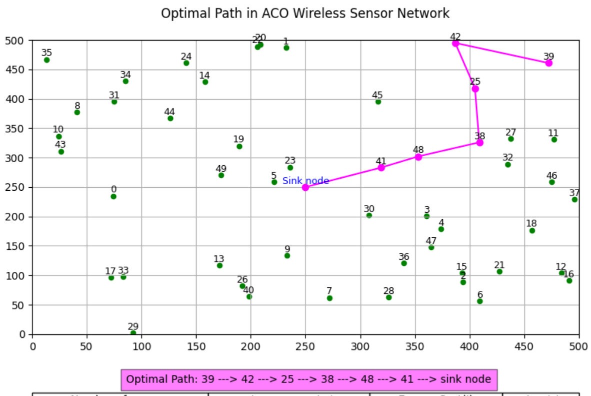
Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
7	585.02700	0.00580	0.19621

Fig. 11. The optimal route selected by the Enhanced ACO algorithm from sensor 29 to the sink node (50-sensor topology).



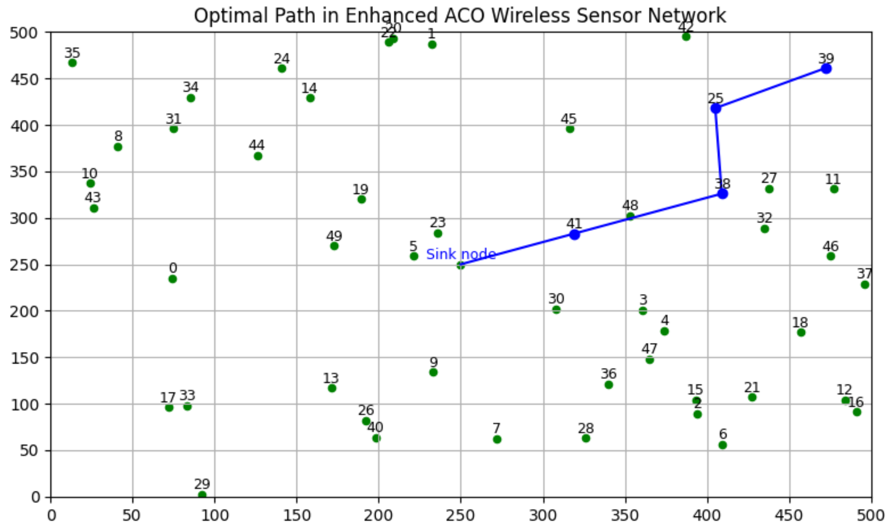
Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
7	562.13330	0.00555	0.00009

Fig. 12. The optimal route selected by Dijkstra's algorithm from sensor 29 to the sink node (50-sensor topology).



Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
6	439.07079	0.00412	0.14133

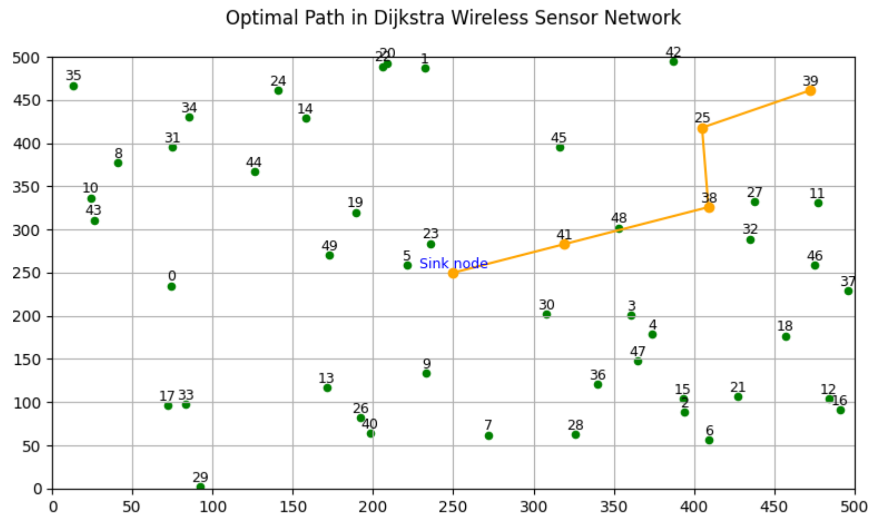
Fig. 13. The optimal route selected by the ACO algorithm from sensor 39 to the sink node (50-sensor topology).



