applied sciences

MDPI

Article

Data Transmission in Wireless Sensor Networks Based on Ant Colony Optimization Technique

Lin Wu [1], Ahmad Yahya Dawod [1,*] and Fang Miao [2]

[1] International College of Digital Innovation, Chiang Mai University, Chiang Mai 50200, Thailand; lin_wu@cmu.ac.th
[2] Big Data Research Institute, Chengdu University, Chengdu 610106, China; miaofang@126.com
* Correspondence: ahmadyahyadawod.a@cmu.ac.th

Abstract: A wireless sensor network (WSN) is a wireless network composed of sensor nodes, widely used in military, industrial, and agricultural fields. In these practical application scenarios, due to the limited routing search ability of wireless sensor nodes and the small coverage range of a WSN, it faces the problem of a relatively short network lifecycle. Therefore, increasing a WSN's coverage range and reducing node energy consumption are of high value. This article proposes a method of using the ant colony algorithm (ACOD) to calculate the optimal routing of a WSN, the K-means algorithm for clustering, and the whale algorithm (WOA) and a backpropagation (BP) neural network to increase the coverage range of a WSN. After multiple experiments, the total energy consumption of WSN nodes has been effectively reduced, the coverage range of the WSN has been increased, and the lifecycle of the WSN has been extended.

Keywords: wireless sensor network; ant colony; K-means; whale optimization algorithm (WOA)

check for updates

Citation: Wu, L.; Dawod, A.Y.; Miao, F. Data Transmission in Wireless Sensor Networks Based on Ant Colony Optimization Technique. *Appl. Sci.* **2024**, *14*, 5273. https://doi.org/10.3390/app14125273

Academic Editor: Alessandro Lo Schiavo

Received: 7 April 2024
Revised: 7 June 2024
Accepted: 10 June 2024
Published: 18 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Wireless sensor networks are a key technology and an indispensable part of an expanding range of applications, from military operations to environmental monitoring, healthcare, and public safety initiatives [1]. As an indispensable component of the next-generation Internet of Things (IoT), wireless sensor networks are constantly evolving, consisting of numerous sensor nodes (SNs) that collectively perceive, collect, process, and transmit data in a self-organizing manner. Despite the increasing applicability and dependency of wireless sensor networks, there are still some challenges that hinder their performance optimization. It is crucial to resolve the issues related to EE, network lifespan, and hot issues that continue to disrupt overall network functionality and performance [2].

A major challenge in optimizing the performance of wireless sensor networks is to efficiently deploy nodes to enhance network coverage. Node deployment can be deterministic or random and continuous spatial or grid-based, and is a research hotspot in the field of wireless sensor networks. The current gold standard involves deterministic deployment, achieving maximum coverage with limited resources, although random deployment represents a suboptimal approach due to imbalanced deployment and high sensor requirements [3].

In addition, due to the limited battery power of wireless sensors, establishing energy-saving routing algorithms is a noteworthy issue. In traditional WSNs with static aggregation nodes, wireless sensors transmit information in a multi-hop manner, resulting in energy hotspots or energy holes. Nodes near the aggregation node quickly deplete their energy due to the concentration of data traffic. To address this issue, recent research has utilized the mobility of aggregation nodes and used the ant colony optimization algorithm (ACOD) to calculate the optimal trajectory for mobile aggregation nodes [4].

A clustering method has been developed for energy efficiency, data transmission reliability, and scalability to optimize sensor node layout and select the optimal number of

sensors [5]. However, current methods find it difficult to balance the number of sensors with the number of cluster nodes and the quality of the link between sensor nodes. A method of combining the K-means algorithm with the whale algorithm (WOA) has been proposed to optimize these challenges and provide a more realistic computational model [6].

As optimizing WSN performance is crucial for fully unleashing the potential of the Internet of Things, this article provides a comprehensive method for optimizing network coverage, routing, and clustering. We propose a holistic model that integrates the K-means algorithm for routing, the whale algorithm (WOA), and a BP neural network for the improved whale algorithm with maximum coverage [7].

The contribution of this study lies in developing an energy-saving, reliable, and scalable routing method, optimizing the performance of WSNs in terms of basic indicators, and developing a scalable network optimization model. The rest of this article provides a detailed introduction to the developed model, its implementation in a simulation environment, and an evaluation and comparison of performance indicators between our proposed model and traditional methods.

In the past few decades, wireless sensor networks have become a critical component in many applications, including military operations, public safety systems, environmental monitoring, healthcare, and more. This increasing correlation is due to the numerous advantages of wireless sensor networks, including their self-organizing ability, distributed network characteristics, and widespread application efficiency. However, the inherent characteristics of a single sensor node, such as limited power supply and transmission range, still pose a daunting challenge in designing effective and efficient routing protocols for wireless sensor networks.

The basic premise of a typical cluster-based architecture is to divide the network runtime into several rounds, each consisting of three stages—cluster head (CH) selection, cluster formation, and data transmission. Although there have been some successful implementations, some current clustering protocols face limitations due to their inability to accurately simulate clustering in sensor networks, which may lead to unrealistic estimates of power consumption. In addition, reliability issues often arise from protocols relying solely on the distance between two nodes as an indicator of their link quality, thereby ignoring the inherent asymmetric characteristics of WSNs.

In addition, algorithms that maximize network coverage play an important role in extending network lifespan, enhancing network load balancing, and improving network robustness, which has aroused great interest. Various strategies have been proposed to increase coverage, but achieving maximum coverage on a limited number of sensors remains a research hotspot in the field of wireless sensor networks. The main challenge lies in optimizing the position and distribution of sensors, which has been proven to be a non-deterministic polynomial (NP) (NP) problem.

Given these challenges and growing demands, we are committed to providing efficient, scalable, comprehensive, and optimized solutions for the performance of wireless sensor networks. This study is expected to make significant contributions by highlighting innovative methods and strategies to improve the performance, reliability, and effective deployment of wireless sensor networks.

1.1. Goals and Contributions

The main purpose of this study is to propose and implement a comprehensive model for optimizing network performance based on wireless sensor networks. This study aims to address key issues that plague modern wireless sensor networks, such as low energy efficiency, hotspot issues, and a shortened network lifespan, which can significantly reduce overall network performance. To alleviate these issues, the model integrates ant colony optimization-based routing, K-means clustering, and an improved whale algorithm.

Specifically, our proposed model divides the network into different clusters, each led by a CH, and uses a combination of the K-means algorithm and whale algorithm (WOA) to determine the optimal node layout with the optimal number of CHs, cluster nodes, and

link quality. In order to balance the load of wireless sensor nodes, the ACOD is applied to determine the optimal path for mobile aggregation nodes to communicate with CHs, while considering the distance between them. This method greatly improves the energy-saving effect of the network in the communication process. Finally, in order to expand coverage, we adopted an upgraded whale algorithm [8].

The prominent feature of this article is that it effectively solves the clustering, routing, and network coverage problems in wireless sensor networks simultaneously. The implementation of a new individual coding mechanism has optimized the interaction between the two. In addition, considering real-world network coverage and communication models, the accuracy of performance evaluation has been improved. This article also conducts critical research on different types of networks based on sensor energy, homogeneous networks and heterogeneous networks, and conducts important simulation experiments on several famous clustering protocols.

The main contributions of this study can be summarized as follows:

I. Solving the coverage optimization problem of WSNs (wireless sensor networks):

We explore how to effectively deploy sensor nodes while ensuring sensor network connectivity to maximize the coverage of monitoring areas and avoid blind spots and node redundancy. This article studies the WOA-BP coverage algorithm, which achieves a more uniform node distribution and higher coverage.

II. Solving energy consumption and optimization

Sensor nodes in WSNs are usually powered by batteries and have limited energy. A large amount of data perception, processing, and transmission leads to rapid energy consumption, reducing the lifecycle of sensors.

This article adopts the K-means algorithm for routing. By improving the clustering method between sensor nodes, the optimal path for data transmission is achieved, and the energy consumption of nodes is reduced.

III. Solving data processing and performance optimization

Due to the large scale of nodes in WSNs, a huge amount of data is generated, and the types of data are diverse, including numbers, text, images, and videos. This leads to network congestion, high storage costs, and insufficient upload bandwidth. This article adopts the ant colony algorithm (ACOD) to calculate the optimal routing algorithm for WSNs, optimize network protocol design, and improve network transmission efficiency.

1.2. Related Work

Recently, countless technologies have been proposed to improve the performance of wireless sensor networks [9]. Among them, clustering and routing optimization have become key methods that have made significant contributions to network lifespan and efficiency [10].

Various protocols, such as the Low-Energy Adaptive Clustering Protocol (LEACH) and its variants, namely the Enhanced Heterogeneous LEACH Protocol (EHE-LEACH) for extending lifespan, the Single-Hop Energy-Saving Clustering Protocol (S-EECP), and the Multi-Hop Energy-Saving Clustering Protocol (M-EECP), have been developed to optimize clustering based on local information of SNs. These protocols use probability calculations to elect CHs and ensure the distribution of energy loads between clusters. However, their decision-making process is limited by limited local information and often excludes the selection of the optimal CH set.

