Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6, 8582-8610
2024
Publisher: Learning Gate
DOI: 10.55214/25768484.v8i6.3848

Energy-efficient clustering in wireless sensor networks using metaheuristic algorithms

Kadhim Hayyawi Flayyih[1*], Mohsen Nickray[2]
[1,2]Department of Computer Engineering and Information Technology, University of Qom, Qom, Iran:
best.young2013@gmail.com (K.H.F.).

Abstract: Energy management in Wireless Sensor Networks (WSNs) remains a critical challenge, particularly in clustering processes. This article compares three optimization algorithms—Grasshopper Optimization Algorithm (GOA), Bat Algorithm (BA), and Whale Optimization Algorithm (WOA)—to achieve energy-efficient clustering and extend network lifetime. Initial cluster head placement is performed using K-means clustering, and a novel cost function is introduced that considers energy consumption and node distribution, enhancing the network's efficiency and resilience. The algorithms are evaluated across three scenarios with varying base station (BS) placements. In the simplest scenario, with the BS centrally located, GOA slightly outperforms WOA in extending network lifetime, although WOA remains competitive. BA, while energy-efficient, lags behind GOA and WOA. As complexity increases with BS placement at the edge, WOA demonstrates superior energy management, delaying node death and extending network lifetime more effectively than GOA and BA. In the most challenging scenario, where the BS is placed in a remote corner, WOA emerges as the most effective algorithm, maintaining network performance and balancing energy consumption for the longest duration. GOA, while relatively strong, shows faster network lifetime decline, particularly in later stages, whereas BA faces significant challenges, leading to quicker node failures. Overall, this study highlights the importance of efficient clustering and optimization for prolonging WSN lifetimes. WOA excels in complex scenarios, while GOA leads in simpler environments. Integrating K-means clustering with the novel cost function enhances algorithm performance, contributing to the development of resource-efficient WSNs, especially in resource-constrained settings.
Keywords: *Bat algorithm, Clustering, Whale optimization algorithm, Wireless sensor network.*

1. Introduction

In the era of rapidly advancing technology, Wireless Sensor Networks (WSNs) have become integral to various applications, from environmental monitoring to industrial automation. However, the energy limitations of sensor nodes remain a critical challenge, directly affecting network performance and lifetime [1]. Efficient energy management is essential, particularly in clustering processes, which play a pivotal role in optimizing data transmission and reducing energy consumption [2]. Traditional clustering methods often fall short in addressing the dynamic and resource-constrained nature of WSNs, necessitating the adoption of advanced optimization techniques.

Recent advancements in metaheuristic algorithms, such as Grasshopper Optimization Algorithm (GOA), Bat Algorithm (BA), and Whale Optimization Algorithm (WOA), have shown promise in solving multi-objective optimization problems [3]. These algorithms offer innovative approaches to energy-efficient clustering by balancing node energy consumption and improving network resilience. Despite their potential, a comparative analysis of these methods under diverse network conditions remains limited.

This Article addresses this gap by evalsuating GOA, BA, and WOA in energy-efficient clustering for WSNs. By integrating K-means clustering for initial cluster head placement and introducing a novel

History: Received: 08 November 2024; Revised: 16 November 2024; Accepted: 07 December 2024; Published: 23 December 2024
* Correspondence: best.young2013@gmail.com

cost function that considers energy distribution and node placement, the research aims to enhance the network's overall efficiency and lifetime. The findings contribute to developing sustainable WSNs, especially in resource-constrained environments.

Multiple studies have proposed methods to improve the energy efficiency of clustering in Wireless Sensor Networks (WSNs).

Subramani, N et al.in paper [3] focuses on the design of a clustering approach based on metaheuristics with a routing protocol for Underwater Wireless Sensor Networks (UWSN), named MCR-UWSN. The goal of the MCR-UWSN solution is to select an efficient set of Cluster Heads (CHs) and a path to the destination. The MCR-UWSN solution involves designing clustering strategies based on the Cultural Emperor Penguin Optimization (CEPOC) algorithm to form the clusters. Additionally, a multi-hop routing solution, combined with a Grasshopper Optimization Algorithm (MHR-GOA), is derived using multi-input components. The performance of MCR-UWSN was validated, and the results are evaluated based on various metrics. Experimental results highlighted the advanced performance of the MCR-UWSN approach in comparison to recent state-of-the-art methods.

Saadati, M et al. In article [4], the importance of wireless sensor networks (WSNs) and the challenges related to their energy limitations are discussed. Clustering and multi-hop routing methods are introduced as effective solutions for increasing WSN lifespan. This article proposes a method using Graph Neural Networks (GNN) to create static clusters of equal size, aiming to balance energy consumption among nodes. Additionally, a distributed cluster head selection scheme and a centralized routing protocol are implemented to establish dedicated routes to the base station, preventing node overheating. Simulation results show that this approach outperforms similar protocols in terms of lifespan and coverage.

Sulthana, N et al. In article [5], wireless sensor networks (WSNs) are highlighted for their broad applications and advantages, yet energy consumption remains a major challenge. Addressing this, the study proposes the Energy-Efficient Lifetime-Aware Cluster-Based Routing (EELCR) technique to improve energy efficiency in WSNs. The EELCR utilizes the modified Giant Trevally Optimization (MGTO) algorithm for balanced clustering to reduce energy consumption, while the Optimal Squirrel Search (OSS) algorithm is applied to select the best cluster head (CH), extending network lifespan. Each CH employs optimal selective Huffman compression to minimize area overhead. Additionally, a hybrid deep learning model combining Deep Neural Networks (DNN) and Granular Neural Networks (GNN), called DGNN, is used to optimize data transmission from CHs to the base station. Simulations show that EELCR improves the average compression rate by 9.346% and significantly extends network lifetime by 51.88% in node density scenarios and by 52.625% over simulation rounds compared to existing methods.

Debasis, K et al. In article [6], the Energy-Efficient Clustering Algorithm (EECA) is proposed to extend the lifetime of wireless sensor networks (WSNs) by reducing unnecessary energy consumption. Since sensor nodes rely on small batteries, idle listening can deplete energy quickly. The EECA model divides the target area into small regions, with an Artificial Neural Network (ANN) selecting one node in each region as the cluster head (CH). Only nodes with a minimum energy level participate in the CH selection, with ANN scoring candidates based on residual energy, detected events, distance to the base station, and neighbor count. The node with the highest score becomes the CH, and a limit on cluster size prevents overly large clusters. To further conserve energy, only nodes near an event transmit data to the CH, reducing redundant communication. Additionally, CHs briefly check for incoming signals at the beginning of each slot, turning off the radio if no transmission is detected, minimizing idle listening. Experimental results demonstrate that EECA achieves greater energy savings compared to other medium access control protocols.

Nedham, W et al. In article [7], the growing importance of Wireless Sensor Networks (WSNs) across various applications, including smart cities, the Internet of Things (IoT), and environmental monitoring, is discussed. Given WSNs' energy constraints, the study emphasizes the need for energy-aware protocols to sustain network performance. Hierarchical techniques, particularly clustering, are highlighted as effective strategies for improving network scalability, reducing latency, and enhancing energy efficiency. Clustering divides the network into sub-networks with Cluster Heads (CH) that manage communication within clusters, thus extending network lifespan. This study provides a comprehensive review of various clustering techniques, classifying and evaluating them based on cluster characteristics, CH attributes, and clustering methodologies, offering insights into their effectiveness for large-scale WSNs.

El Khediri, S. In paper [8], a comprehensive review of several clustering protocols proposed for Wireless Sensor Networks (WSNs) is presented. The clustering algorithms are classified into four categories: (1) Cluster-based protocols for homogeneous nodes, (2) Cluster-based protocols for heterogeneous nodes, (3) Clustering protocols based on fuzzy logic methods, and (4) Clustering protocols based on heuristic methods. This classification is based on the network organization of these protocols and the strategies used to manage clustering procedures. To evaluate the performance of these protocols, the features, performance, and clustering methods are used as the main parameters to compare the four categories of clustering approaches.

Surenther, I. et al. In article [9], a Deep Learning based Grouping Model Approach (DL-GMA) is proposed to address the energy constraints of Wireless Sensor Networks (WSNs), which limit network lifespan. DL-GMA utilizes deep learning, specifically Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM), to optimize energy efficiency in cluster formation, Cluster Head (CH) selection, and CH maintenance. Evaluated on metrics such as Energy Efficiency (88.7%), Network Stability (90.8%), Network Scalability (87.1%), Congestion Level (18.3%), and Quality of Service (QoS) (93.4%), DL-GMA significantly improves WSN energy utilization, network longevity, and data transmission efficiency. This model offers a promising solution to extend WSN lifespan and optimize network performance through intelligent grouping and deep learning.

Mittal, M et al. In paper [10], two energy-efficient protocols, namely the Adaptive Low-Energy Clustering Hierarchy and Energy-Efficient Sensor Routing, are redesigned with consideration of current application scenarios. Neural networks are integrated to improve the energy efficiency results, with a Levenberg-Marquardt Neural Network (LMNN) employed. Additionally, a sub-clustered protocol derived from LEACH is proposed for further enhancement. Simulation results show that Sub-LEACH with LMNN outperforms its competitors in terms of energy efficiency. Moreover, end-to-end delay is evaluated, with Sub-LEACH proving to be the best among the existing strategies.

Dinesh, K et al. In article [11], a trust-aware neuro-fuzzy-based clustering approach combined with the Sparrow Search Optimization Algorithm (NF-SSOA) is introduced to address energy efficiency and secure data transmission challenges in Wireless Sensor Networks (WSNs). Due to the resource limitations and vulnerability of WSNs, the NF-SSOA protocol aims to enhance energy optimization while ensuring secure communication. The protocol uses neuro-fuzzy clustering for effective node grouping and the sparrow search algorithm for optimized routing. Additionally, an ECC-based digital signature provides lightweight key management, encryption, and node authentication, with pseudo-random identity generation for anonymous data transmission. Implemented on the NS3 simulator, the protocol demonstrates improvements in energy consumption, throughput, network delay, network lifetime, and packet delivery ratio compared to existing protocols. The NF-SSOA protocol shows notable resilience against security threats and enhances the overall quality of service in WSNs.

Lilhore, U et al. In article [12], a depth-controlled energy-balanced routing protocol is proposed, which can adjust the depth of low-energy nodes and swap them with higher-energy nodes to ensure balanced energy usage. This energy-efficient routing protocol is based on an advanced Genetic Algorithm (GA) and data fusion techniques. The proposed protocol improves an existing GA by adding encryption, crossover, and mutation strategies, helping to specify nodes and optimize routing decisions. It uses an enhanced backpropagation neural network for data fusion operations, leveraging a highly optimized momentum strategy that aids in selecting only energy-efficient nodes, thus reducing redundant selections and minimizing data transmission energy. The protocol also incorporates an advanced cluster head node selection strategy capable of analyzing the remaining energy and directions of participating nodes. Simulation results show that the proposed model achieves 86.7% packet delivery ratio, 12.6% energy consumption, and a 10.5% packet loss ratio, outperforming depth-based and energy-efficient depth-based routing methods in underwater wireless sensor networks.

Wireless Sensor Networks (WSNs), with their extensive applications in fields such as the Internet of Things (IoT), smart agriculture, and environmental monitoring, have become a critical area of research in information and communication technology. However, the energy constraints of sensor nodes remain a fundamental challenge in extending network lifetime. Previous studies have demonstrated that clustering nodes and selecting appropriate cluster heads can optimize energy consumption and enhance communication efficiency. Nonetheless, existing methods face significant challenges.