Optimal Path: 39 ---> 25 ---> 38 ---> 41 ---> sink node

Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
4	347.92844	0.00354	0.36601

Fig. 14. The optimal route selected by the Enhanced ACO algorithm from sensor 39 to the sink node (50-sensor topology).

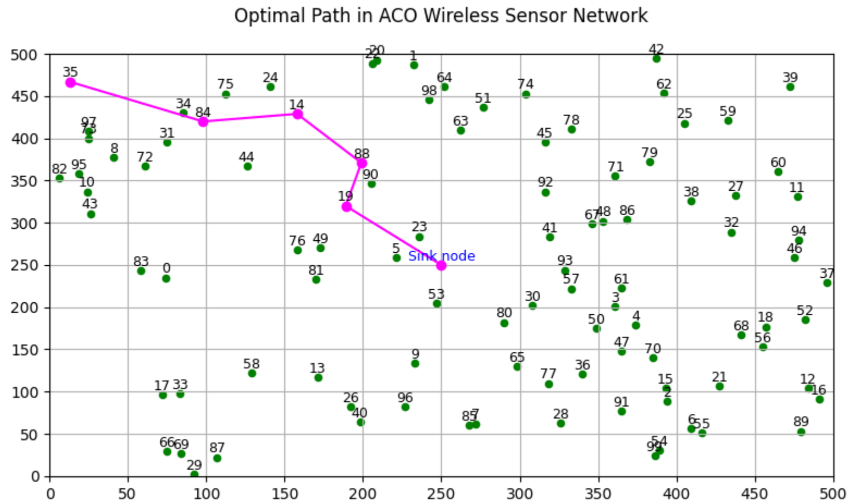


Optimal Path: 39 ---> 25 ---> 38 ---> 41 ---> sink node

Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
4	347.92844	0.00354	0.00008

Fig. 15. The optimal route selected by Dijkstra's algorithm from sensor 39 to the sink node (50-sensor topology).

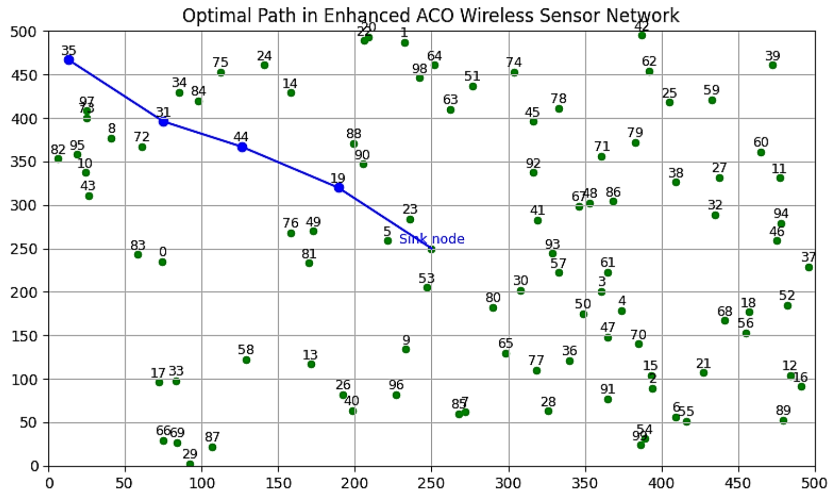
By increasing the number of sensors to 100 the challenges of choosing the optimal route are increased. The proposed method is applied to the 100-sensor topology to demonstrate its performance. It outperforms the traditional ACO in terms of energy consumption, but the dynamic adjustment of the alpha and beta values leads to increased execution time. Figures 16-18 show the routes from sensor 35 to the sink node, while Figs. 19-21 show the routes from sensor 52 to the sink node selected by the three methods.



Optimal Path: 35 ---> 84 ---> 14 ---> 88 ---> 19 ---> sink node

Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
5	373.64868	0.00353	0.20857

Fig. 16. The optimal route selected by the ACO algorithm from sensor 35 to the sink node (100-sensor topology).