To improve these protocols, the literature proposes using clustering protocols based on evolutionary algorithms (EAs). These protocols utilize the global knowledge of the network and optimize CH selection using predefined objective functions. For example, the LEACH-C protocol is optimized using the simulated annealing (SA) method, while the clustering protocol based on particle swarm optimization (PSO-C) and the CH protocol based on particle swarm optimization (PSO-CH) use PSO technology, which considers EE, data

transmission reliability, and scalability in the EC model [11]. At the same time, some people have focused on combining clustering and routing, resulting in two-layer protocols such as TPSO-CR and iCSHS. The TPSO-CR protocol attempts to solve inter-cluster communication problems through a two-layer approach, which determines the optimal routing tree and CH. However, the iCSHS protocol uses optimization algorithms based on cuckoo searching and routing algorithms based on improved harmonious searching. However, these protocols also bring their own challenges, including potential higher EC due to an increase in the number of active nodes and extended execution time, which requires further research [12].

In addition, significant progress has been made in the routing protocols of wireless sensor networks, such as the Mobile Convergence-Based Routing Protocol (MSRP) and the Energy-Saving Routing Protocol based on PEGASIS, which are being developed to alleviate the problem of low energy efficiency. The emergence of mobile convergence points, combined with ant colony optimization techniques, has provided significant improvements in EE and network lifespan [13].

Despite these efforts, the challenge of optimizing network performance in WSNs still exists. Specifically, most protocols assume that the SN is location aware, which is unrealistic in practical situations. In addition, using traditional first-order radio models for EC calculations often leads to overly idealized estimates, ignoring a considerable amount of EC from monitoring activities. Therefore, a realistic and comprehensive model is needed to optimize network performance in WSNs to address these issues.

A. Overview of Previous Research

In recent years, wireless sensor networks have undergone tremendous development and have been widely used in various social and military applications, including habitat monitoring, disaster intervention, medical care, production equipment status recognition, and emergency search and rescue. The functionality and efficiency of these applications largely depend on the coverage range of wireless sensor networks, which in turn depends on the effective deployment of sensor networks. However, traditional node deployment algorithms, including deterministic and random deployments, have inherent limitations [14].

In deterministic deployment, the position of nodes is selectively determined to achieve specific deployment goals, providing maximum coverage with a limited number of sensors. However, the node selection process is often plagued by computational complexity, especially in large-scale networks. On the other hand, although random deployment provides a suboptimal approach that is easy to implement, it often leads to imbalanced deployment, requiring additional sensors to achieve the same level of coverage as deterministic deployment.

To address these shortcomings, researchers have introduced advanced algorithms such as the whale colony algorithm and ant colony optimization algorithm to deploy nodes. For example, a recent study proposed a new whale swarm optimization algorithm that transforms the original network coverage optimization problem into a multi-objective optimization problem. By combining this algorithm with a BP neural network, the initial population distribution is more ideal and the search ability of nodes is enhanced. Meanwhile, the ant colony optimization algorithm (ACOD) developed for mobile aggregation nodes provides an optimal trajectory, with distance being the determining factor.

Despite these advancements in node deployment, existing technologies still suffer from drawbacks such as low energy efficiency, shortened network lifespan, and hot issues that adversely affect network performance. Each of these challenges requires a comprehensive approach that can provide reliable, scalable, and energy-efficient alternatives for routing in wireless sensor networks, laying the foundation for our current research.

B. Gap analysis

Due to the potential for improving efficiency in wireless sensor networks, they hold great potential in various application fields. However, despite extensive research on this topic, there are still some challenges that have not been resolved.

Firstly, existing methods for optimizing WSN performance using algorithmic techniques require sensor nodes to be in a working or dormant state, with the latter consuming less energy. However, this method does not take into account the energy conversion that occurs during the state transition process, nor does it take into account the energy waste that may occur when the node is in a dormant state for a long time. Therefore, there is a knowledge gap in how to best balance energy use between active and dormant states and optimize performance.

In addition, although the use of mobile aggregation nodes in traditional optimization methods has been proven to prolong network lifespan, it cannot effectively manage "hotspot" issues. These strategies also do not take into account the impact of the distance between CHs and mobile aggregation nodes on the overall performance of the network. Traditional ant colony algorithms only use current information to guide ants to subsequent nodes, which leads to them deviating from the target over time. Therefore, there is a gap in developing a technique that incorporates distance factors into finding the optimal movement trajectory for mobile aggregation nodes.

The key is to design a solution that optimizes the initial population distribution to improve global search efficiency and ultimately improve network coverage [15].

The whale algorithm is a swarm intelligence-based algorithm, inspired by the collective behavior of whales. Whales form a group when swimming, and they cooperate with each other to achieve common goals through mutual coordination. This behavior can be seen as a form of collective intelligence, as they can achieve complex tasks through mutual communication and collaboration [16].

In the process of information transmission in wireless sensor networks, there may be obstacles or signal interference that affect coverage. To achieve this task, several nodes in wireless sensor networks, like whale schools, need to perceive the surrounding environment in real time and be able to make quick decisions. Therefore, this algorithm process adopts the whale algorithm to achieve fast and accurate decision-making [17].

The first step in the algorithm process is to establish an environmental model. This model includes the location, speed, and information content of several nodes in a wireless sensor network, as well as the location and size of obstacles in the surrounding environment [18]. This model can be implemented through sensors and other devices. Once an environmental model is established, information transmission among several nodes in a wireless sensor network becomes smooth.

In sensor node data transmission, nodes will perceive the surrounding environment in real time and make decisions based on whale algorithms. The basic idea of the whale algorithm is to divide the entire population into several subgroups and have each subgroup search for the optimal solution separately [19]. This process can be achieved iteratively, with each iteration updating the state of the population and finding the optimal solution.

In this algorithm process, several wireless sensor nodes can be regarded as a subgroup whose task is to find the optimal path. To achieve this task, wireless sensor nodes will use the whale algorithm to find the optimal solution [20]. This process can be seen as an optimization problem, which involves finding a path that enables wireless sensor nodes to transmit information in complex terrain and avoid collisions with obstacles [21].

2. Materials and Methods

2.1. Ant Colony

The ant colony optimization algorithm is a probabilistic algorithm that simulates the foraging behavior of ants in nature, with characteristics of distribution calculation, positive feedback of information, and heuristic searching [22]. In ant colony optimization-based routing algorithm simulations, the parameters are α, β, and ρ. The numerical value of these parameters is crucial for the performance and effectiveness of the algorithm. These parameters play different roles in ant colony algorithms and affect the decision-making process of ants when choosing paths.

Parameter α: The factor that usually represents the importance of pheromones determines the degree to which ants are influenced by pheromones when choosing a path. A pheromone is a chemical substance released by ants during their search for food, used to transmit pathway information [23]. When the value of α is high, ants are more inclined to choose paths with higher pheromone concentration, which helps the algorithm quickly converge to the optimal solution. However, an excessively large α value may cause the algorithm to fall into local optima too early while ignoring other potentially better paths [24]. Therefore, adjustments need to be made based on specific issues deriving from the α value to strike a balance between global search and local search.

Parameter β: The factor representing the importance of heuristic information reflects the degree to which ants are influenced by heuristic information (such as path length, congestion level, etc.) when choosing a path [25]. Heuristic information helps ants discover better paths during the exploration process. When the value of β is large, ants are more inclined to choose paths with better heuristic information, which helps the algorithm to avoid becoming stuck in local optima during the search process. However, an excessively large β value may cause the algorithm to ignore the role of pheromones, thereby reducing the convergence speed of the algorithm. Therefore, it is also necessary to make adjustments according to specific issues deriving from the β value to achieve an efficient search process of the algorithm.

Parameter ρ: This parameter indicates the volatility coefficient of pheromones, which determines the rate of decay of pheromones along the path. The volatilization of pheromones is an important step in the ant colony algorithm, which helps to prevent the algorithm from becoming stuck in local optima and promotes ants to explore other paths. When the value of ρ is large, pheromone decay is faster, which helps ants discover new and better paths more quickly [26]. However, an excessively large ρ value may cause pheromones to quickly disappear on the path, making it difficult for ants to find a stable and optimal path [27]. Therefore, it is necessary to choose the appropriate one based on the characteristics and requirements of the problem. The ρ value is used to achieve the stability and efficiency of the algorithm [28].

Parameters α, β, *and* ρ. The data size setting is not supported by a comprehensive mathematical model. It is necessary to comprehensively consider factors such as the characteristics of the problem, the complexity of the network environment, and the performance requirements of the algorithm [29]. Through reasonable parameter settings and adjustments and multiple experimental optimizations, the efficient, stable, and accurate performance of ant colony optimization-based routing algorithms can be achieved [30].