Many traditional clustering algorithms, such as LEACH, while improving energy consumption, struggle to maintain optimal performance when the network scales or nodes are unevenly distributed. Metaheuristic algorithms, including Genetic Algorithms and Particle Swarm Optimization, have also been employed. However, these methods often exhibit limitations, such as a tendency toward local search or requiring complex configurations, particularly in multi-objective optimization. Furthermore, in multi-objective optimization approaches, achieving a balance between energy consumption and network lifetime is often inadequately addressed.

To overcome these limitations, this research proposes the use of two advanced metaheuristic algorithms—Whale Optimization Algorithm (WOA) and Bat Algorithm (BA)—for energy-efficient clustering in WSNs. The Whale Optimization Algorithm, inspired by the hunting behavior of whales, excels in global search and is well-suited for cluster formation optimization. Conversely, the Bat Algorithm, which mimics the echolocation behavior of bats, performs better in local search and is effective in fine-tuning clustering parameters.

The proposed method combines the advantages of these two algorithms to deliver optimized performance. It simultaneously optimizes criteria such as energy consumption, distance to the base station, and node workload, resulting in more stable clustering. Simulation results demonstrate that this approach significantly improves network lifetime and optimizes energy consumption compared to similar methods.

This method is not only applicable to WSNs with uneven node distributions but also performs effectively in scalable scenarios. By reducing computational overhead and extending network lifetime, the proposed approach presents a practical and efficient solution for real-world WSN applications.

The structure of this paper is organized as follows: Section 2 provides the basic concepts needed for understanding the proposed method. Section 3 introduces the proposed methodology, detailing the design and implementation of the Whale Optimization Algorithm (WOA) and Bat Algorithm (BA) for energy-efficient clustering. Section 4. Provides configuration and settings. Section 5 presents the results and analysis, including a comparison of WOA and BA in terms of energy efficiency, network lifespan, and cluster performance under different scenarios. Section 6 Provides a general discussion of the three scenarios performed.section 7 provides a comparative analysis between the proposed algorithms and existing approaches from the literature, highlighting the advantages and limitations of each method. Finally, Section8 concludes the paper by summarizing the findings and suggesting future research directions in the field of energy-efficient clustering in WSNs.

2. Basic Concepts

In this section, a detailed overview of the fundamental principles and key concepts required to understand the proposed method is presented.

2.1. Whale Optimization Algorithm (WOA)

The Whale Optimization Algorithm (WOA) is a metaheuristic algorithm inspired by the hunting behavior of humpback whales. These whales employ a unique hunting strategy known as bubble-net feeding, where they create spiral-shaped bubbles around their prey to encircle and move towards it. The WOA, introduced by Ali Mirjalili and colleagues in 2016, simulates this hunting strategy to search for optimal solutions in complex optimization problems [13].

The primary purpose of the Whale Optimization Algorithm is to find the best solution for complex, non-linear optimization problems where the objective function may have multiple peaks and valleys. In such cases, traditional methods struggle to locate global optima. WOA, like other metaheuristic algorithms, explores a wide solution space to approximate the best or nearly optimal solution. This algorithm is widely applied in fields like engineering optimization, artificial intelligence, neural network training, routing optimization, and energy management problems [14].

The WOA algorithm operates through three main mechanisms:

1. Encircling the prey: Whales encircle their prey and move toward it.
2. Bubble-net attacking: Whales move towards their prey in a spiral, simulating the bubble-net strategy.
3. Searching for prey: Whales explore the search space to find new hunting positions.

Each mechanism is explained in detail below, with relevant mathematical formulas.

2.1.1. Encircling the Prey

In their hunting, humpback whales first encircle their prey before moving towards it. In WOA, the best solution found so far is assumed to be the prey (or optimal solution), and other whales move toward this position. This encircling process is mathematically modeled as follows:

$$\vec{D} = |\vec{C}.\vec{X}^* - \vec{X}(t)| \qquad (1)$$

$$\vec{X}(t+1) = \vec{X}^* - \vec{A}.\vec{D} \qquad (2)$$

where:

- $\vec{X}^*$ is the position of the best solution (prey).
- $\vec{X}(t)$is the current position of the whale at iteration t.
- $\vec{D}$ represents the distance between the whale and the prey.
- $\vec{A}$ and $\vec{C}$ are coefficient vectors calculated as follows:

$$\vec{A} = 2.\vec{a}.\vec{r} - \vec{a} \qquad (3)$$

$$\vec{C} = 2.\vec{r} \qquad (4)$$

where:

$\vec{a}$ is linearly decreased from 2 to 0 over the course of iterations.

$\vec{r}$ is a random vector with values between 0 and 1.

2.1.2. Bubble-net Attacking Method (Exploitation Phase)

This phase simulates the bubble-net feeding behavior of humpback whales, where they spiral around their prey while gradually closing in. Two movement patterns are used, selected with a 50% probability: shrinking encircling mechanism and spiral updating position.

2.1.2.1. Shrinking Encircling Mechanism

In this approach, whales directly move towards their prey by reducing the distance, calculated with the encircling formulas mentioned above.

2.1.2.2. Spiral Updating Position

Whales move in a spiral path towards the prey. The mathematical representation of this spiral path is:

$$\vec{X}(t+1) = \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \vec{X}^* \qquad (5)$$

where:

$\overrightarrow{D'} = |\vec{X}^* - \vec{X}(t)\ |$ represents the distance between the whale and the prey.

b is a constant that defines the spiral shape.

l is a random number in the range [-1 , 1].

2.2. Search for Prey (Exploration Phase)

In this phase, whales are encouraged to explore the search space to discover new potential solutions. This is achieved by generating random positions in the search space when $|\vec{A} > 1|$ guiding whales towards random solutions. This feature helps the WOA avoid getting trapped in local optima and explore the search space more effectively.

The exploration phase is mathematically represented as:

$$\vec{D} = |\vec{C}.\vec{X}_{rand} - \vec{X}(t)| \qquad (6)$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A}.\vec{D} \qquad (7)$$

where:

$\vec{X}_{rand}$ is a randomly selected whale's position.

2.3. The General WOA Algorithm Steps are as Follows

1- Initialize: Set up the whale population, number of iterations, and parameters $\vec{a}$, $\vec{A}$ and $\vec{C}$.

2- Objective Function Evaluation: Evaluate the objective function for each whale.

3- Identify Best Whale: Select the whale with the best objective value as the current best solution.

4- Update Whale Positions:

If $|\vec{A}|<1|$: Execute encircling and bubble-net methods.

If $|\vec{A}|>1|$: Perform the exploration phase.

5- Update Parameter $\vec{a}$: Decrease $\vec{a}$ with each iteration for convergence.

6- Termination: If the stopping condition (e.g., maximum iterations) is met, stop and return the best solution.

2.4. Grasshopper Optimization Algorithm (GOA)

The Grasshopper Optimization Algorithm (GOA) is a metaheuristic optimization algorithm inspired by the social behavior and movement patterns of grasshoppers. This algorithm is designed to find optimal points in complex optimization problems by mimicking the collective movement and food-searching behavior of grasshoppers. GOA is particularly useful in solving nonlinear and complex optimization problems, as it combines two main behaviors—random movement and targeted movement—to explore the search space extensively and find optimal solutions [15].

In nature, grasshoppers live in large groups to survive and find food sources. Their group movement helps them discover better paths and resources. In GOA, this behavior is divided into two main parts: random movement and targeted movement [16].

Random Movement: This type of movement allows grasshoppers to explore the search space more broadly, finding new and different solutions. In the GOA algorithm, this movement helps prevent the algorithm from getting trapped in local optima.

Targeted Movement: In this type of movement, grasshoppers move toward food sources (optimal points). In the algorithm, this movement helps guide the search toward optimal areas and leads to convergence to the best solution.

GOA Algorithm Stages

The GOA algorithm consists of three main stages [17]:

1. **Initial Population Setup:** First, a population of grasshoppers is randomly created in the search space. Each grasshopper is considered a possible solution, represented by a feature vector.
2. **Fitness Calculation:** For each grasshopper in the population, the value of the objective function is calculated. This value determines the performance or success rate of the grasshopper in reaching the goal. The goal in the GOA algorithm is typically to minimize or maximize an objective function.
3. **Position Update of Grasshoppers:** To update the position of each grasshopper, the algorithm uses two factors:

- **Random Movement:** To maintain diversity and prevent getting stuck in local optima.
- **Targeted Movement:** To guide the grasshoppers toward desirable areas in the search space.

2.5. Grasshopper Movement Equation

To update the position of the grasshoppers, the following equations are used:

$$X_i^{k+1} = X_i^k + A.X_G^k - X_i^k + R.(X_i^k - X_j^k) \quad (8)$$

In this equation:

- X_i^k is the position of the i grasshopper at the k iteration.
- X_G^k is the best-known position at the k iteration.
- A is a random value that controls the grasshopper's targeted movement.
- R is a random value that determines the grasshopper's random movement.
- X_i^k is the position of another grasshopper considered as a neighbor.

2.6. Bat Algorithm

The Bat Algorithm (BA) is a metaheuristic optimization algorithm inspired by the echolocation behavior and social behavior of bats. The algorithm mimics the bats' ability to move in groups and search for food, aiming to find optimal solutions in complex optimization problems. BA is particularly useful in solving nonlinear and complex optimization problems and works by combining two main behaviors: random movement and targeted movement, which allows it to explore the search space extensively to find optimal solutions [18].

The Bat Algorithm utilizes two key features observed in bats' behavior for moving in their environment:

1. **Echolocation**: Bats use sound waves to locate themselves and their surroundings.
2. **Ability to vary sound intensity and frequency**: Bats adjust their sound intensity and frequency to navigate effectively and locate food.

The Bat Algorithm consists of three main stages:

1. **Initial Movement of Bats**: Initially, a population of bats is randomly distributed in the search space. Each bat represents a potential solution and is characterized by a vector of features.
2. **Calculate Fitness and Update Position**: The fitness of each bat is continuously updated based on the quality of the solution it represents. The fitness function is typically the objective function to be minimized or maximized
3. **Update Positions and Velocities**: The position of each bat is updated based on two factors:

Random Movement: To maintain diversity and avoid premature convergence.

Movement towards Best Positions: To guide the bats towards optimal regions of the search space.

In the Bat Algorithm, the position and velocity of each bat are updated continuously during iterations. The primary equations are as follows:

2.6.1. Velocity Update

$$v_i^{k+1} = v_i^k + \left(x_i^k - x_g^k\right).\alpha + \beta.\left(x_i^k - x_i^{best}\right) \quad (9)$$

Where:

v_i^{k+1}: Updated velocity of bat iii at iteration k+1.
v_i^k: Current velocity of bat i at iteration k.
x_i^k: Current position of bat i at iteration k.
x_g^k: Global best position at iteration k.
x_i^{best} Best known position of bat i.
α : Random factor controlling the movement.
β: Step size that controls frequency.

2.6.2. Position Update

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (10)$$

Where:

x_i^{k+1}: Updated position of bat i at iteration k+1.
x_i^k: Current position of bat i at iteration k.
v_i^{k+1}: Updated velocity of bat i.

2.6.3. Frequency and Intensity Update

$$f_i = f_{min} + (f_{max} - f_{min}).\lambda \quad (11)$$

$$A_i = A_0.(1 - e^{-\gamma.t}) \quad (12)$$

Where:

$f_i: Frequency\ of\ bat\ i$
$A_i: Intensity\ of\ bat\ i's\ movement.$
$f_{max}\ and\ f_{min}$: Minimum and maximum frequencies
λ: Random factor for adjusting frequency.
A_0: Initial intensity value.
γ: Damping factor for intensity decay.
t: Iteration step.