Optimal Path: 35 ---> 31 ---> 44 ---> 19 ---> sink node

Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
4	324.37843	0.00319	2.34910

Fig. 17. The optimal route selected by the Enhanced ACO algorithm from sensor 35 to the sink node (100-sensor topology).

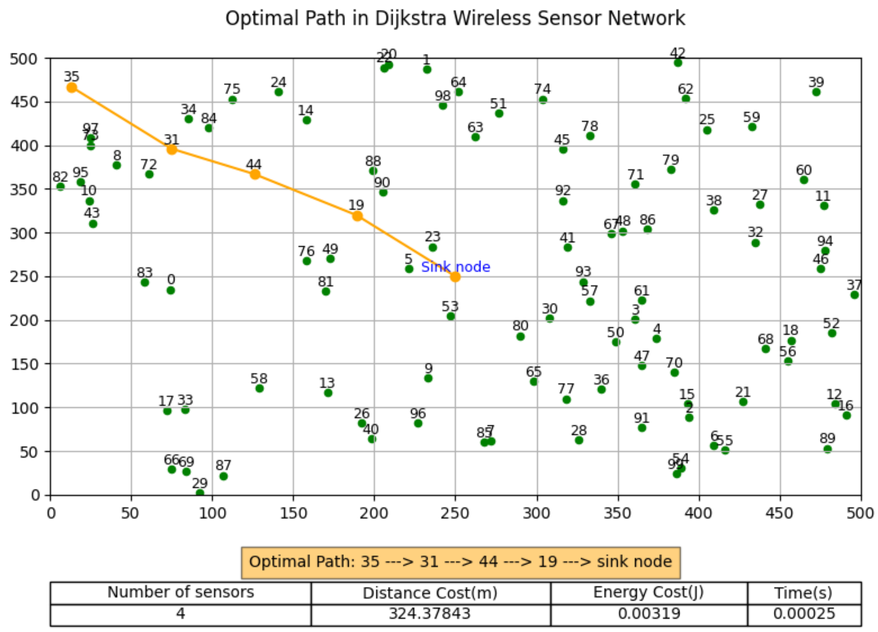


Fig. 18. The optimal route selected by Dijkstra's algorithm from sensor 35 to the sink node (100-sensor topology).