Assuming there is a base station with n and m wireless sensor nodes, these m wireless sensor nodes can choose the next base station with a certain probability, and $P_{ij}^{k}(t)$ is the probability of the k-th sensor node from base station i to base station j at time t:

$$f(x) = \begin{cases} \frac{\tau_{ij}^{\alpha}(t)\ \eta_{ij}^{\beta}(t)}{\sum_{s \in allowed_k} \tau_{is}^{\alpha}\ \eta_{is}^{\beta}},\ j \in allowed_k \\ 0,\ j \not\in allowed_k \end{cases} \quad (1)$$

Among them, $\tau_{ij}(t)$ represents the pheromone content on path (i, j) at time t; $\eta_{ij}(t)$ is the visibility on path (i, j) at time t, which is the heuristic information of path $(i,\ j)$. It is generally set as η_{ij}=$1/d_{ij}$, where d_{ij} is the distance between base station i and j; $allowed_k$ is the base station that node k can access in the next step; α is the heuristic factor, representing the relative importance of pheromones $(\alpha \geq 0)$; P is the expected factor, representing the importance of visibility $(\beta \geq 0)$. And the corresponding pheromones left by the sensor nodes on the route they pass through are as follows:

$$\Delta\tau_{t+1} = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij} \quad (2)$$

$$\Delta\tau_{ij} = \sum_{i=1}^{m} \Delta\tau_{ij}^{k} \tag{3}$$

Among them, ρ is the volatility coefficient $(0 < \rho < 1)$ of the pheromones, $1 - \rho$ is the residual coefficient of the pheromones, is the pheromone content left by node k on path (i, j), and $\Delta\tau_{ij}$ is the sum of the pheromones left by all ants passing through path (i, j) at the end of a cycle.

Using the experimental data from the TSPLIB test library as the object, the main analysis focuses on the total number of wireless sensor nodes m, the heuristic factor α, the pheromone volatility coefficient ρ, and the expected factor β. After 15 tests, Table 1 is obtained:

Table 1. The experimental results using the ant density model.

	Parameter	Average Value	Optimal Solution	Worst Solution
α	0	688.67	660.45	714.34
	0.5	479.56	451.98	503.65
	1	463.34	443.23	479.45
	2	511.67	489.12	530.23
	5	677.83	660.84	714.56
β	0	929.26	887.78	957.87
	1	463.24	443.78	479.88
	2	429.44	423.82	437.91
	5	428.89	423.90	433.56
	10	427.78	424.66	430.34
	20	431.88	426.45	438.83
ρ	0.3	633.89	612.56	663.78
	0.5	604.24	619.91	566.67
	0.7	463.24	443.74	479.17
	0.9	433.48	426.67	440.63
	0.999	434.89	429.94	440.94

The range of α values is [0.5, 5], the range of β values is [1, 20], and the range of ρ values is [0.3, 0.99]. The obtained optimal solution of α is 5, of β is 0, and of ρ is 0.5

2.2. WSN Shortest Distance

For the convenience of calculations, the wireless sensor network is divided into a grid with each intersection point as the wireless sensor node. In Figure 1, the turtle is the wireless sensor node A, and the little white rabbit is the wireless sensor node B. Each line in the figure can pass through. How can we find the shortest distance between the turtle and the little white rabbit?

Suppose there is a turtle chasing a little white rabbit. The position of the turtle is as shown. The position of the little white rabbit is as shown. Suppose the little white rabbit is stationary. If the turtle can only move left, right, up, and down, then these are the points that the turtle can reach in the next round, and these points are called the points to be explored.

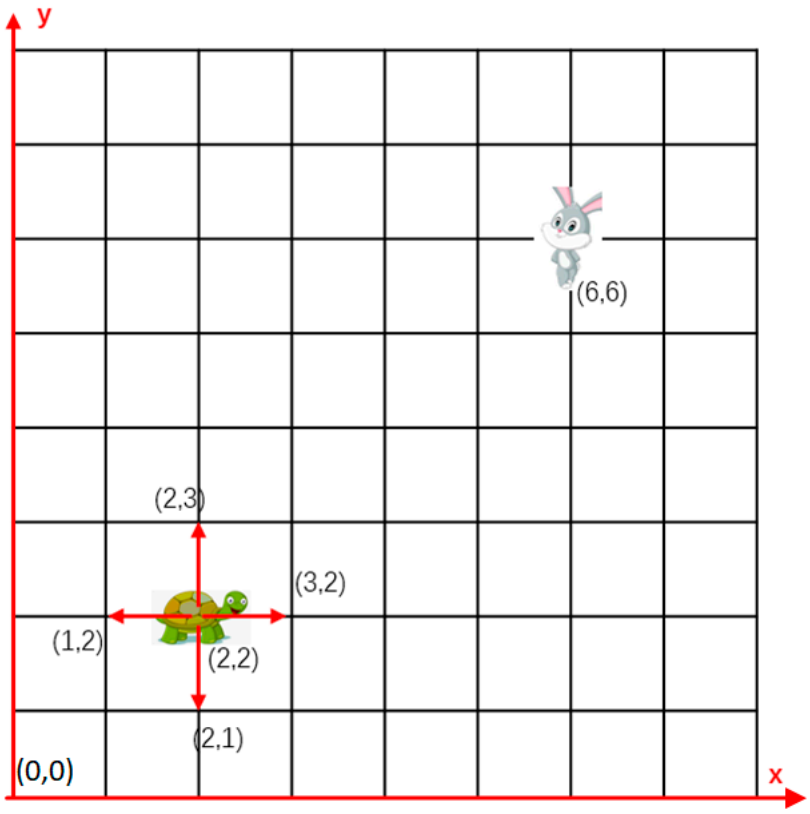


Figure 1. Initial position of turtle.

2.2.1. Step 1

Find the nodes that can be reached in the next step and add these points to the frontier array to be explored, also known as the boundary array.

Calculate the cost of the new addition point, where cost (F) = current cost (G) + estimated cost (H), expressed as F = G + H.

The current cost G is the distance traveled from the starting point to the current point; for example, the current cost of these four points is equal to the distance required to reach (2,2) points, which is 1.

The estimated cost H is the Manhattan distance from the current point to the endpoint (the sum of the differences in the horizontal and vertical coordinates).

So, the estimated costs for the four points to be explored are 9, 9, 7, and 7, respectively.

The current cost/estimated cost result is shown in the Figure 2:

2.2.2. Step 2

Arrange the points to be explored in ascending order of cost; the order of the points to be explored in the sorted array to be explored is: (3,2), (2,3), (1,2), (2,1).

Next, take out the first point with the lowest cost as the target point for this round of the little turtle, which is the starting point for the next round. Remove this point from the frontier array to be explored and add it to the already explored array already'frontier. The following situation will be obtained Figure 3:

At this point, the little turtle is at position (3,2), and for a more vivid expression, the points to be explored are marked in blue and the points that have been explored are marked in green (currently these are (2,2) and (3,2), and the points to be explored are (1,2), (2,1), and (2,3)).

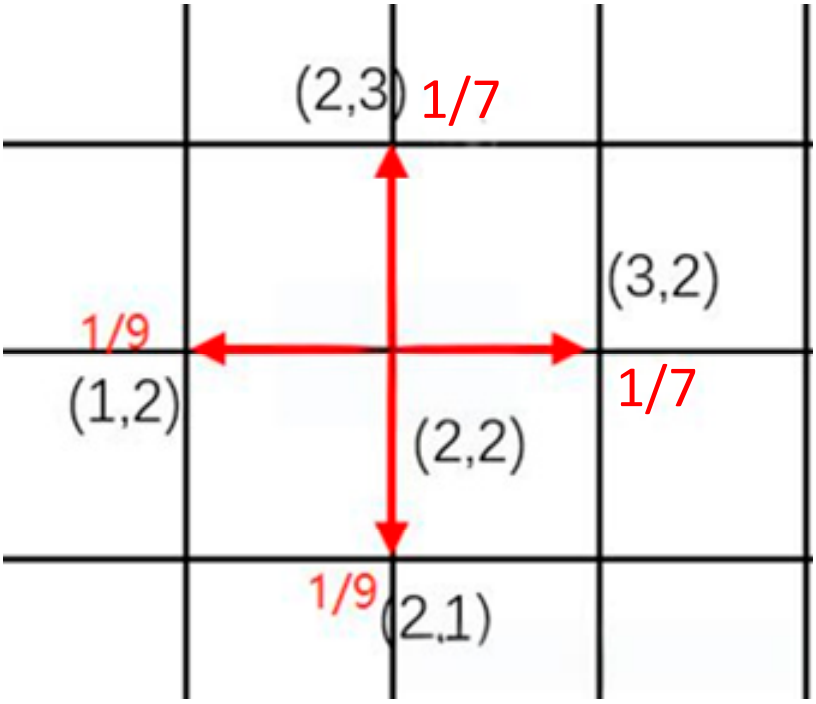


Figure 2. Turtle displacement.