2.7. K-Means

The K-Means algorithm is an unsupervised learning method designed for clustering data into distinct groups or clusters by minimizing the variance within each cluster. This could be especially useful because it can offer a great initial answer by considering the node's dispersion in the study area. The primary objective is to partition nodes into k clusters, where each node is assigned to the cluster with the nearest centroid, which serves as the representative of the cluster [19].

ISSN: 2576-8484
Vol. 8, No. 6: 8582-8610, 2024
DOI: 10.55214/25768484.v8i6.3848

The algorithm begins with the initialization of k cluster centroids, which can be selected either randomly or through specific initialization techniques. K-Means is very sensitive to the initialization of the centroids, as different starting points can lead to different clustering outcomes. Hence, the k-means++ algorithm is used for initialization of cluster centroids in this study.

The k-means++ algorithm uses a heuristic to find cluster centroids for k-means clustering. According to [54], k-means++ improves the running time of Lloyd's algorithm, and the quality of the final solution. The k-means++ algorithm chooses cluster centroids as follows:

1. Select an observation uniformly at random from all nodes. The chosen observation is the first cluster centroid and is denoted c_1.
2. Compute distances from each observation to c_1. Denote the distance between c_j and the observation m as $d(x_m,c_j)$.
3. Select the next centroid, c_2 at random from all nodes with the probability defined in equation (13).

$$\frac{d^2\left(x_m,c_1\right)}{\sum_{j=1}^{n_N} d^2\left(x_j,c_1\right)} \qquad (13)$$

4. To choose center j:

a. Compute the distances from each observation to each centroid, and assign each observation to its closest centroid.

b. For $m = 1, \ldots, n_N$ and $p = 1, \ldots, j-1$, select centroid j at random from all nodes with the following probability:

$$\frac{d^2\left(x_m,c_p\right)}{\sum_{\{h;x_h\in C_p\}} d^2\left(x_h,c_p\right)} \qquad (14)$$

where C_p is the set of all observations closest to centroid c_p and x_m belongs to C_p.

In other words, choose each next center based on a probability that is proportional to its distance from the nearest center you have already selected.

5. Repeat step 4 until K cluster centroids are chosen.

Following initialization, the algorithm enters the assignment phase, where each node x_i is assigned to the nearest centroid. This is determined using the Euclidean distance between the node and the centroid, calculated as:

$$d(x_i,\mu_j) = \sqrt{\sum_{m=1}^{M}(x_{im}-\mu_{jm})^2} \qquad (15)$$

where x_i represents the nodes' location in the study area, μ_j denotes the cluster centroid, and M is the dimensionality of the data. The data point x_i is assigned to the cluster C_j for which the distance $d(x_i, \mu_j)$ is the smallest.

After the assignment phase, the centroids are updated by calculating the mean of the data points within each cluster. The new centroid μ_j of cluster C_j is computed as:

$$\mu_j = \frac{1}{J_j}\sum_{x_i\in C_j} x_i \qquad (16)$$

where C_j is the set of nodes assigned to cluster j, and J_j is the number of nodes in that cluster.

The algorithm iteratively alternates between the assignment and update steps until convergence. Convergence occurs when the centroids no longer change, or when 100 iterations are reached. The K-Means algorithm seeks to minimize the within-cluster sum of squares (WCSS), also known as inertia. This function is defined as:

$$J = \sum_{j=1}^{K} \sum_{x_i \in C_j} \| x_i - \mu_j \|^2 \qquad (17)$$

This function represents the sum of squared distances between each node and its assigned centroid, summed over all clusters. The algorithm's goal is to minimize this value, thereby ensuring that nodes are as close as possible to their respective centroids.

3. Methodology

In this paper, the proposed approach is developed to optimize energy consumption and extend the lifespan of Wireless Sensor Networks (WSNs). This hybrid method integrates the K-means clustering algorithm with three metaheuristic algorithms: the Grasshopper Optimization Algorithm (GOA), the Bat Algorithm (BA), and the Whale Optimization Algorithm (WOA). This combination aims to exploit the advantages of each method and address their limitations, thereby optimizing the clustering process and the selection of cluster heads. The proposed approach includes the following steps: energy modeling, cost function design, initialization using the K-means algorithm, and the application of three optimization algorithms to solve the clustering problem.

First, the network energy model is designed. In WSNs, the transmitter-receiver unit of sensor nodes is typically the primary energy consumer. To better understand energy consumption, the transmitter-receiver is divided into three components: the power amplifier of the transmitter, the transmitter electronics, and the receiver electronics. Additionally, free-space channel models with power loss $d^2$and multi-path channel models with power loss d^4 are used to describe energy dissipation during transmission. As a result, the energy required for transmission (Etr) and reception (Ere)can be represented by the following equations:

$$E_{tr}{}^{i,r} = p_{tr} . \begin{cases} e_{elec} + e_{fs}.d^2 & if\ d \le d_0 \\ e_{elec} + e_{mp}.d^4 & if\ d > d_0 \end{cases} \qquad (18)$$

$$E_{re} = p_{re} \cdot e_{elec} \qquad (19)$$

Where p_{tr} and P_{re} represent the sizes of the transmitted and received packets, respectively. e_{elec} denotes the energy consumed per bit by the radio electronics of the transmitter and receiver. The parameters e_{fs} and e_{mp} are related to the characteristics of the power amplifier, while d0 is the threshold distance, defined by the following equation:

$$d_0 = \sqrt{\frac{e_{fs}}{e_{mp}}} \qquad (20)$$

Additionally, if data aggregation is employed (e.g., by cluster heads that collect data from connected nodes), the cluster head incurs an energy cost for processing one bit of data. Therefore, the total energy cost of aggregation can be expressed by the following equation:

$$E_{da} = p_{da} \cdot e_{agr} \qquad (21)$$

Where p_{da} represents the size of the data packets that need to be aggregated. By integrating these energy equations, the total energy consumption (energy dissipation) can be determined as follows:

$$E_l = E_{tr} + E_{re} + E_{da} \qquad (22)$$

Then, the clustering objectives in wireless sensor networks (WSN) are defined as an optimization problem and then the proposed cost function to be solved using all three methods presented in this study is introduced. It is assumed that the maximum number of cluster heads is predefined.Hence, K decision variables with the lower and upper bounds equal to (0, nN] are considered in the proposed method, where nN is the number of nodes and K is the maximum number of the cluster heads. These decision variables are first converted to integers using the ceiling function. Next, the repetitive values are discarded and a unique vector with a maximum length of K is obtained. The vector is the IDs of cluster heads that are selected in one scenario evaluation (a population member in an iteration). Moreover, since the repetitive values are discarded, the length of the vector can be varied from 1 to K. This makes the proposed method flexible in finding the optimal number of cluster heads. Once the cluster head IDs have been recognized, the nodes of each cluster can be determined by checking the distances between each node and cluster head. In other words, the nodes belong to the clusters where their cluster head is closer to them. After determining the cluster heads and links, the scenario should be evaluated. In this regard, the cost function is designed carefully by focusing on both energy consumption and energy distribution. Our first goal is to reduce the overall energy loss of the nodes. This is done by calculating the summation of energy loss of all nodes in the network. In a WSN, there are two categories of sensors. One is nodes that send data to cluster heads and do not receive any data. The other is cluster heads which send data to the base station and receive data from nodes of that cluster. For the normal nodes, the pre and pda are zero since they don't receive any data while the ptr is equal to the packet size of a single node which is considered to be p0 for all nodes in this study. The cluster heads, however, consume both receiving and aggregating energies. Considering there are Jk nodes in the kth cluster, the pre and pda are determined as Jk. p0 and (Jk +1). p0, respectively. This is because there are Jk packages that are received in the kth cluster head but there are Jk +1 packages to be aggregated by counting the package size of the kth cluster head itself. Furthermore, the ptr of cluster heads can be calculated as (1+ρ.Jk). p0 where ρ is the data aggregation ratio. By calculating the energy consumption of each node, the first part of the cost function which is related to the energy consumption is calculated as follows:

$$F1=\sum_{k=1}^{K}\left(\sum_{j=1}^{J_k}E_{l,CM}^{(k,j,r)}\right)+\sum_{k=1}^{K}E_{l,CH}^{(k,r)} \qquad (23)$$

where $E_{l,CM}^{(k,j,r)}$ is the energy loss for the j^{th} cluster member of the k^{th} cluster in the r^{th} round and $E_{l,CH}^{(k,r)}$ is the energy loss for the k^{th} cluster head in the r^{th} round

By only focusing on the energy loss of the nodes, we are more likely to postpone the round of the first node death (FND). However, the network might lose its energy fast after the FND since it doesn't forecast its situation in the next several rounds. This mostly happens when the nodes with critical positions die since the network had been relying on them to reduce the energy consumption in each round before the FND. Hence, we should keep energy distributed to prevent future undesirable consequences that arise from our decisions in each round. Most papers don't consider this critical factor and some of them try to keep energy the same over the network or over the cluster [20]. However, neither of these approaches consider the best distribution of energy in the network. According to equation (23), the consumed energy of each node in each round is correlated with the square or the fourth power of its distance from the destination (dependent on the distance). Thus, the far nodes should keep more energy since they need it in the last rounds of the network lifetime. In this study, the predicted node energy of the next round based on the present configuration and links are considered for the evaluation of energy distribution. Generally, the predicted energy can be calculated as follows:

$$E_{pre}^{(i,r+1)}=E_{rem}^{(i,r)}-E_{l}^{(i,r)} \qquad (24)$$

Where $E_{pre}^{(i,r+1)}$ is the predicted energy of the i^{th} node in the $(r+1)^{th}$ round, $E_{rem}^{(i,r)}$ is the remained energy of the i^{th} node in the r^{th} round, and $E_{l}^{(i,r)}$ is the lost energy (consumed energy) of the i^{th} node in the r^{th} round.

ISSN: 2576-8484
Vol. 8, No. 6: 8582-8610, 2024
DOI: 10.55214/25768484.v8i6.3848

As mentioned earlier the consumed energy is correlated with the square of distance for near destinations and the fourth power of the distance for far destinations. Hence, we should consider one of these conditions as the final pattern of the remaining energy distribution. In this study, the square of distance is considered for the remaining energy distribution pattern. This decision is made based on a simple idea: to have less energy difference between sensors and increase their flexibility.

Therefore, the ideal predicted energy of the next round would be as follows:

$$E_{pre}^{(i,r+1)} = \alpha^{(r)} \cdot d^{(i,b)^2} \quad (25)$$

Where $\alpha^{(r)}$ is a fixed coefficient for all nodes in the r^{th} round and $d^{(i,b)}$ is the distance between the i^{th} node and the base station. Figure 1 shows the ideal energy distribution based on equation (26). However, keeping all nodes in a specific distribution in all rounds is practically impossible. Thus, the second part of the cost function to be minimized is described as:

$$F2 = \left| E_{pre}^{(i,r+1)} - \alpha^{(r)} \cdot d^{(i,b)^2} \right| \quad (26)$$

Where $\alpha^{(r)}$ can be obtained using equation (26) as follows:

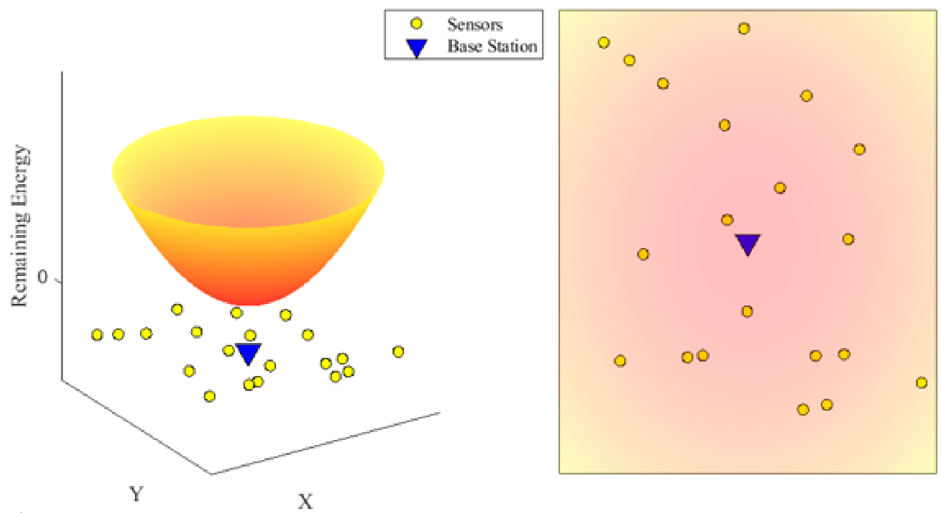


Figure 1.
The energy distribution which is trying to be preserved using the proposed cost function.

$$\alpha^{(r)} = \frac{\sum_i E_{pre}^{(i,r+1)}}{\sum_i d^{(i,b)^2}} \quad (27)$$

Finally, by defining both parts of cost functions, the final optimization problem can be obtained as the weighted sum of them. However, it should be noted that these cost functions have different units, and adjusting weight is critical to letting both parts have their impact. Hence, the final cost function is described as follows:

$$\min\,[F1 + \beta . F2] \quad (28)$$

where β is the trade-off coefficient for the final cost function.