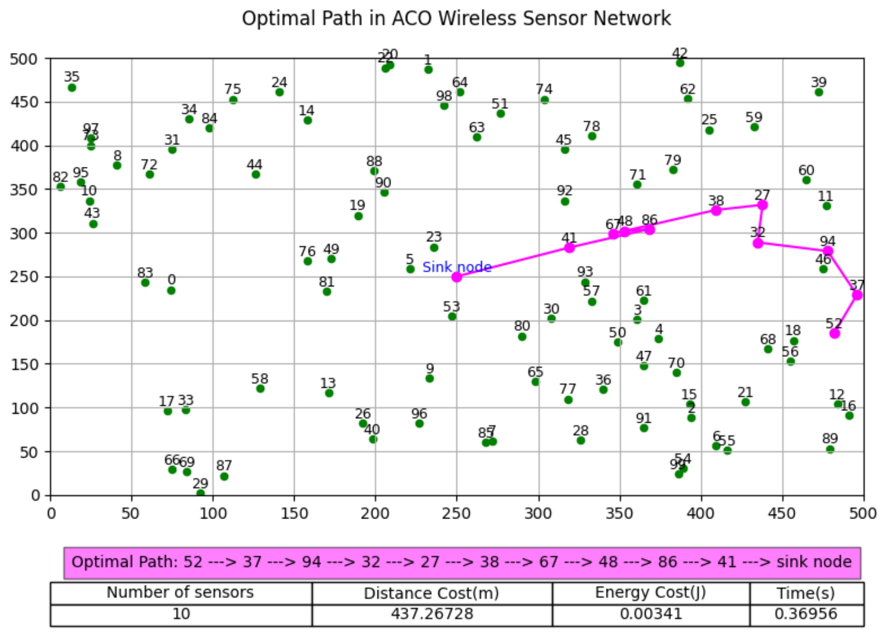
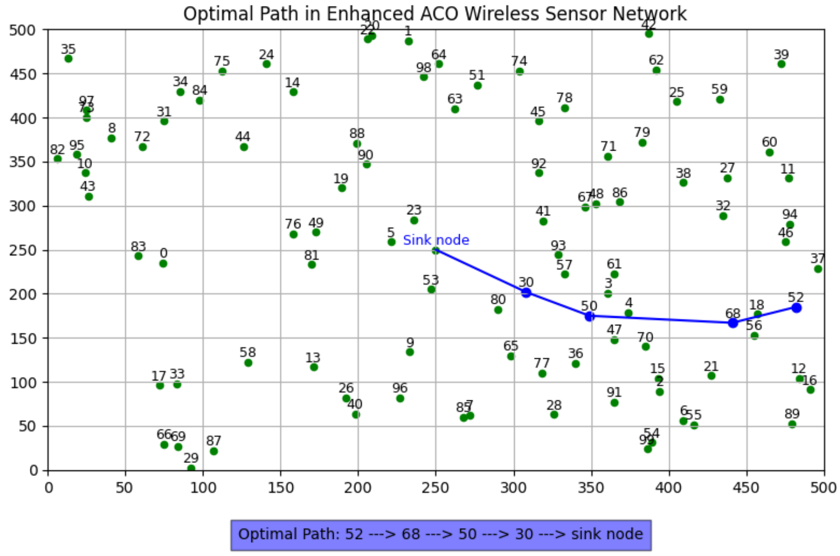
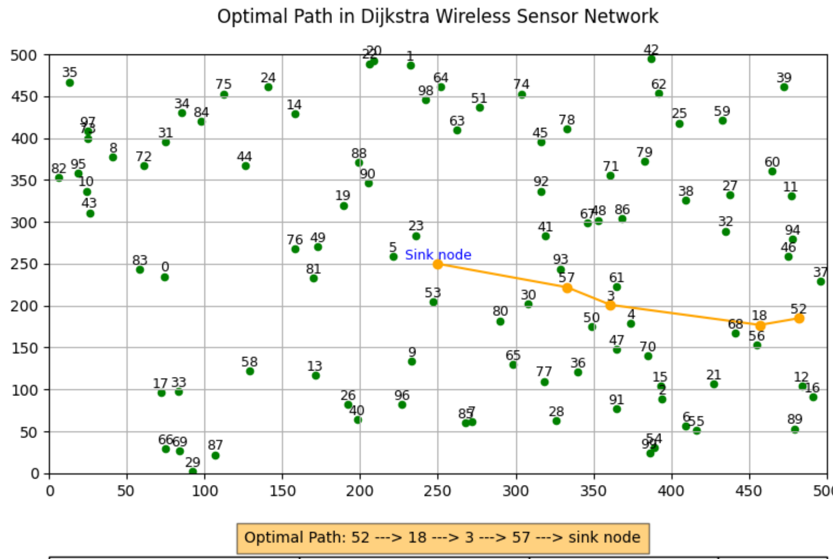


Fig. 19. The optimal route selected by the ACO algorithm from sensor 52 to the sink node (100-sensor topology).



Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
4	261.50227	0.00232	1.86523

Fig. 20. The optimal route selected by the Enhanced ACO algorithm from sensor 52 to the sink node (100-sensor topology).



Number of sensors	Distance Cost(m)	Energy Cost(J)	Time(s)
4	247.79901	0.00239	0.00026

Fig. 21. The optimal route selected by Dijkstra's algorithm from sensor 52 to the sink node (100-sensor topology).

Figures 22-24 show the average run time, energy consumption, and distance for each algorithm running within the three topologies. Despite adding a slight complexity that requires additional time in implementation, the performance of the proposed enhanced ACO algorithm is better and more stable than the traditional ACO.