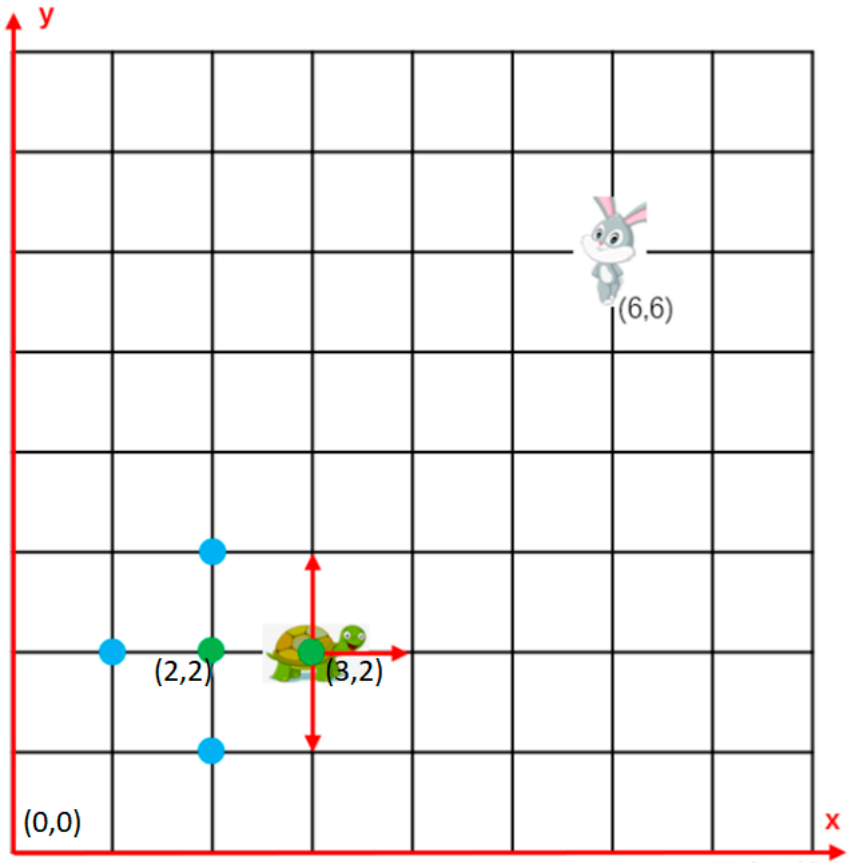


Figure 3. Turtle's 2nd position.

2.2.3. Step 3

Record the path from the current point to the starting point, which can be represented as a matrix of 2 rows and 2 columns in Matlab [(2,2), (3,2)].

Next, determine whether the current node is the endpoint. If it is not the endpoint, continue with step 1 and search for the next point that can be explored out of (3,3), (4,2), and (2,1).

As shown in Figure 4, the next step is to find the points that the little turtle can go to in the next round and check whether the surrounding points are in the explored array alreadyfrontier. If they are, ignore them, and then check whether these points are already in the explored array frontier. If they are, compare the current cost G of the point with the current cost G2 required to reach the current position after passing through the little turtle's current position. If G > G2, change the previous node of the point (which can be understood as the parent node) to the node where the little turtle is currently located, and update its current cost to G2.

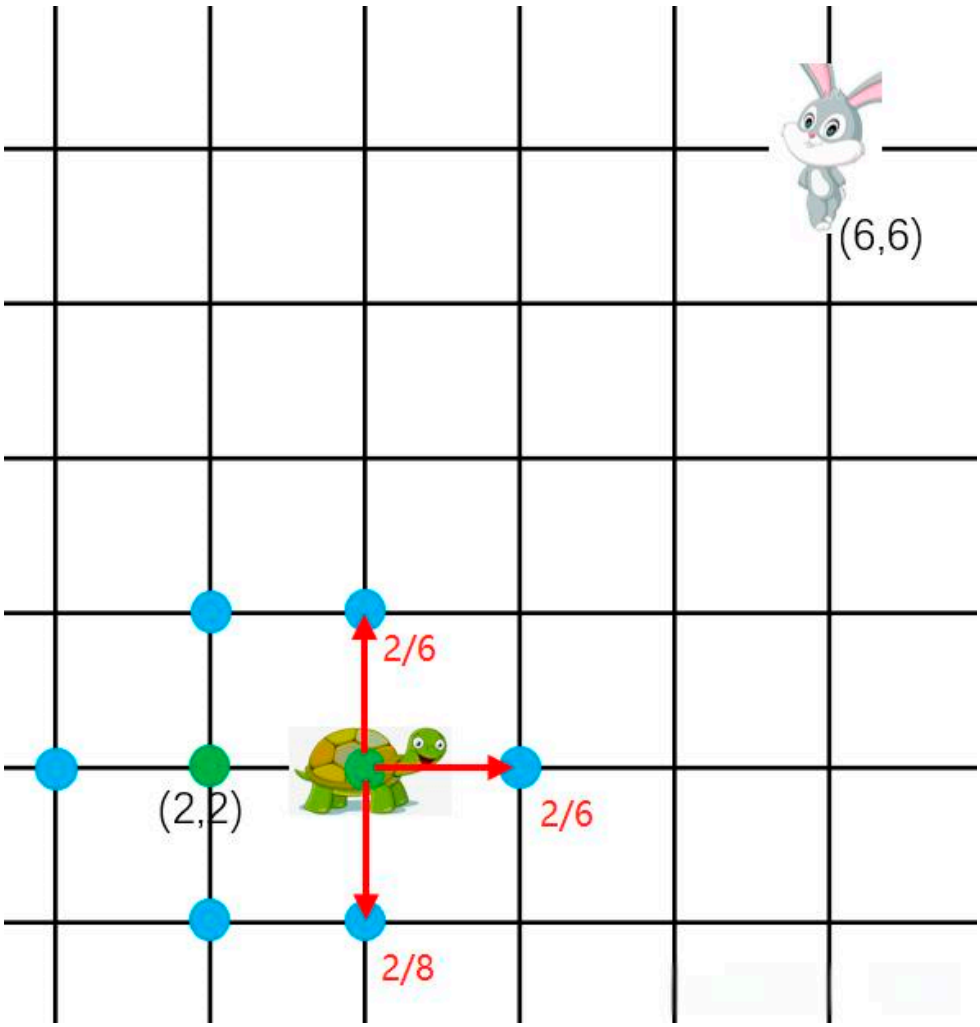


Figure 4. Turtle's second position representing cost.

As shown in Figure 5, By searching for the points to be explored in the order of bottom left to top right, the minimum cost obtained in the first two selections is (3,2) and (4,2). If you follow the order of top right to bottom left, as shown in Figure 6, the minimum cost obtained will be (2,3) and (2,4). This does not affect the final shorte st path length, but the paths are different. By repeating the above three steps, you can obtain the shortest path. The green dots in the figure represent the shortest path.

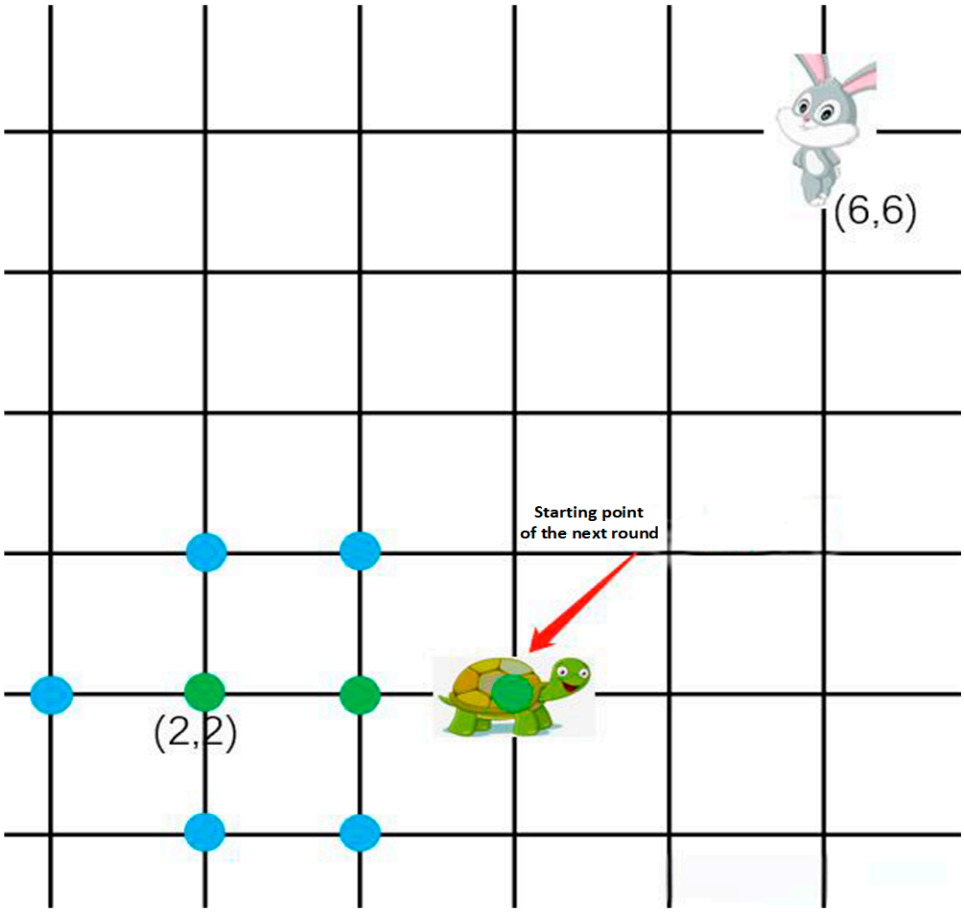


Figure 5. Turtle's 3rd position.