When the optimization problem is completely formulated and the optimization algorithm and its decision variables are determined, we can solve the problem of WSN clustering. However, the crucial point to be noted is that the computational resources and time are limited in each round. Thus, we should solve the problem in the most efficient way. To make the problem more efficient we need to

DOI: 10.55214/25768484.v8i6.3848

restrict the population size or the maximum number of iterations in the optimization problem which might lead to achieving a suboptimal solution and reducing the overall performance. Therefore, another solution is proposed in this study to address this issue. In this regard, the K-means clustering algorithm is adopted to initially specify the location of cluster heads. K-means can propose a semi-optimal solution since the cluster heads would be properly distributed and the distances from cluster members are considered to be minimal. By initializing the population, we can start from a semi-optimized solution, preventing the optimization problem from investigating the odd and worthless solutions. This leads to a fast convergence and allows us to reduce the computational-related hyperparameters of the optimization algorithm without being concerned about reducing overall performance.

The K-Means algorithm is an unsupervised learning method designed for clustering data into distinct groups or clusters by minimizing the variance within each cluster. This could be especially useful because it can offer a great initial answer by considering the node's dispersion in the study area. The primary objective is to partition nodes into k clusters, where each node is assigned to the cluster with the nearest centroid, which serves as the representative of the cluster.

The time complexity of K-Means is $O(n_N \cdot K \cdot d \cdot I)$, where n_N is the number of nodes, K is the number of clusters, d is the dimensionality of the data, and I is the number of iterations. While K-Means is efficient for moderate-sized datasets, it can become computationally expensive for large datasets or high-dimensional data. Since the data dimension is 2, the number of iterations is 100, and the number of nodes and the number of clusters is limited, it can be computationally efficient for the initialization of the optimization algorithm.

In the first method, the Grasshopper Optimization Algorithm (GOA) is used to solve the problem. This algorithm simulates the swarming behavior of grasshoppers to guide the processes of exploration and exploitation. The positions of the grasshoppers are updated by combining social forces, gravity, and wind. By adjusting the control parameters, the algorithm achieves a suitable balance between exploration and exploitation, preventing it from getting trapped in local optima.

The second method employs the Bat Algorithm (BA), which is inspired by the echolocation behavior of bats. In this algorithm, frequency, velocity, and cluster head indices are dynamically updated. A local search mechanism is also utilized to enhance accuracy and exploit the best solutions

The third method is based on the Whale Optimization Algorithm (WOA), which simulates the social behavior of humpback whales. This algorithm leverages three main mechanisms: prey encirclement, spiral movement, and prey search. By adjusting control parameters and combining exploration and exploitation phases, WOA demonstrates excellent performance in optimizing the clustering process.

Finally, all three methods are evaluated in a uniform simulation environment, and their results are compared in terms of energy consumption, energy balance, and network lifetime.

4.1. Configuration and Setting

In this section, the configurations and adjustments that we made which are needed for simulating the proposed method are presented. In this regard, the WSN focuses on a $200m \times 200m^2$ area with 100 nodes that are randomly deployed in the sensing area. The maximum number of cluster heads (K) is considered equal to 10 which can be a rational choice based on the existence of a total of 100 nodes in the network. Furthermore, to model the energy consumption, the initial energy of each node is considered equal to 1J, the data packet size is equal to 500 bytes, the data aggregation ratio is considered 0.01, and other adjustments are reported in Table 1. Moreover, some other configurations are related to the proposed method. Beta is one important coefficient that makes a trade-off between two parts of the cost function. In this study, a β equal to 0.0008 is considered which gives a little more weight to the energy consumption part (F1). There are also some other parameters related to WOA, GOA, and BA, the detailed configuration of which is reported in Table 1.

ISSN: 2576-8484
Vol. 8, No. 6: 8582-8610, 2024
DOI: 10.55214/25768484.v8i6.3848

Table 1.
The parameters' setting used in this study.

Parameter	**Value**
WSN area	200m×200m
Number of nodes	100
Maximum number of cluster heads	10
Initial energy	1J
Data aggregation ratio	0.01
Data packet size	500 bytes
e_{elec}	50 nJ/but
e_{fs}	10 pJ/bit/m^2
e_{mp}	0.0013 pJ/bit/m^4
e_{da}	10 nJ/bit/signal
Population size of GOA	30
Maximum iterations of GOA	50
Population size of WOA	30
Maximum iterations of WOA	50
Population size of BA	30
Maximum iterations of BA	50
α	0.00125
β	0.5

4.2. Simulation Results

In this section, the obtained results from a comprehensive evaluation of all three methods proposed in this study are presented. To evaluate the performance comprehensively, we consider three scenarios. Each scenario has a different sensor and base station deployment as follows:

- Scenario 1: The BS is in the middle of the interesting area (the coordinate of (100,100)).
- Scenario 2: The BS is located at a coordinate of (0,95) in the interesting area.
- Scenario 3: The BS is located at the coordinate of (170, 180) in the interesting area.

The performance of each method is evaluated in all three scenarios and will be compared to each other. The obtained results for each scenario are reported in the rest of this section.

4.3. Scenario 1: Base Station Located at the Center of the Area

In this scenario, the BS is centrally located, providing equal opportunity for all nodes to communicate with the BS. This balanced positioning reduces the energy consumption needed for communication, making it the least challenging scenario. Figure 2 illustrates the lifetime curves of all three methods (WOA, GOA, and BA). To compare the obtained results, the performance of the all methods is evaluated based on first node dying (FND), half node dying (HND), and last node dying (LND).

ISSN: 2576-8484
Vol. 8, No. 6: 8582-8610, 2024
DOI: 10.55214/25768484.v8i6.3848

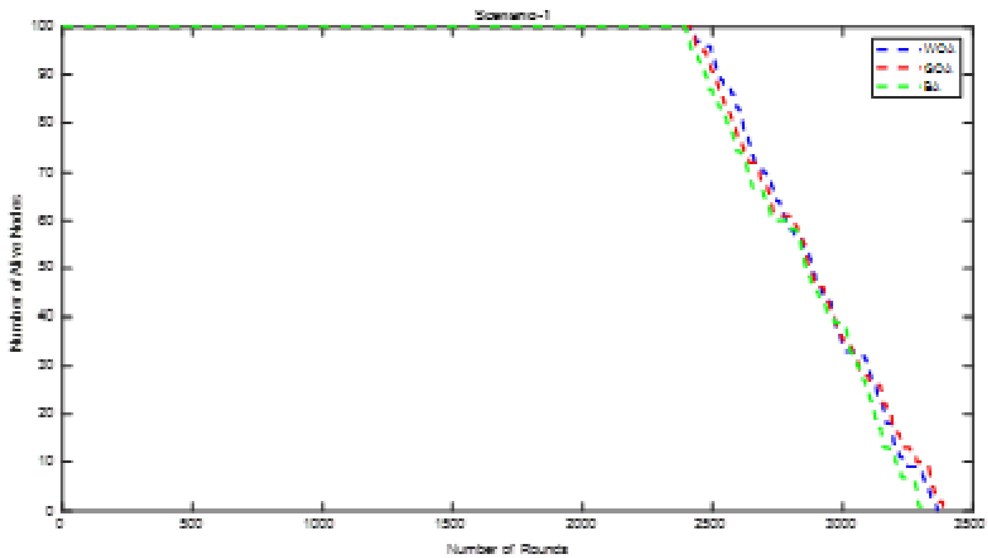


Figure 2.
The lifetime curves of WOA, GOA, and BA using the proposed method in the first scenario.

As can be seen, WOA exhibits a robust performance, with the FND recorded at 2415 rounds. This reflects a balanced energy distribution strategy that delays the first node's failure. WOA reaches HND at 2882 rounds, showing that half of the network's nodes remain active for a considerable time after the first node's failure. The LND occurs at 3364 rounds, indicating a gradual decline in network functionality as nodes deplete their energy. WOA's relatively long network lifespan demonstrates effective energy management across all nodes. GOA shows slight differences compared to WOA. The FND is marginally later at 2417 rounds, indicating a very similar initial energy distribution as WOA. However, the HND occurs earlier than WOA at 2870 rounds, suggesting that nodes begin failing more rapidly during the middle stages of the network's operation. Despite this, GOA outperforms WOA in terms of overall network longevity, with the LND occurring at 3402 rounds, which is 38 rounds longer than WOA. This difference highlights GOA's ability to conserve energy in the latter stages of the network's life, resulting in a more prolonged overall lifespan.

BA displays the earliest FND at 2399 rounds, showing a slight difference in energy distribution compared to WOA and GOA. BA's HND occurs at 2858 rounds, indicating a quicker depletion of energy across half the network. Finally, the LND is recorded at 3308 rounds, making BA the algorithm with the shortest overall network lifespan in this scenario. The difference in performance between BA and the other algorithms suggests that BA is less efficient in conserving energy, leading to earlier network failures. However, the differences are relatively small, with BA trailing behind WOA by 56 rounds and GOA by 94 rounds in terms of LND.

Overall, GOA emerges as the best-performing algorithm in this scenario, with a longer overall network lifespan compared to WOA and BA. The difference in LND between GOA and WOA (38 rounds) suggests that GOA is more effective in extending network life, particularly during the later stages. WOA maintains a strong performance, with only slight differences compared to GOA, particularly in HND, where WOA outlasts GOA by 12 rounds. BA, while still competitive, shows a more rapid decline in node life, with the differences in FND, HND, and LND pointing to slightly less efficient energy management. Despite these differences, all three algorithms perform well in this least challenging scenario, with GOA showing the most pronounced advantage in overall network longevity. Table 2 summarizes the findings of this section. Moreover, Figures 3, 4, and 5 illustrate the structure of WSN in the first four rounds using WOA, GOA, and BA, respectively.

ISSN: 2576-8484
Vol. 8, No. 6: 8582-8610, 2024
DOI: 10.55214/25768484.v8i6.3848

Table 2.
The obtained results of the first scenario.