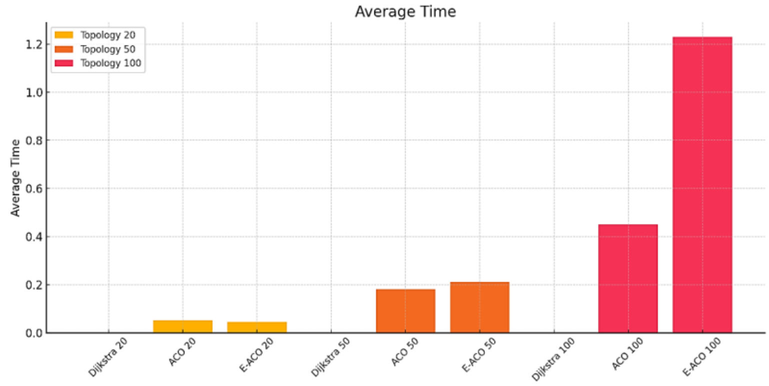


Fig. 22. The average run time of the three methods running within the three topologies.

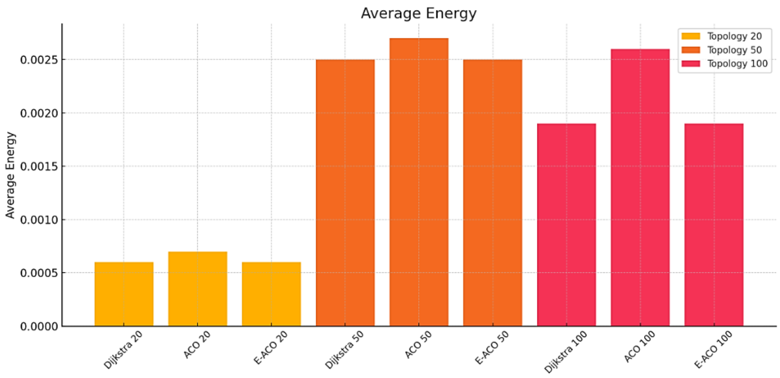


Fig. 23. The average energy consumption of the three methods running within the three topologies.

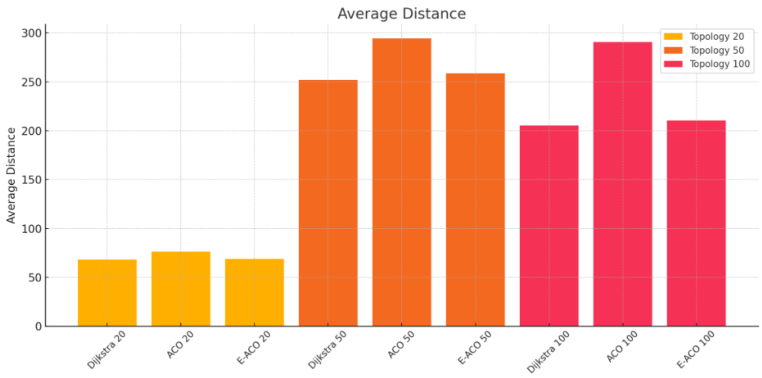


Fig. 24. The average run time of the three methods running within the three topologies.

This study can be improved or extended by suggesting other optimization methods like Bee Algorithm, Pelican Optimization Algorithm, Cuckoo Search Optimization, Social Spider optimization, Whale Optimization Algorithm, and Particle Swarm

Optimization [32-39]. One can present real-time implementation of proposed method utilizing embedded design systems like a LabView software platform [40-45].

6. Conclusion

This study discusses the enhancement that is added to the ACO by dynamically adjusting alpha and beta parameters during the iterations of ACO to balance exploration and exploitation. This improvement enhances ACO performance to save energy in WSNs by selecting optimal routes with minimum hops. A comparison between traditional ACO and proposed ACO is conducted in terms of energy consumption, also the outcomes of enhanced ACO are compared with the outcomes of the Dijkstra algorithm.

Enhanced ACO outperforms the traditional ACO in terms of energy efficiency across all tested topologies. Enhanced ACO and the Dijkstra algorithm achieved average energy of about 0.0006, 0.0025, and 0.0019 joules for the 20, 50, and 100 sensor topologies, respectively. While the traditional ACO achieved 0.0007, 0.0027, and 0.0026 joules. The enhancements in the proposed method led to consistent energy savings. These improvements make enhanced ACO a more suitable option for applications where energy efficiency is a critical concern.