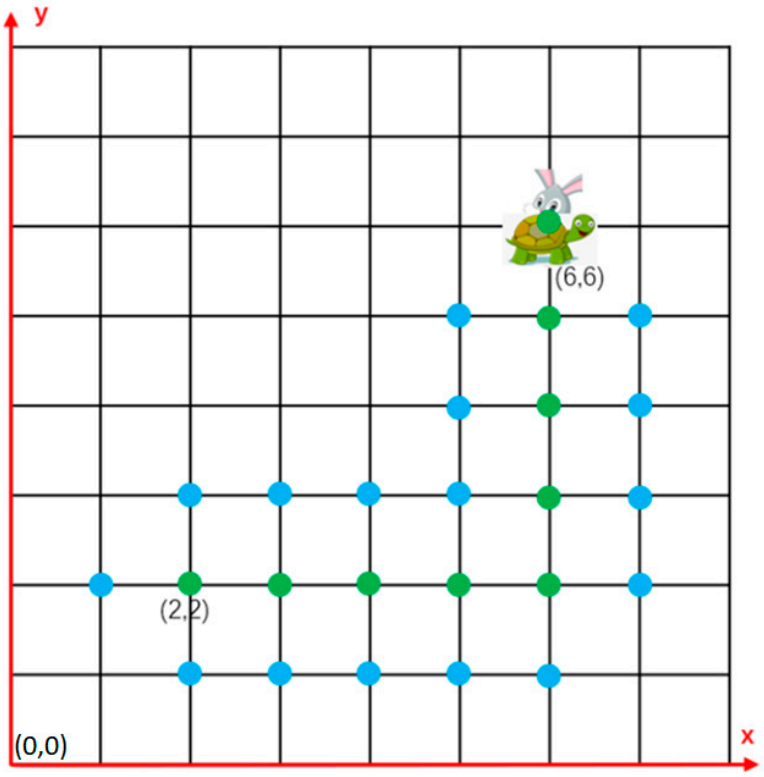


Figure 6. Turtle's path to finish line.

2. Implementation of steps in WSN shortest distance algorithm.

(1) Add the WSN's starting node to the frontier array to be explored, with a cost of 0.
(2) Identify the node with the lowest cost F in the frontier, remove it from the frontier, and add it to the explored array already_frontier.
(3) Determine if the point is the endpoint; if not, continue.
(4) Identify the four points around the WSN node and determine if they are in the already'frontier array. If they are, ignore them (of course, these four points cannot exceed the map range or be located in obstacles).
(5) Determine whether these points are already in the frontier of the array to be explored. If they are in the frontier of the array to be explored, compare the current cost G of the point with the current cost G2 of reaching the point through the current node. If G > G2, change the previous node of the point to the current node, update its current cost to G2, and update its cost F.
(6) Add the remaining points that meet the requirements to the frontier, calculate the cost F, and arrange the frontier array according to the cost F.
(7) Simultaneously record the path, that is, each node should record its previous node to facilitate the final backtracking of the path and find the shortest path.

Repeat steps (2) to (7) until reaching the endpoint to terminate the loop.

2.3. The Combination of the Ant Colony Algorithm and Finding the Shortest Path Algorithm

The ACOD algorithm steps:

The specific steps of the ACOD algorithm are as follows:

Step 1: each node sends its current energy and position coordinates to the Sink node.

Step 2: We use the WSN shortest distance algorithm to calculate the initial optimal path from the Sink node to each sensor node in the network, and reverse plan the routing path from each node to the Sink node. And we send the routing path to each sensor node.

Step 3: each node adopts the optimal path routing generated by the Sink node.

When a node in the network fails or its energy reaches a threshold, the local node that uses this node for route forwarding uses an improved ant colony algorithm to dynamically plan the routing path.

Step 4: k ants start from each local node and select the next hop node based on Equations (1) (ant colony) and (2) (ant colony), until the Sink node is reached.

Step 5: When reaching the Sink node, the Sink node generates a return ant that returns along the original path, and during the return process, it updates the pheromones of each node according to Equation (3) (ant colony).

If the maximum number of iterations is reached, the algorithm ends. Otherwise, we return to step 4 to continue searching for the optimal path.

Step 6: When the algorithm reaches the maximum number of iterations Ite, we re-evaluate the local nodes.

We generate a new routing path and route according to the routing path.

Using Matlab2023a experimental simulation, we obtain the wireless sensor area: the black lines represent obstacles or WSN fault nodes, and the red lines represent WSN optimal paths.

As shown in Figures 7 and 8, "*" represents the wireless sensor node, and the black line represents the obstacle. When the wireless sensor node encounters the obstacle, the signal transmission is blocked and the network topology changes. The shortest path has been found, with a total distance of 76.8112 and a time of 0.24689 s.

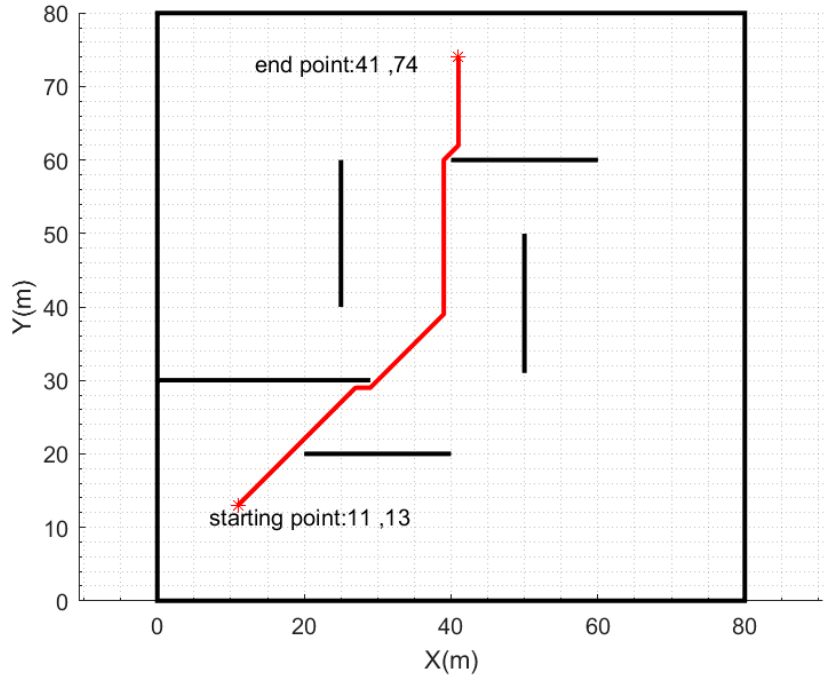


Figure 7. An optimization path.

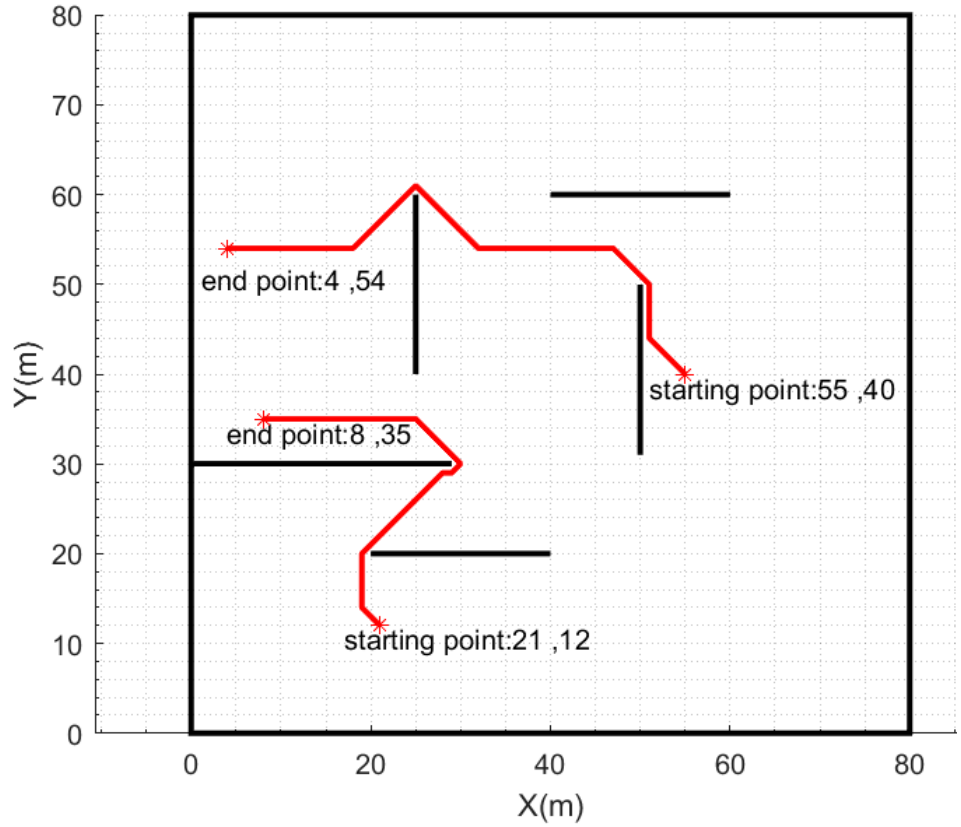


Figure 8. Multiple WSN optimization paths.

2.4. Increase the Coverage of WSNs

We use a newly added optimized clustering method to determine WSN clustering.

In wireless sensor network clustering, the number of clusters and initialization centers are uncertain and need to be improved.