	FND	HND	LND
WOA	2415	2882	3364
GOA	2417	2870	3402
BA	2399	2858	3308

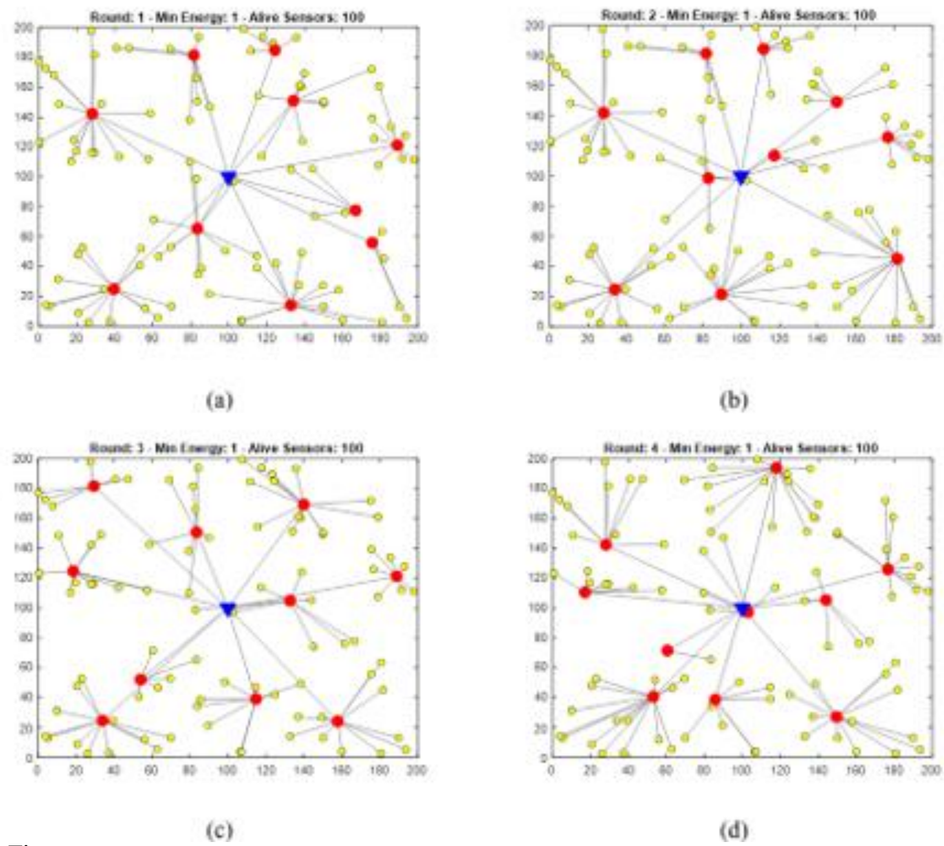
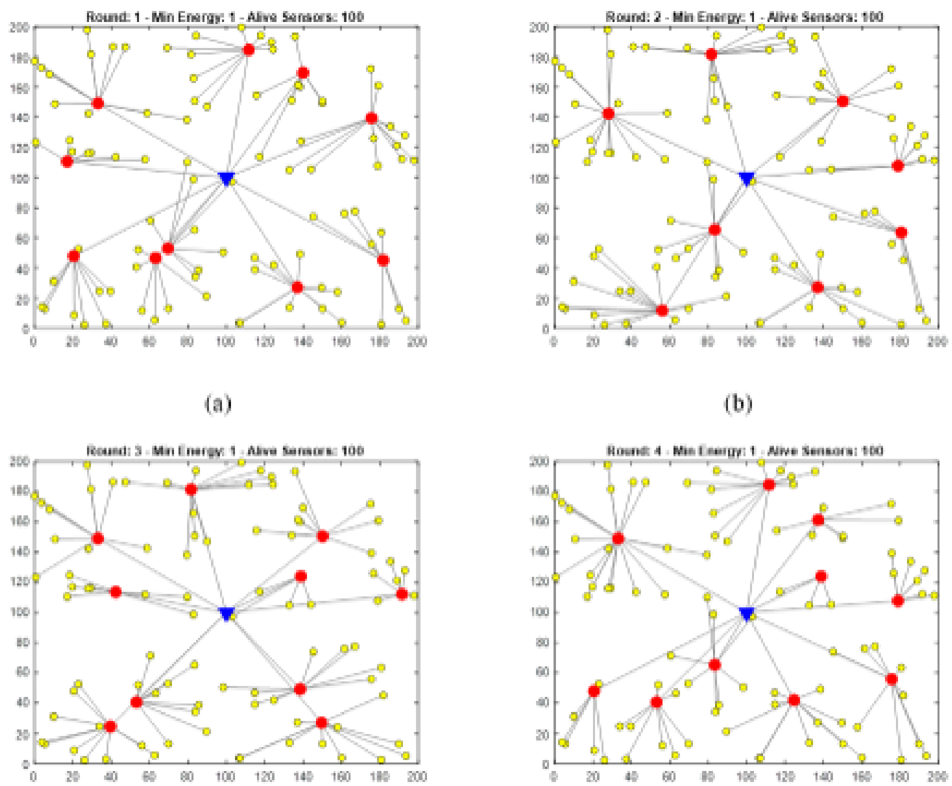


Figure 3.
The structure of the WSN in (a) round 1, (b) round 2, (c) round 3, and (d) round 4 of solving the proposed cost function using the WOA algorithm and K-means in the first scenario.

DOI: 10.55214/25768484.v8i6.3848



(a) (b)

(c) (d)

Figure 4.
The structure of the WSN in (a) round 1, (b) round 2, (c) round 3, and (d) round 4 of solving the proposed cost function using the GOA algorithm and K-means in the first scenario.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 8582-8610, 2024
DOI: 10.55214/25768484.v8i6.3848

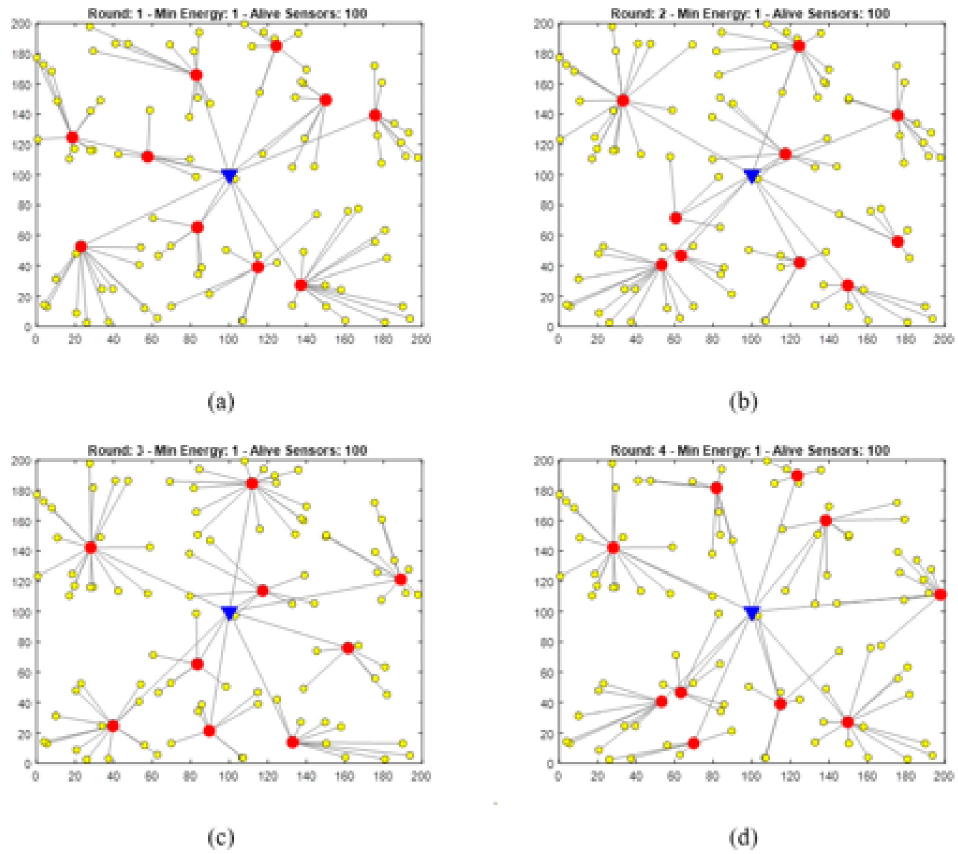


Figure 5.
The structure of the WSN in (a) round 1, (b) round 2, (c) round 3, and (d) round 4 of solving the proposed cost function using the BA algorithm and K-means in the first scenario.

4.4. Scenario 2: Base Station Located at the Edge of the Area

In this scenario, the BS is positioned at the edge of the area of interest, specifically at the coordinate (0,95). This location creates a more challenging environment for the network, as nodes farther from the BS require more energy to communicate, potentially leading to faster energy depletion. Figure 6 illustrates the lifetime curves of all three methods (WOA, GOA, and BA) in this scenario.

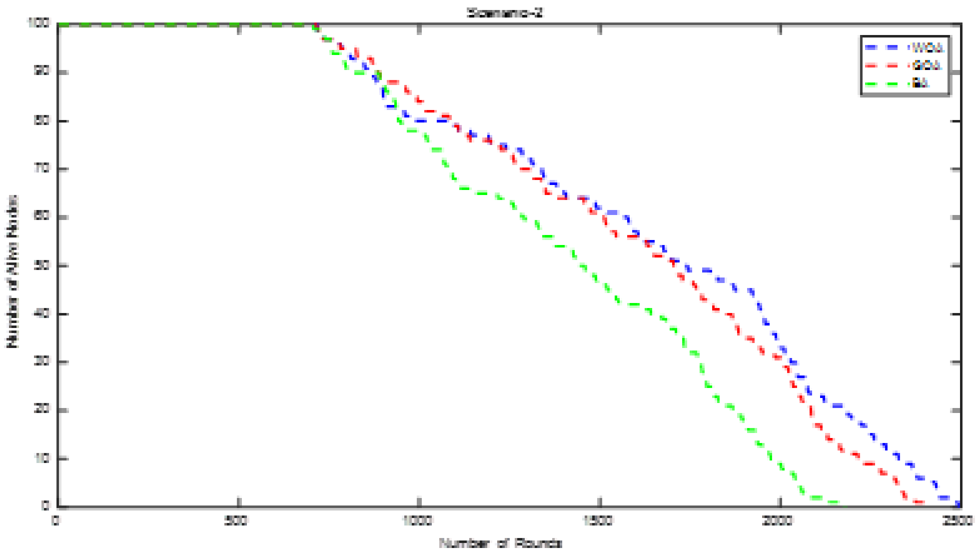


Figure 6.
The lifetime curves of WOA, GOA, and BA using the proposed method in the second scenario.

As it can be observed, WOA shows strong performance in this scenario, with the FND occurring at 718 rounds. This indicates that WOA effectively distributes energy across the network, even in this more demanding setting. The HND is recorded at 1729 rounds, meaning half of the network nodes continue functioning for a significant period after the first node's failure. Finally, the LND occurs at 2492 rounds, demonstrating that WOA sustains the network for an extended time, even under the more challenging conditions of this scenario. GOA exhibits a performance similar to WOA but with slight differences. The FND is recorded at 716 rounds, almost identical to WOA. However, the HND occurs earlier at 1704 rounds, indicating that the network begins to lose nodes more quickly during the middle stages compared to WOA. GOA's LND is observed at 2410 rounds, which is 82 rounds shorter than WOA, suggesting that GOA is somewhat less effective at managing energy in the later stages of the network's operation. Nonetheless, GOA still performs well, maintaining network functionality for a substantial period. Finally, BA shows a more pronounced difference in performance compared to WOA and GOA according to Figure 4. The FND occurs earlier at 707 rounds, indicating that nodes begin to fail slightly sooner. BA's HND is recorded at 1440 rounds, a noticeable decline compared to both WOA and GOA, suggesting a quicker energy depletion across the network. The LND occurs at 2178 rounds, marking the shortest network lifespan among the three algorithms in this scenario. The differences in BA's performance suggest that it is less efficient at managing energy, especially in challenging scenarios like this one, where nodes farther from the BS consume more energy. In summary, WOA stands out as the top-performing algorithm in this more challenging scenario, achieving the LND of 2492 rounds. The difference of 82 rounds between WOA and GOA indicates that WOA is slightly more effective at extending the network's life, particularly during the later stages. GOA, while close to WOA in performance, shows a quicker decline in HND and LND, reflecting a slightly less efficient energy management strategy. BA, with the earliest FND and shortest LND, demonstrates the most rapid energy depletion, making it less suitable for scenarios where nodes are farther from the BS. Despite these differences, all three algorithms show their strengths in different aspects, with WOA providing the best overall performance in this scenario. Table 3 shows the obtained FND, HND, and LND of each method. Moreover, Figures 7, 8, and 9 depict the WSN in the first four rounds of this scenario using WOA, GOA, and BA, respectively.