Although Dijkstra's algorithm performs consistently well in optimizing energy consumption within WSNs, it can be inefficient for larger networks. Its deterministic nature enables precise route calculation, minimizing energy dissipation throughout the network. Dijkstra's algorithm is computationally efficient, providing an advantage in scenarios where consistency and low complexity are essential. Enhanced ACO improves the performance of ACO, it selects optimal paths and shows stability throughout different runs.

Nomenclatures

d	Distance between nodes
E_{amp}	Amount of energy consumed to transmit a bit over the air
E_{elec}	Amount of energy consumed by the transmitter or receiver circuitry
E_{rx}	Energy consumed by a single node during reception
E_{tx}	Energy consumed by a single node during transmission
$\mathcal{L}$	Number of transmitted bits

Greek Symbols

α	Pheromone trails
β	Heuristic information
ρ	Pheromone evaporation rate
p	Probability in the ACO
τij	Pheromone level on edge (i→j)
ηij	Heuristic information

Abbreviations

ACO	Ant Colony Optimization
CHs	cluster heads
IoT	Internet of Things
WSNs	Wireless Sensor Networks

References

1. Patel, B.C.; Sinha, G.R.; and Goel, N. (2020). *Introduction to sensors*. In Sinha, G.R. (Ed), *Advances in Modern Sensors*. *IOP Publishing*, 1-21.
2. Al Hasan, R.A.; and Hamza, E.K. (2023). An improved intrusion detection system using machine learning with singular value decomposition and principal component analysis. *International Journal of Intelligent Engineering & Systems*, 16(4), 25-38.
3. Ray, P.P. (2018). A survey on internet of things architectures. *Journal of King Saud University-Computer and Information Sciences*, 30(3), 291-319.
4. Al-Ani, K.W.; Abdalkafor, A.S.; and Nassar, A.M. (2020). An overview of wireless sensor network and its applications. *Indonesian Journal of Electrical Engineering and Computer Science*, 17(3), 1480-1486.
5. Javaid, M.; Haleem, A.; Rab, S.; Pratap Singh, R.; and Suman, R. (2021). Sensors for daily life: A review. *Sensors International*, 2, 100121.
6. Ahmad, I.A.; Al-Nayar, M.M.J.; and Mahmood, A.M. (2022). Investigation of energy efficient clustering algorithms in WSNs: A review. *Mathematical Modelling of Engineering Problems*, 9(6) , 1693-1703.
7. Jaiswal, K.; and Anand, V. (2020). EOMR: An energy-efficient optimal multi-path routing protocol to improve QoS in wireless sensor network for IoT applications. *Wireless Personal Communications*, 111(4), 2493-2515.
8. Nourildean, S.W.; Hassib, M.D.; and Mohammed, Y.A. (2022). Internet of things based wireless sensor network: A review. *Indonesian Journal of Electrical Engineering and Computer Science*, 27(1), 246-261.
9. Alhasan, R.A.; and Hamza, E.K. (2023). A Novel CNN model with dimensionality reduction for WSN intrusion detection. *Revue d'Intelligence Artificielle*, 37(5), 1121-1131.
10. Hamza, E.K.; Salman, K.D.; and Jaafar, S.N. (2023). *Wireless sensor network for robot navigation*. In Azar, A.T.; Kasim Ibraheem, I.; and Jaleel Humaidi, A. (Eds), *Mobile robot*: *Motion control and path planning*. Springer International Publishing, 1090, 643-670.
11. Ramadhan, N.M.; Majeed, A.M.; and Raafat, S.M. (2022). A survey of power consumption minimization in WSN using feedback control strategies. *Iraqi Journal of Computers*, *Communications*, *Control & Systems Engineering* (*IJCCCE*), 22(4), 37-47.
12. Yan, X.; Huang, C.; Gan, J.; and Wu, X. (2022). Game theory-based energy-efficient clustering algorithm for wireless sensor networks. *Sensors*, 22(2), 478.
13. Gunigari, H.; and Chitra, S. (2023). Energy efficient networks using ant colony optimization with game theory clustering. *Intelligent Automation & Soft Computing*, 35(3), 3557-3571.
14. Hamza, E.K.; and 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.
15. Khudhur, D.D.; and Croock, M.S. (2021). Application of self-managing system in greenhouse with wireless sensor network. *Iraqi Journal of Computers*, *Communications*, *Control & Systems Engineering* (*IJCCCE*), 21(1), 53-61.