(1) We use the canopy algorithm to preprocess the data:

In order to increase the anti-interference ability, the K-means clustering algorithm has been optimized and improved. The specific algorithm is as follows (as shown in Figure 9):

Randomly delete one point from a set of points, create a canopy containing that point, and iterate over the remaining set of points.

For each point, if its distance from the first point is less than T1, then that point is added to this cluster.

In addition, if the distance is less than T2, remove this point from this set. This point, which is very close to the origin, will avoid all future calculations and no longer focus on other canopy centers. This algorithm loops until the initial set is empty, aggregating a set of canopies, each containing one or more points. Each point can be included in more than one canopy. After establishing the canopies, those containing fewer data points can be deleted, which can reduce signal interference.

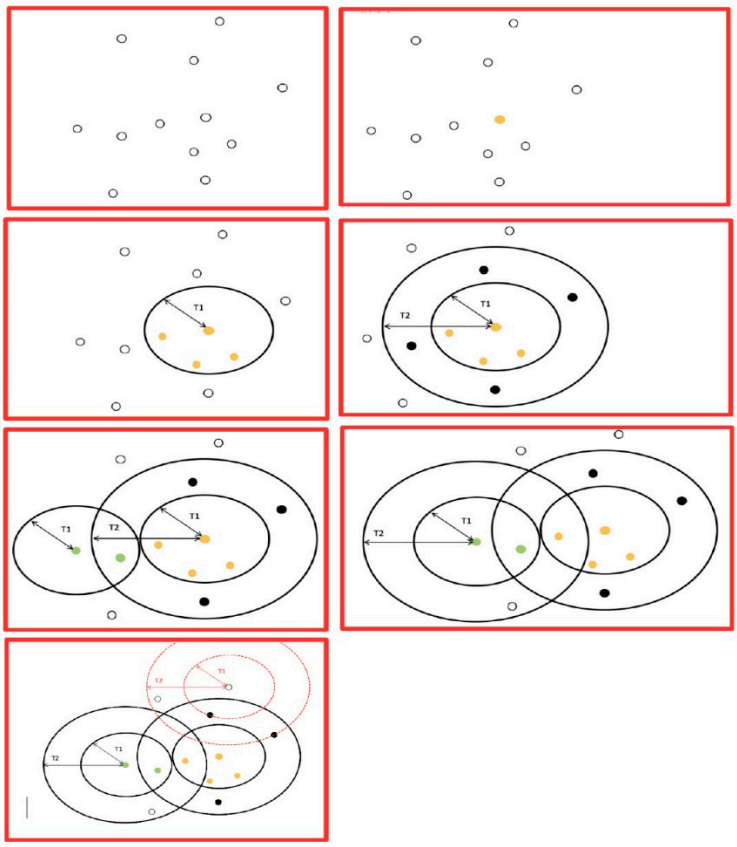


Figure 9. The steps of the Canopy algorithm are as follows.

(1) Put all data into the list and select two distances T1, T2, T1 > T2.
(2) While (list is not empty)

{
Randomly select a node as the center of the canopy and delete the point from the list;
Traversing the list:
For any record, calculate its distance to each canopy;
If the distance is less than T2, mark these data strongly and delete this record from the list;
If the distance is less than T1, mark these data weakly;
If the distance to any canopy center is greater than T1, then treat this record as a new canopy center and remove this element from the list;
}

It is important to note the adjustment of parameters:

When T1 is too large, it can cause many points to belong to multiple canopies, which may result in closer distances between the center points of each cluster and less obvious differences between clusters.

When T2 is too large, increasing the number of strongly labeled data points will reduce the number of clusters; if T2 is too small, it will increase the number of clusters and also increase the calculation time.

(2) Optimizing clustering methods to determine WSN clustering

i. Optimize the selection of initial cluster centers

As shown in Figure 10, if cluster centers are too dense, most data points will form a cluster. Clustering centers that are too scattered can lead to sparse data points and unreasonable classification. Choosing the optimal cluster center is crucial. An excellent cluster center needs to meet the high density of nodes around the cluster center and the long distance between each cluster center.

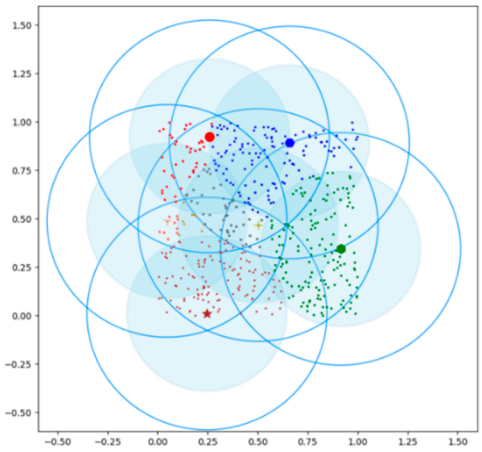


Figure 10. Clustering of the canopy algorithm.

(a) Average distance

Data point collection is as follows: $X = \{X_1, X_2, \cdots, X_n\}$. Calculate the average distance between all data points as follows:

$$AveDis = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} d(X_i, X_j) \tag{4}$$

In the formula, represents the distance between two points.

$d(X_i, X_j)$ represents the distance from node X_i to node X_j, n represents the maximum number of wireless sensor nodes, and $AveDis$ represents the average distance between all nodes.

(b) Cluster radius

When the number of cluster centers and the total cluster area are determined, the ideal radius for obtaining clusters is as follows:

$$\mathrm{R} = \frac{S}{n} \tag{5}$$

(c) Data point density

Calculate the standard deviation of the distance between each data point and other data points as the density of that data point. The density of the data point is $density_i$, calculated as follows:

$$density_i = \frac{\sqrt{\sum_{j-1}^{n}(d(X_i, Y_i) - AveDis)^2}}{\sqrt{n-1}} \quad (6)$$

The basic idea for selecting the initial clustering center for data point density is to use the data point with the highest density as the first initial clustering center C_1. Then, calculate the distance from the remaining data points to C_1 (Euclidean distance); if the distance is less than the average distance *AveDis* between each data point, then move it into the class of C_1. Then, select the data point with the highest density from the data points outside the cluster radius of the first initial cluster center C_1 as the second initial cluster center C_2, and continue the cycle until k initial cluster centers are selected.

The specific steps for selecting the initial cluster center are as follows:

(1) Initialize the parameter k, cluster radius r, and data point set X;
(2) Calculate the average distance between all non-cluster-center data points in X according to Formulas (1) and (3) *AveDis* and the data point density *density* in X;
(3) Select the data point with the highest density among all data points as the first cluster center C_1.
(4) Select the data point with the highest density of data points outside the initial cluster center I and cluster radius J as the second cluster center C_2;
(5) Continuously perform steps 2, 3, and 4 until k initial cluster centers are selected.

ii. Improve the K-means algorithm process

Combine the canopy algorithm with the K-means algorithm to improve the K-means algorithm process. Set the set of data points in the improved K-means algorithm as follows:

$$\bigcup_{i=1}^{k} C = \{C_1, C_2, \cdots, C_N\} \quad (7)$$

$$C_i \neq \phi \; i = 1, 2, \cdots, k \quad (8)$$

$$C_i \cap C_j = \phi, i, j = 1, 2, 3, \cdots, k, i \neq j \quad (9)$$

The Euclidean distance calculation method between two data points X_i and Y_j is as follows:

$$d(X_i, Y_j) = \sqrt{\sum_{l=1}^{d}\left(x_{il} - x_{jl}\right)^2} \quad (10)$$

In the formula, $d(X_i, Y_j)$ represents the distance between data points X_i and Y_j, $x_{il} = x_{i1}, x_{i2}, \cdots, x_{id}$ and $x_{jl} = x_{j1}, x_{j2}, \cdots, x_{jd}$ represent data points with d-dimensional attributes, and d represents the dimension of the data points.

The objective function E adopts the sum of squared errors function as follows:

$$E = \sum_{i=1}^{k}\sum_{j=1}^{n_i}\left|X_{ij} - c_i\right|^2 \quad (11)$$

In the formula, n_i represents the number of data points in class i, and X_{ij} represents the j-th data point in class i.

The objective function E is a standard function represented by the sum of squared errors. The smaller the value of the objective function, the smaller the difference in data points within the class, and the greater the difference in data points between different classes, the better the clustering effect.

The specific steps to improve the K-means algorithm are as follows:

(1) Initialize the data point set X and cluster radius r;
(2) Run the canopy algorithm to obtain the number of clusters k;
(3) Calculate the average distance *AveDis* and data point density *density* between non-cluster-center data points;
(4) Select the data point with the highest density among all data points as the first initial clustering center;
(5) Select the data point with the highest density outside the cluster radius of the first initial cluster center as the second initial clustering center;
(6) To determine whether k cluster centers have been selected, proceed to step 7; otherwise, jump to step 3;
(7) Calculate the distance (Euclidean distance) between all data points in the data point set X and all initial cluster centers.

Then, based on the distance from the data points to each initial cluster center, assign the data points to the nearest cluster.