Table 3.
The obtained results of the second scenario.

	FND	HND	LND
WOA	683	1388	2022
GOA	637	1386	1917
BA	394	1242	1767

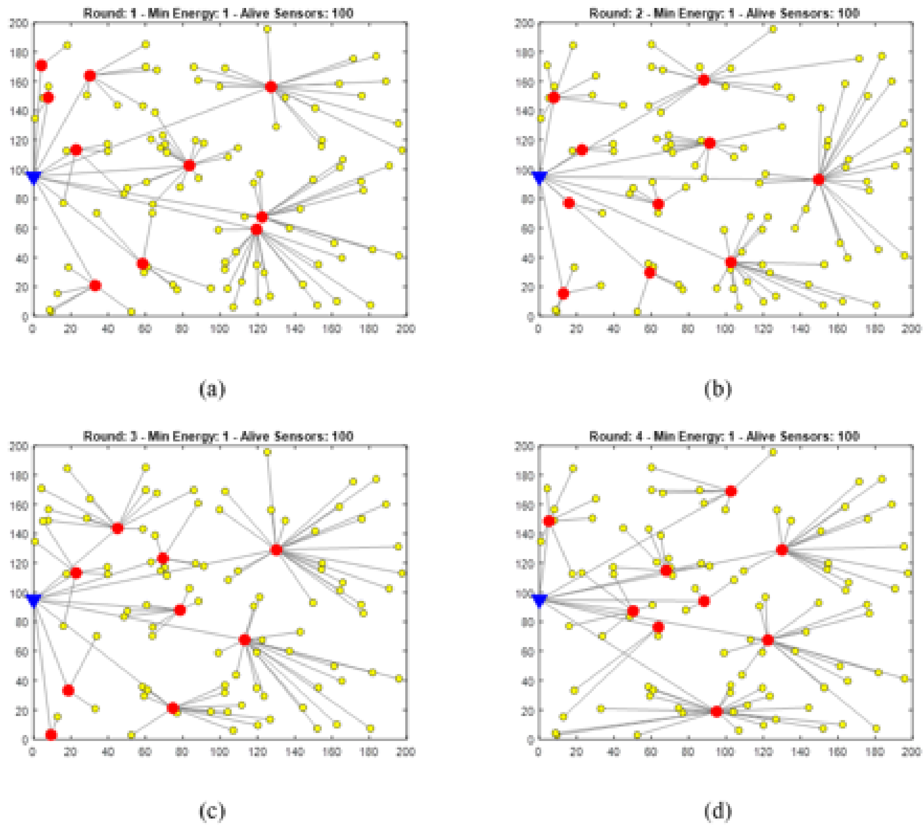


Figure 7.
The structure of the WSN in (a) round 1, (b) round 2, (c) round 3, and (d) round 4 of solving the proposed cost function using the WOA algorithm and K-means in the second scenario.

ISSN: 2576-8484
Vol. 8, No. 6: 8582-8610, 2024
DOI: 10.55214/25768484.v8i6.3848

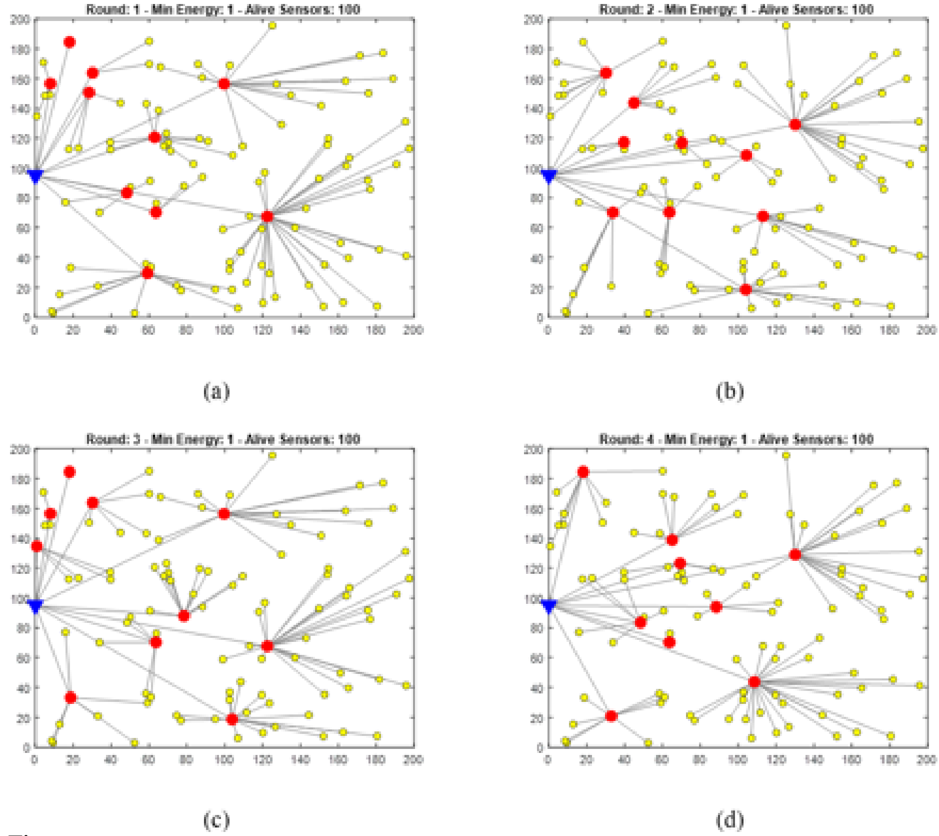


Figure 8.
The structure of the WSN in (a) round 1, (b) round 2, (c) round 3, and (d) round 4 of solving the proposed cost function using the GOA algorithm and K-means in the second scenario.

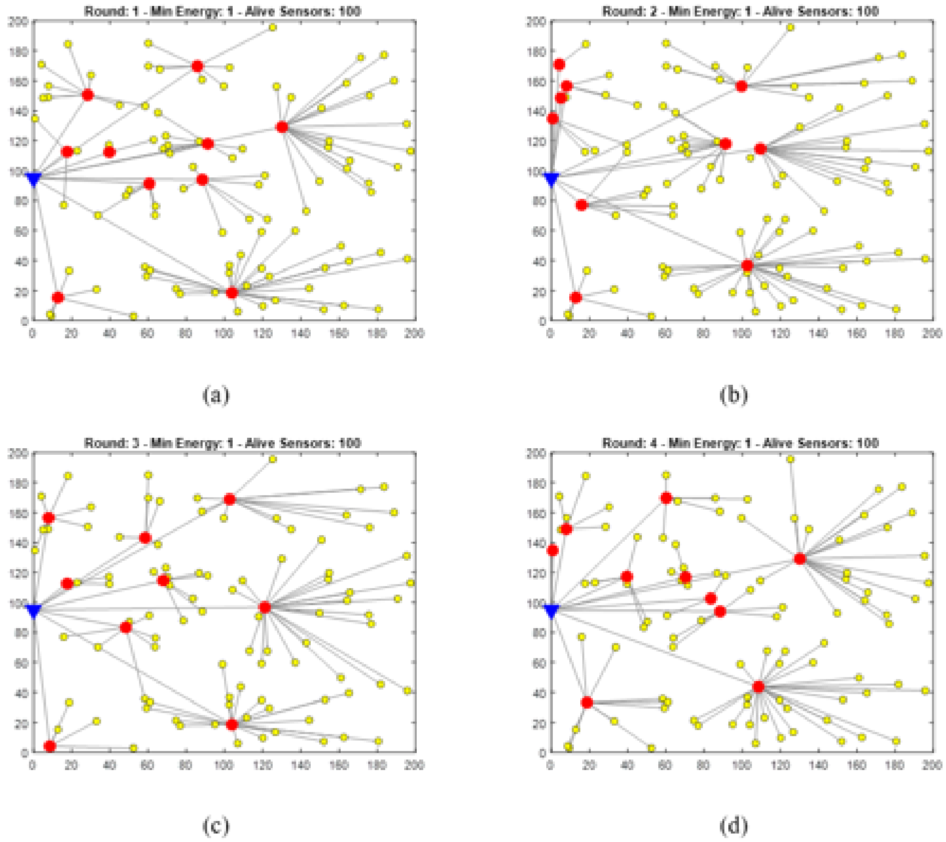


Figure 9.
The structure of the WSN in (a) round 1, (b) round 2, (c) round 3, and (d) round 4 of solving the proposed cost function using the BA algorithm and K-means in the second scenario.

4.5. Scenario 3: Base Station Located at a Distant Corner of the Area

In this most challenging scenario, the base station (BS) is located at a distant corner of the area, at the coordinates (170,180). This positioning creates significant challenges for nodes that are far from the BS, as they require more energy for communication, leading to faster energy depletion. Figure 10 shows the lifetime curves of all three methods in this scenario.

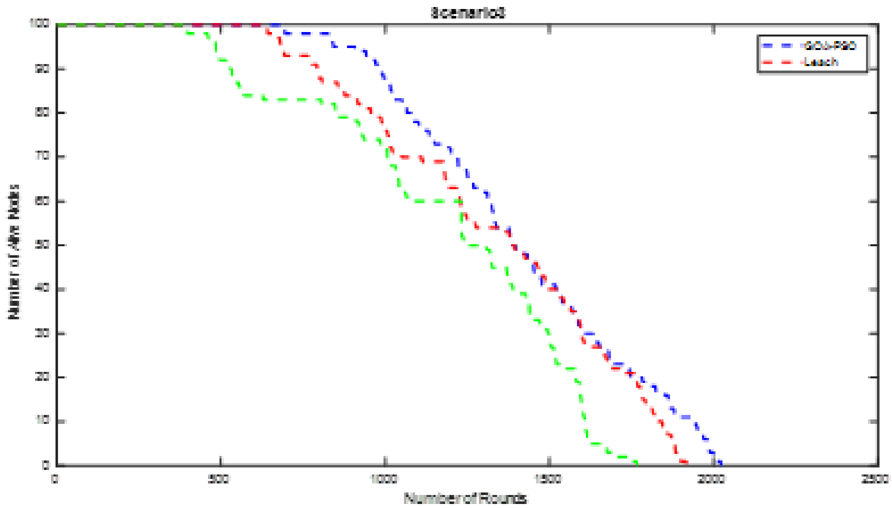


Figure 10.
The lifetime curves of WOA, GOA, and BA using the proposed method in the third scenario.