16. Mohammed, L.A.; Hasan, A.M.; and Hamza, E.K. (2024). Optimizing energy efficiency in wireless sensor networks using dijkstra's algorithm. *Instrumentation Mesure Métrologie*, 23(4), 307-316.
17. Kaur, S.; and Mahajan, R. (2018). Hybrid meta-heuristic optimization based energy efficient protocol for wireless sensor networks. *Egyptian Informatics Journal*, 19(3), 145-150.
18. Abderrahim, M.; Hakim, H.; Boujemaa, H.; and Touati, F. (2019). Energy-Efficient Transmission Technique based on Dijkstra Algorithm for decreasing energy consumption in WSNs. *Proceedings of the* 2019 19th *International Conference on Sciences and Techniques of Automatic Control and Computer Engineering* (*STA*), Sousse, Tunisia, 599-604.
19. Abed, R.A.; Hamza, E.K.; and Humaidi, A.J. (2024). A modified CNN-IDS model for enhancing the efficacy of intrusiondetection system. *Measurement*: *Sensors*, 35, 101299.
20. Jain, S.; and Agrawal, N. (2020). Development of energy efficient modified LEACH protocol for IoT applications. *Proceedings of the* 2020 12th *International Conference on Computational Intelligence and Communication Networks* (*CICN*), Bhimtal, India, 160-164.
21. Prajapat, R.; Yadav, R.N.; and Misra, R. (2021). Energy-efficient k-Hop clustering in cognitive radio sensor network for internet of things. *IEEE Internet of Things Journal*, 8(17), 13593-13607.
22. Abdulzahra, A.A.; Khudor, B.A.Q.; and Alshawi I.S. (2023). Energy-efficient routing protocol in wireless sensor networks based on bacterial foraging optimization. *Indonesian Journal of Electrical Engineering and Computer Science*, 29(2), 911-920.
23. Mustafa, Z.; and Hamza, E.K. (2024).Enhancing hybrid system based on reinforcement learning. *International Journal of Intelligent Engineering and Systems*, 17(1), 596-611.
24. Heinzelman, W.R.; Chandrakasan, A.; and Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. *Proceedings of the* 33rd *Annual Hawaii International Conference on System Sciences*, Maui, HI, USA, 1,10.
25. Dorigo, M.; Maniezzo, V.; and Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, *Part B* (*Cybernetics*), 26(1), 29-41.
26. Dorigo, M.; and Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2-3), 243-278.
27. Nayyar, A.; and Singh, R. (2017). Ant colony optimization (ACO) based routing protocols for wireless sensor networks (WSN): A survey. *International Journal of Advanced Computer Science and Applications*, 8(2), 148-155.
28. Husain, S.S.; Rasheed, L.T.; Mahmod, R.A.; Hamza, E.K.; Noaman, N.M., and Humaidi, A.J. (2023). Design of RISE control for respiratory system. *Proceedings of the* 2023 *IEEE* 8th *International Conference on Engineering Technologies and Applied Sciences* (*ICETAS*), Bahrain, Bahrain.
29. Abderrahim, M.; Hakim, H.; Boujemaa, H.; and Touati, F. (2019). A clustering routing based on dijkstra algorithm for WSNs. *Proceedings of the* 2019 19th

International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (*STA*), Sousse, Tunisia, 605-610.