(8) Update all cluster centers, recalculate the mean of each cluster, and adjust the position of the point closest to the mean. The nearest point serves as the new cluster center;
(9) Continuously perform steps 7 and 8 until the positions of all cluster centers no longer change or reach the maximum number of iterations;
(10) Output the clustering results.

iii. Improved K-means algorithm for WSN clustering

The design of clustering routing for wireless sensor networks first requires reasonable clustering of nodes in WSNs.

Each node in a WSN can be regarded as a data point, and the number of clusters and the initial cluster head position can be improved, as mentioned earlier.

The K-means algorithm is used to calculate the clustering results, and the improved K-means algorithm can meet the clustering requirements of wireless sensor networks.

The specific design steps of the WSN clustering method based on the improved K-means algorithm are as follows:

(1) Input the wireless sensor network node set X and clustering radius r;
(2) Run the canopy algorithm to obtain the number of clusters k;
(3) Calculate the average distance *AveDis* and node density *density* between non-cluster-head nodes;
(4) Select the node with the highest node density as the cluster head;
(5) Select the node with the highest density outside the clustering radius of the cluster head as the next cluster head;
(6) Determine whether to select k cluster heads. If yes, proceed to step 7; otherwise, jump to step 3;
(7) Substitute the initial cluster head and the number of clusters into the K-means algorithm for clustering;
(8) Output the clustering results.

iv. Simulation verification

In order to test the performance of the improved K-means clustering algorithm proposed in this article, the MATLAB 2023a simulation platform was used for calculation.

We conducted simulation experiments using four typical datasets from UCI, namely Iris, Wine, Seeds, and Glass.

Using clustering accuracy as the evaluation criterion, the improved K-means algorithm and the traditional K-means algorithm were used to conduct ten experiments on four different datasets to obtain the clustering accuracy, and the accuracy of each cluster was evaluated.

The average was obtained, and the experimental results are shown in Table 2.

Table 2. Node mean square error of wireless sensor networks.

Number of Hidden Layer Nodes	Mean Squared Error of Training Set
4	0.030469
5	0.027156
6	0.02298
7	0.034099
8	0.018178
9	0.027816
10	0.018269
11	0.022957
12	0.020507
13	0.020523

From the Table 3, it can be seen that the improved K-means algorithm has higher accuracy, which is also the result of optimizing the selection of initial cluster centers.

Table 3. Comparison of clustering accuracy.

Dataset	Tradition K-Means	Improve K-Means
Iris	88.1%	91.2%
Wine	69.7%	71.1%
Seeds	90.6%	92.9
Glass	75.4%	78.9%

(3) Improving the Whale Algorithm to Determine WSN Cluster Heads

Due to the weak global exploration ability of the whale algorithm, it is prone to becoming stuck in local searching. Therefore, improvements should be made to the whale algorithm.

i. Basic Principles of the Whale Algorithm

The standard WOA simulates the unique search methods and trapping mechanisms of humpback whales, mainly including three important stages: trapping prey, bubble net predation, and searching for prey. The position of each humpback whale in the WOA represents a potential solution, and by continuously updating the whale's position in the solution space, the global optimal solution is ultimately obtained.

(a) Surround and hunt prey

$$D_{best} = \left| C \cdot X_{best}^{t} - X_{i}^{t} \right| \tag{12}$$

$$X_{i}^{t+1} = X_{best}^{t} - A \cdot D_{best} \tag{13}$$

$$A = (2 \times r - 1)a$$

In Equations (9) and (10), X_i^{t+1} represents the $t+1$-th iteration of the i-th individual, X_{best}^{t} represents the position of the optimal whale in the t-th iteration, D_{best} is the displacement of the optimal individual in the current t-th iteration, α decreases linearly from 2 to 0 with the number of iterations, $(0, 1)$ is a random number of C, and L is a random number uniformly distributed within $(0, 2)$.

(b) Bubble web predation

$$X(t+1) = D' \times e^{bl} \times \cos(2\pi l) + X^*(t) \tag{14}$$

$$D' = |X^*(t) - X(t)| \tag{15}$$

(c) Search for prey

$$D'' = |C \cdot X_{rand}(t) - X(t)| \tag{16}$$

$$X(t+1) = X_{rand}(t) - A \cdot D \tag{17}$$

ii. Improving the Whale Algorithm

To further improve the convergence speed and optimization accuracy of the standard whale algorithm, this paper adjusts the convergence factor a and contraction probability p in the whale algorithm to improve the convergence speed and optimization accuracy.

(1) Improvement in convergence factor: In the standard whale algorithm, the value of the convergence factor a decreases with the number of iterations, so the algorithm has a larger search range and stronger search ability in the early stage, and gradually tends toward the optimal solution in the later stage. Due to the linear decrease in the value of a, the algorithm may experience a slow convergence speed in the early stage, a small search range in the later stage, and falling into local optima, resulting in insufficient optimization accuracy.

The improvement method adjusts the convergence factor a to a nonlinear reduction formula:

$$\alpha = \alpha_0 - \frac{t}{t_{max}} e^{(a_1 - a_0)^l} \tag{18}$$

In the formula, α_0 is the starting value of a, a_1 is the final value of a, l is a constant, and t is the current number of iterations.

t_{max} is the maximum number of iterations.

(2) Improvement in contraction probability: in the standard whale algorithm, the p value is set to 0.5, which means surrounding prey and using gas.

The probability of bubble net predation is equal throughout the entire process. Due to the constant equality of the two probabilities, the algorithm may experience a slow convergence speed.

As the number of iterations gradually increases, the individual gradually approaches the optimal solution, and the probability of surrounding the prey should gradually be greater than the probability of net predation by using bubbles.

The improvement method in this article adjusts the contraction probability p to a linear reduction formula:

$$p_{surround} = 0.5 + t * u \tag{19}$$

$$p_{BubbleNet} = 1 - p_{surround} \tag{20}$$

In the formula, u is the adjustment parameter, and t is the current iteration number.

In summary, the specific implementation steps of the improved whale algorithm are as follows:

(1) Initialize the population and location information of whales, and set the maximum number of iterations t_{max};
(2) Calculate the fitness of each whale;
(3) Record the optimal whale position;
(4) If the termination condition is met, proceed to step 8; otherwise, proceed to step 5;
(5) Update parameters A,C,a,t;
(6) To determine if p is less than $0.5 + t * u$, proceed to step 7. Otherwise, use bubble nets for predation. Use the mathematical models in (12) and (13) to update the position and proceed to step 2;

(7) Determine if A is less than 1. If so, surround the prey and use the mathematical model to update the position. Go to step (2). Otherwise, search for prey and use the mathematical models in (16) and (17) to update the position. Go to step (2);
(8) Output the optimal solution.

iii. Determine the WSN cluster heads

The improved whale algorithm will be used for clustering, assuming that each whale contains the position of m cluster heads (m is the optimal number of cluster heads). The structure of each whale is as follows:

$$x_i = \{a_{i1}, a_{i2}, a_{i3}, \cdots, a_{im}\}$$

Among them, a_{ij} refers to the j-th cluster head vector of the i-th whale in cluster c_{ij}. Each whale represents a clustering method, and the entire whale population represents multiple clustering methods. The improved whale algorithm takes the $fitness$ function as the fitness function, takes its minimum value as the optimal value, and finds the position of the whale when $fitness$ takes the minimum value. At this time, the position of the m cluster heads in the optimal whale is the optimal clustering method.

The specific steps for WSN cluster head selection based on the improved whale algorithm are as follows:

(1) Initialize the whale population; the position of the i-th whale is as follows:

$$x_i = \{a_{i1}, a_{i2}, a_{i3}, \cdots, a_{im}\}$$

where m is the optimal number of cluster heads. Set the termination condition for the algorithm;
(2) Calculate the fitness value of each whale;
(3) Record the optimal individual and location;
(4) If the termination condition is met, proceed to step 6; otherwise, proceed to step 5;
(5) Update a and p based on the number of iterations;
(6) Update the whale position and proceed to step 2;
(7) Output the optimal solution and obtain the optimal cluster head.

Generate 200 two-dimensional data samples in Matlab2023a, randomly distributed in a square of 200 m $\times$ 200 m.

Within the scope, as shown in Figure 11, conduct ten control experiments on ten different sets of data, and compare them with the traditional K-means performance.

Compare the clustering performance of the LEACH-C and EEUC algorithms to verify the superiority of the improved K-means algorithm in this paper.

The black "*" in the figures below represents ordinary nodes, and the red "*" represents cluster heads.

Figures 12–15, respectively, show the traditional K-means, LEACH-C, EEUC, and improved algorithms.