According to the obtained results, WOA shows resilience even in this demanding scenario, with the FND occurring at 683 rounds. This indicates that WOA manages to delay the first node's failure despite the increased energy demands. The HND is recorded at 1388 rounds, meaning that half of the network's nodes remain operational for a significant duration after the first failure. The LND occurs at 2022 rounds, demonstrating that WOA effectively sustains the network for an extended period, even under the most challenging conditions. WOA's performance here highlights its ability to balance energy consumption across the network, leading to the longest network lifespan in this scenario. Moreover, GOA also performs well in this scenario, but with some differences compared to WOA. The FND is observed at 637 rounds, earlier than WOA, indicating a quicker initial energy depletion. However, GOA closely follows WOA in terms of HND, with the half node dying at 1386 rounds—only 2 rounds earlier than WOA. The LND occurs at 1917 rounds, which is 105 rounds shorter than WOA. This difference suggests that while GOA effectively manages energy during the middle stages, it is slightly less efficient at conserving energy in the later stages of the network's operation compared to WOA.

BA exhibits the most rapid decline in performance in this scenario. The FND occurs significantly earlier at 394 rounds, indicating that nodes start failing much sooner than with WOA and GOA. BA's HND is recorded at 1242 rounds, a noticeable decline compared to the other algorithms, showing that half of the network's nodes deplete their energy relatively quickly. Finally, the LND is observed at 1767 rounds, marking the shortest network lifespan among the three algorithms in this scenario. The differences in BA's performance suggest that it struggles to manage energy efficiently in such a challenging environment, leading to earlier node failures and a quicker overall network decline.

In a nutshell, WOA stands out as the best-performing algorithm in this most challenging scenario, with the longest network lifespan (LND) at 2022 rounds. The difference of 105 rounds between WOA and GOA underscores WOA's superior ability to extend the network's life, particularly during the later stages of operation. GOA, while performing strongly in the early and middle stages, shows a quicker decline towards the end, reflecting a slightly less effective energy management strategy compared to WOA. BA, with the earliest FND and shortest LND, demonstrates a much more rapid energy depletion, making it less suitable for highly demanding scenarios like this one. Despite the challenging conditions, all three algorithms demonstrate their capabilities, with WOA emerging as the most resilient in sustaining the network over time. Table 4 illustrates the summary of the mentioned analysis. Furthermore, Figures 11, 12, and 13 show the WSN structure in the first four rounds of the third scenario using WOA, GOA, and BA, respectively.

Table 4.
The obtained results of the third scenario.

	FND	HND	LND
WOA	718	1729	2492
GOA	716	1704	2410
BA	707	1440	2178

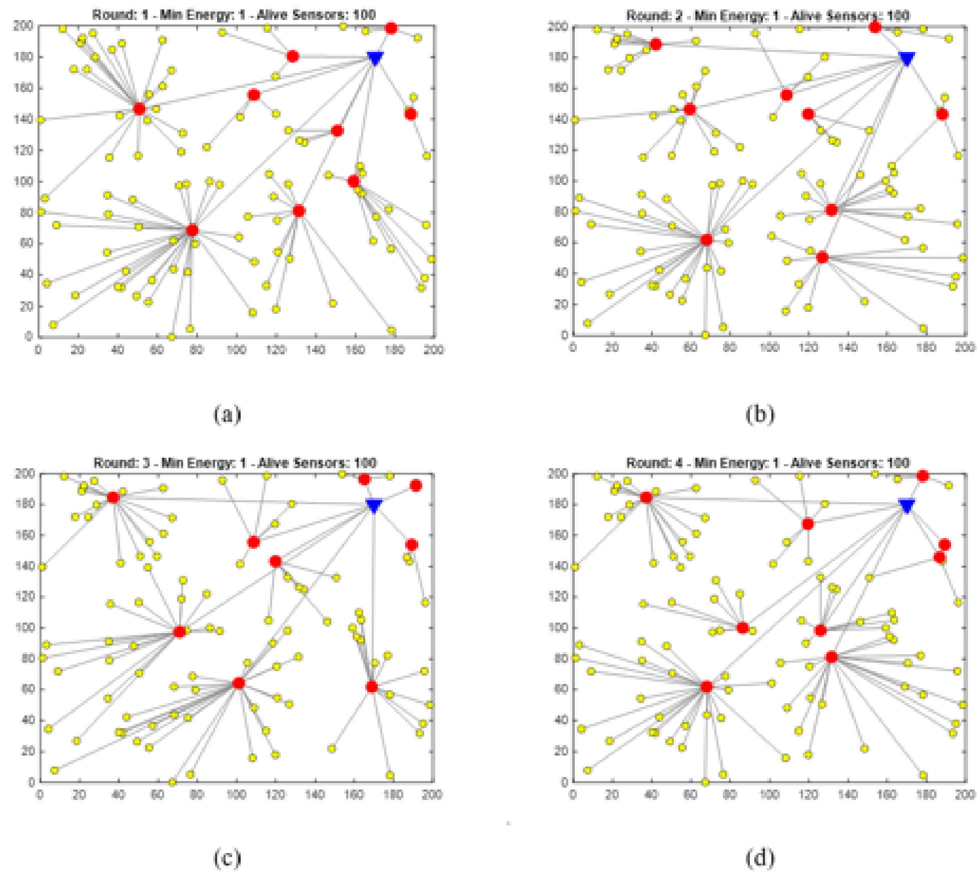


Figure 11.
The structure of the WSN in (a) round 1, (b) round 2, (c) round 3, and (d) round 4 of solving the proposed cost function using the WOA algorithm and K-means in the third scenario.

ISSN: 2576-8484
Vol. 8, No. 6: 8582-8610, 2024
DOI: 10.55214/25768484.v8i6.3848

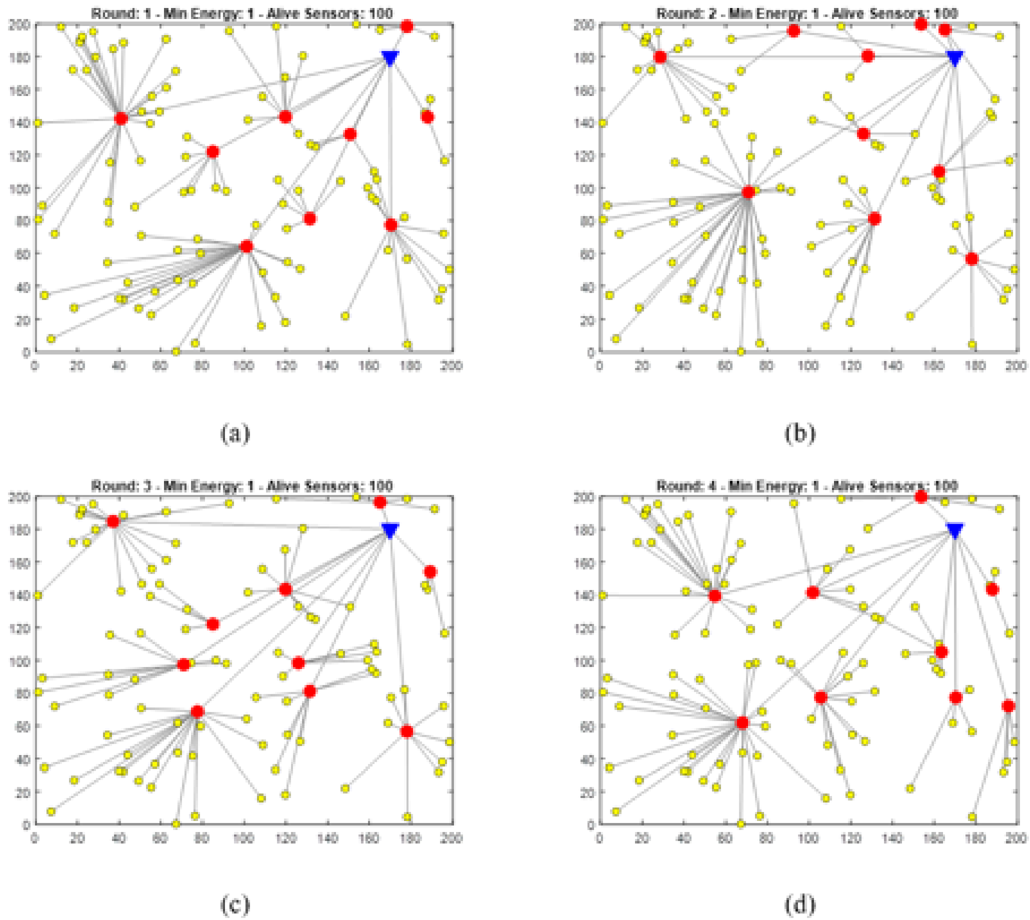


Figure 12.
The structure of the WSN in (a) round 1, (b) round 2, (c) round 3, and (d) round 4 of solving the proposed cost function using the GOA algorithm and K-means in the third scenario.

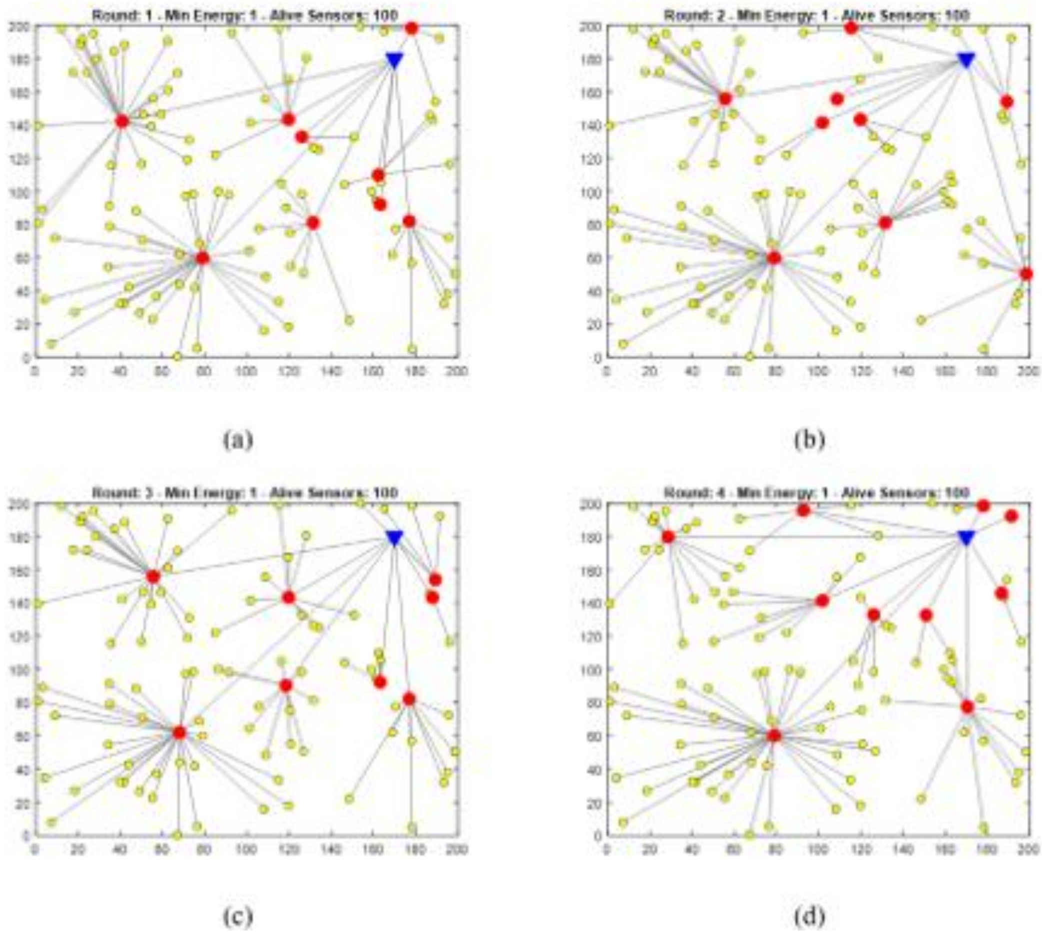


Figure 13.
The structure of the WSN in (a) round 1, (b) round 2, (c) round 3, and (d) round 4 of solving the proposed cost function using the BA algorithm and K-means in the third scenario.