30. Abdulaziz, W.B.; and Croock, M.S. (2022). Design of smart irrigation system for vegetable farms based on efficient wireless sensor network. *Iraqi Journal of Computers*, *Communications*, *Control & Systems Engineering* (*IJCCCE*), 22(1), 72-85.
31. Mohammed, L.A.K.; Hasan, A.M.; and Hamza, E.K. (2024). Pruning and validation techniques enhanced genetic algorithm for energy efficiency in wireless sensor networks. *Ingénierie des Systèmes d'Information*, 29(4), 1305-1314.
32. Kadhim, R.A.; Kadhim, M.Q.; Al-Khazraji, H.; and Humaidi, A.J. (2024). Bee algorithm based control design for two-links robot arm systems. *IIUM Engineering Journal*, 25 (2). 367-380.
33. Khaleel, H.Z.; Ahmed, A.K.; Al-Obaidi, A.S.M.; Luckyardi, S.; Husaeni, D.F.A.; Mahmod, R.A.; and Humaidi, A.J. (2024). Measurement enhancement of ultrasonic sensor using pelican optimization algorithm for robotic application. *Indonesian Journal of Science & Technology*, 9(1), 145-162.
34. Al-Khazraji, H.; Guo, W.; and Humaidi, A.J. (2024). Improved cuckoo search optimization for production inventory control systems. *Serbian Journal of Electrical Engineering*, 21(2), 187-200.
35. Ghanim, T.; Ajel, A.R.; and Humaidi, A.J. (2020). Optimal fuzzy logic control for temperature control based on social spider optimization. *IOP Conference Series*: *Materials Science and Engineering*, 745(1), 012099.
36. Khaleel, R.Z.; Khaleel, H.Z.; Al-Hareeri, A.A.A.; Al-Obaidi, A.S.M.; and Humaidi, A.J. (2024). Improved trajectory planning of mobile robot based on pelican optimization algorithm. *Journal Européen des Systèmes Automatisés*, 57(4), 1005-1013.
37. Waheed, Z.A.; and Humaidi, A.J. (2022). Design of optimal sliding mode control of elbow wearable exoskeleton system based on whale optimization algorithm. *Journal Européen des Systèmes Automatisés*, 55 (4), 459-466.
38. Al-Jodah, A.; Abbas, S.J.; Hasan, A.F.; Humaidi, A.J.; Al-Obaidi, A.S.M.; AL-Qassar; A.A.; and Hassan, R.F. (2023). PSO-based optimized neural network PID control approach for a four wheeled omnidirectional mobile robot. *International Review of Applied Sciences and Engineering*, 14(1), 58-67.
39. Al-Obaidi, A.S.M.; AL-Qassar; A.; Nasser, A.R.; Alkhayyat, A.; Humaidi, A.J.; and Ibraheem, I.K. (2021). Embedded design and implementation of mobile robot for surveillance applications. *Indonesian Journal of Science & Technology*, 6(2), 427-440.
40. Fadhel, M.A.; Humaidi, A.J.; and Oleiwi, S.R. (2017). Image processing-based diagnosis of sickle cell anemia in erythrocytes. *Proceedings of the* 2017 *Annual Conference on New Trends in Information & Communications Technology Applications* (*NTICT*), Baghdad, Iraq, 203-207.
41. Hassan, M.Y.; Humaidi, A.J.; and Hamza, M.K. (2020). On the design of backstepping controller for Acrobot system based on adaptive observer. *International Review of Electrical Engineering* (*IREE*), 15(4), 328.
42. Abdul-Adheem, W.R.; Azar, A.T.; Ibraheem, I.K.; and Humaidi, A.J. (2020). Novel active disturbance rejection control based on nested linear extended state observers. *Applied Sciences*, 10(12), 4069.

43. Al-Kaysi, A.M.; Ali, S.M.; Mahmood, N.S.; and Humaidi, A.J. (2024). Design of ECG monitoring system for cardiovascular patients using LabVIEW. *Journal of Engineering Science and Technology*, 19(4), 1189-1205.
44. Mahmood, R.M.; Ajel, A.R.; Abed, J.K.; Mahmod, R.A.; Al-Dujaili, A.; and Humaidi, A.J. (2023). LabVIEW-based design of smart wireless monitoring system for cardiac patients. *Proceedings of the* 2023 *IEEE* 13th *International Conference on System Engineering and Technology* (*ICSET*), Shah Alam, Malaysia, 46-49.
45. Humaidi, A.J.; Tala'at, E.N.; Hameed, M.R.; and Hameed, A.H. (2019). Design of adaptive observer-based backstepping control of cart-pole pendulum system. *Proceedings of the* 2019 *IEEE International Conference on Electrical*, *Computer and Communication Technologies* (*ICECCT*), Coimbatore, India.