5. Discussion

The findings of this study underscore the significance of algorithm selection in optimizing energy consumption and prolonging network lifespan in WSN. By evaluating the performance of three optimization algorithms—GOA, WOA, and BA—across varying scenarios, several key insights have emerged. In the first scenario, where the base station (BS) is centrally located, the GOA demonstrates a slight edge over WOA, performing marginally better in terms of network longevity. This outcome suggests that in environments where sensor nodes are equidistant from the BS, the swarm-based mechanisms of GOA are particularly effective at balancing load distribution and managing energy consumption. WOA, while close in performance, may benefit from additional refinement in such balanced topologies. BA, though competitive, does not achieve the same level of efficiency as GOA and WOA, indicating potential limitations in its search capabilities in this scenario. As the placement of the BS becomes more challenging, as seen in Scenarios 2 and 3, the differences between the algorithms become more pronounced. WOA emerges as the most robust solution, consistently outperforming both GOA and BA. Its ability to adapt to more complex and unbalanced node distributions highlights the effectiveness of its strategy in managing energy resources under more demanding conditions. GOA, while still a strong contender, begins to show a quicker decline in network performance as the scenario complexity increases, indicating that its performance may be more sensitive to changes in network topology. BA, on the other hand, struggles significantly in these scenarios, with a more rapid depletion of energy and earlier node failures. This pattern suggests that BA may be less suited for WSNs with highly uneven node distributions or when the BS is positioned far from the network center. One of the critical contributions of this research is the introduction of a novel cost function during the clustering phase. By simultaneously considering energy consumption and node distribution, this cost function enhances network flexibility, especially in the final rounds of operation. The K-means clustering used for initial cluster head placement has also proven beneficial in speeding up convergence and preventing the algorithms from being trapped in local optima. The combination of these techniques contributes to the overall success of the proposed method. All in all, the results of this study provide a comprehensive understanding of how different optimization algorithms perform in WSN clustering tasks under varying conditions. WOA, with its superior adaptability, emerges as the most effective algorithm overall, particularly in more complex scenarios. However, GOA shows promise in more balanced network configurations, and further refinement of these algorithms could lead to even greater improvements in WSN performance. Future research should explore additional factors and expand the scope of scenarios to continue advancing the state of WSN optimization techniques.

6. Comparison

This section offers an in-depth comparison between the proposed method and other approaches described in the literature. First, each method will be thoroughly explained, and then a brief summary of each method will be presented in a table format to facilitate comparison. An energy-efficient and reliable routing algorithm based on Dempster-Shafer (DS) evidence theory (DS-EERA) is proposed in [21]. DS-EERA first establishes three attribute indices—neighboring nodes' residual energy, traffic, and the proximity of its path to the shortest path—as the evidence. The entropy weight method is then used to objectively determine the weight of these indices. After establishing the basic probability assignment (BPA) function, the fusion rule of DS evidence theory is applied to merge the BPA function of each index value to select the next hop. Each node in the network transmits data through this routing strategy. Theoretical analysis and simulation results demonstrate that DS-EERA effectively prolongs network lifetime, achieving a lower packet loss rate and improving data transmission reliability. LND of DS-EERA is 1027 rounds for the BS located at the center of the study area. A novel clustering routing algorithm for WSN that combines the Sine Cosine Algorithm (SCA) with Lévy mutation to optimize energy efficiency and network lifetime is presented in [22]. The proposed method enhances traditional SCA by incorporating Lévy flight, which helps to avoid local optima and improve exploration capabilities. The authors evaluate their algorithm against several existing approaches using metrics such as network lifetime, energy consumption, and the number of alive nodes. The results demonstrate that the proposed method outperforms other algorithms in terms of energy efficiency and prolongs the network's operational time, making it suitable for WSN applications where energy conservation is critical. The obtained (HND) of this method for the BS located at the edge of interesting area is 945 rounds. All of the mentioned methods perform satisfactorily in WSN lifetime enhancement. However, all three methods proposed in this study outperforms them all due to its initialization using k-means and the well-crafted cost function

Table 5.
The comparison of the proposed method with other WSN clustering methods.

References	Method	HND for BS at center	LND for BS at center	HND for BS at edge	LND for BS at edge
[21]	DS-EERA (DS evidence theory)	-	1027 rounds	-	-
[22]	Combining sine cosine algorithm and Lévy mutation	-	-	945 rounds	-
Proposed method	Enhanced cost function / K-means / BA	2858 rounds	3308 rounds	1242 rounds	1767 rounds
	Enhanced cost function / K-means / GOA	2870 rounds	3402 rounds	1386 rounds	1917 rounds
	Enhanced cost function / K-means / WOA	2882 rounds	3364 rounds	1388 rounds	2022 rounds

7. Conclusion

This Article addresses the critical challenge of energy resource management in WSN by focusing on the clustering problem. Through the introduction and comparison of three optimization algorithms—Grasshopper Optimization Algorithm (GOA), Bat Algorithm (BA), and Whale Optimization Algorithm (WOA)—the research aims to optimize energy consumption and extend the network's lifespan. The proposed method leverages K-means clustering for the initial placement of cluster heads, combined with a novel cost function that considers both energy consumption and node distribution. This approach enhances the network's flexibility and efficiency, particularly in the later stages of operation. The experimental results across three different scenarios, each with varying base station (BS) locations, demonstrate the effectiveness of the proposed method and the relative performance of the algorithms. In the least challenging scenario, where the BS is centrally located, GOA performs slightly better than WOA, managing to extend the network lifespan further. However, WOA remains competitive, closely following GOA in performance. BA, while demonstrating good energy efficiency, falls short compared to both GOA and WOA in this scenario. As the complexity of the scenario increases, with the BS positioned at the edge of the area, the differences between the algorithms become more pronounced. WOA consistently demonstrates superior energy management, delaying the first node dying (FND) and extending the last node dying (LND) further than GOA and BA. GOA remains competitive, though it shows a quicker decline in network lifespan compared to WOA. BA, on the other hand, struggles in this scenario, with earlier node failures and a shorter overall network lifespan. In the most challenging scenario, where the BS is located in a distant corner, WOA once again proves to be the most resilient algorithm, effectively balancing energy consumption and maintaining network functionality for the longest period. GOA, while still performing well, shows a significant decrease in network lifespan compared to WOA, particularly in the later stages. BA, facing the greatest challenge in this scenario, experiences the fastest energy depletion, resulting in the shortest network lifespan among the three algorithms. Overall, the study highlights the importance of efficient clustering and optimization in WSNs for prolonging network life and improving energy management. Among the three algorithms evaluated, GOA outperforms WOA in the least challenging scenario, while WOA consistently outperforms GOA and BA across more complex scenarios, making it the most effective solution for the clustering problem in WSNs overall. The integration of K-means clustering and the newly designed cost function further enhances the performance of the optimization algorithms, ensuring balanced load distribution and reducing energy consumption. This research contributes to the ongoing efforts to develop more robust and energy-efficient WSNs, particularly in challenging environments where resource management is critical.

References

[1] Mamalis, B., Gavalas, D., Konstantopoulos, C., & Pantziou, G. (2009). Clustering in wireless sensor networks. In *RFID and sensor Networks* (pp. 343-374). CRC Press.

[2] Sasikumar, P., & Khara, S. (2012, November). K-means clustering in wireless sensor networks. In *2012 Fourth international conference on computational intelligence and communication networks* (pp. 140-144). IEEE.

[3] Subramani, N., Mohan, P., Alotaibi, Y., Alghamdi, S., & Khalaf, O. I. (2022). An efficient metaheuristic-based clustering with routing protocol for underwater wireless sensor networks. *Sensors*, *22*(2), 415.

[4] Saadati, M., Mazinani, S. M., Khazaei, A. A., & Chabok, S. J. S. M. (2024). Energy efficient clustering for dense wireless sensor network by applying Graph Neural Networks with coverage metrics. *Ad Hoc Networks*, *156*, 103432.

[5] Sulthana, N. N., & Duraipandian, M. (2024). EELCR: energy efficient lifetime aware cluster-based routing technique for wireless sensor networks using optimal clustering and compression. *Telecommunication Systems*, *85*(1), 103-124.

[6] Debasis, K., Sharma, L. D., Bohat, V., & Bhadoria, R. S. (2023). An energy-efficient clustering algorithm for maximizing lifetime of wireless sensor networks using machine learning. *Mobile networks and applications*, *28*(2), 853-867.

[7] Nedham, W. B., & Al-Qurabat, A. K. M. (2023). A comprehensive review of clustering approaches for energy efficiency in wireless sensor networks. *International Journal of Computer Applications in Technology*, *72*(2), 139-160.

[8] El Khediri, S. (2022). Wireless sensor networks: a survey, categorization, main issues, and future orientations for clustering protocols. *Computing*, 1-63.

[9] Surenther, I., Sridhar, K. P., & Roberts, M. K. (2023). Maximizing energy efficiency in wireless sensor networks for data transmission: A Deep Learning-Based Grouping Model approach. *Alexandria Engineering Journal*, *83*, 53-65.

[10] Mittal, M., de Prado, R. P., Kawai, Y., Nakajima, S., & Muñoz-Expósito, J. E. (2021). Machine learning techniques for energy efficiency and anomaly detection in hybrid wireless sensor networks. *Energies*, *14*(11), 3125.

[11] Dinesh, K., & Santhosh Kumar, S. V. N. (2024). Energy-efficient trust-aware secured neuro-fuzzy clustering with sparrow search optimization in wireless sensor network. *International Journal of Information Security*, *23*(1), 199-223.

[12] Lilhore, U. K., Khalaf, O. I., Simaiya, S., Tavera Romero, C. A., Abdulsahib, G. M., & Kumar, D. (2022). A depth-controlled and energy-efficient routing protocol for underwater wireless sensor networks. *International Journal of Distributed Sensor Networks*, *18*(9), 15501329221117118.

[13] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, *95*, 51-67.

[14] Nasiri, J., & Khiyabani, F. M. (2018). A whale optimization algorithm (WOA) approach for clustering. *Cogent Mathematics & Statistics*, *5*(1), 1483565.

[15] Meraihi, Y., Gabis, A. B., Mirjalili, S., & Ramdane-Cherif, A. (2021). Grasshopper optimization algorithm: theory, variants, and applications. *Ieee Access*, *9*, 50001-50024.

[16] Qin, P., Hu, H., & Yang, Z. (2021). The improved grasshopper optimization algorithm and its applications. *Scientific Reports*, *11*(1), 23733.

[17] Abualigah, L., & Diabat, A. (2020). A comprehensive survey of the Grasshopper optimization algorithm: results, variants, and applications. *Neural Computing and Applications*, *32*(19), 15533-15556.

[18] Yang, X. S., & He, X. (2013). Bat algorithm: literature review and applications. *International Journal of Bio-inspired computation*, *5*(3), 141-149.

[19] Sinaga, K. P., & Yang, M. S. (2020). Unsupervised K-means clustering algorithm. *IEEE access*, *8*, 80716-80727.

[20] Han, Y., Li, G., Xu, R., Su, J., Li, J., & Wen, G. (2020). Clustering the wireless sensor networks: a meta-heuristic approach. IEEE Access, 8, 214551-214564.

[21] Tang, L., Lu, Z., & Fan, B. (2020). Energy efficient and reliable routing algorithm for wireless sensors networks. Applied Sciences, 10(5), 1885.

[22] Guo, X., Ye, Y., Li, L., Wu, R. and Sun, X., 2023. WSN clustering routing algorithm combining sine cosine algorithm and Lévy mutation. IEEE Access, 11, pp.22654-22663.