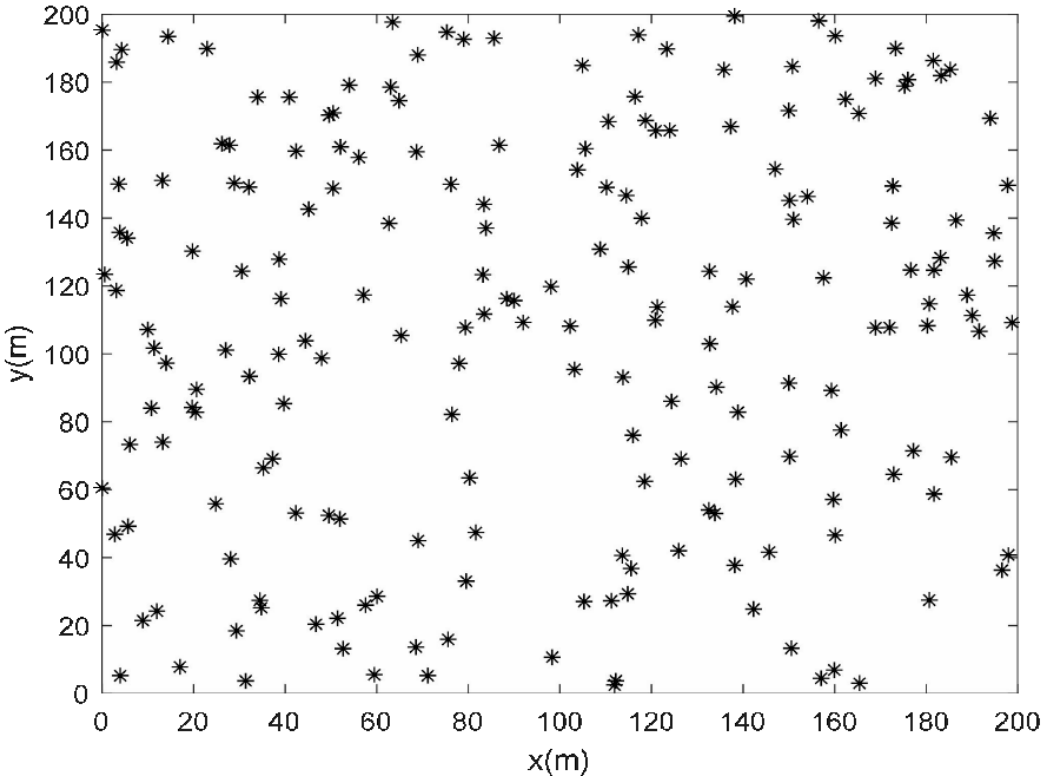


Figure 11. Scene diagram.

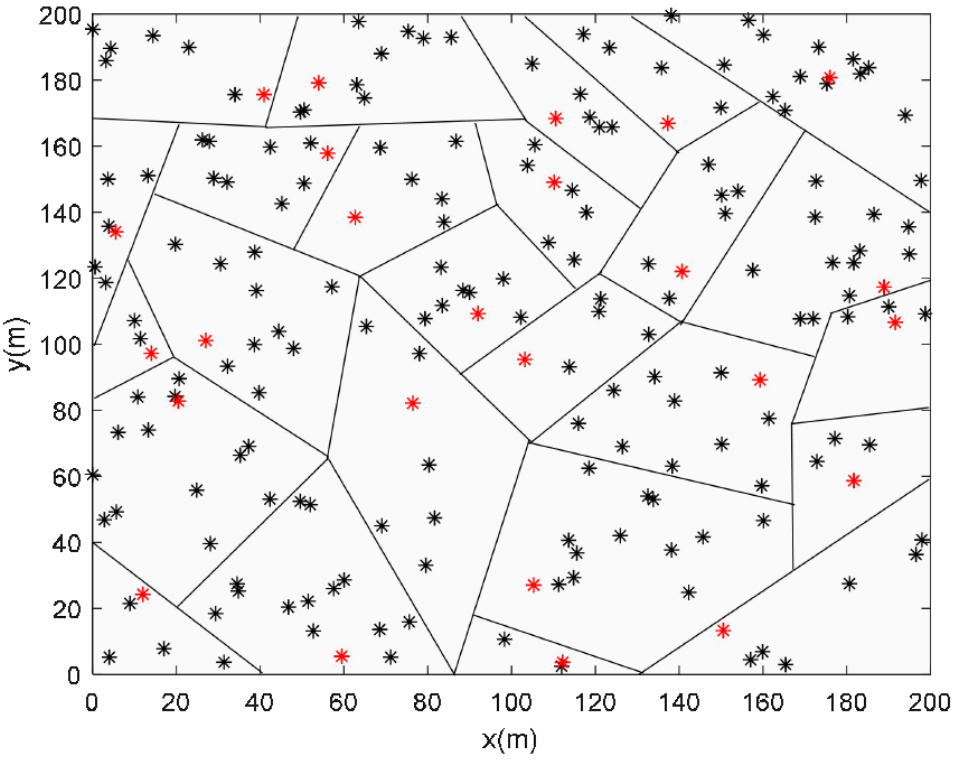


Figure 12. K-means clustering.

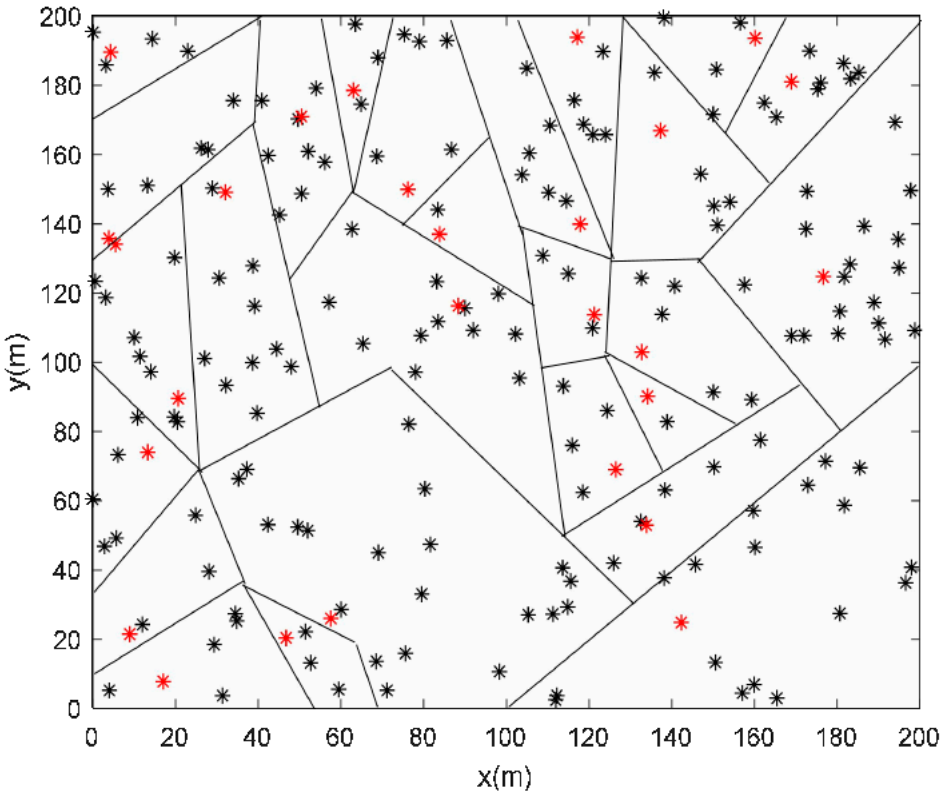


Figure 13. LEACH-C clustering.

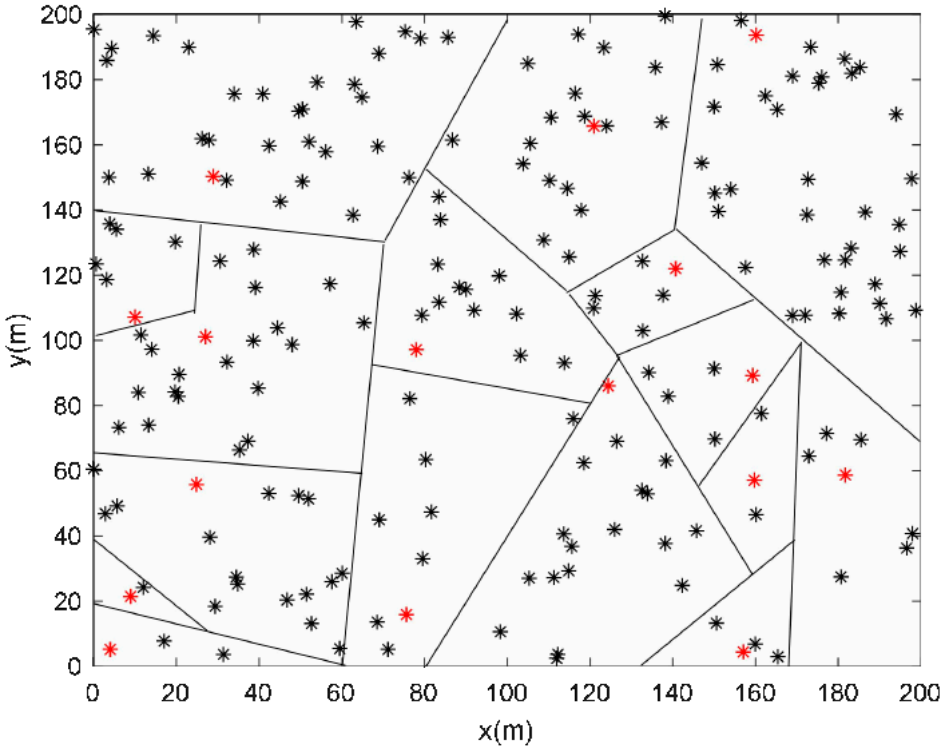


Figure 14. EEUC clustering.

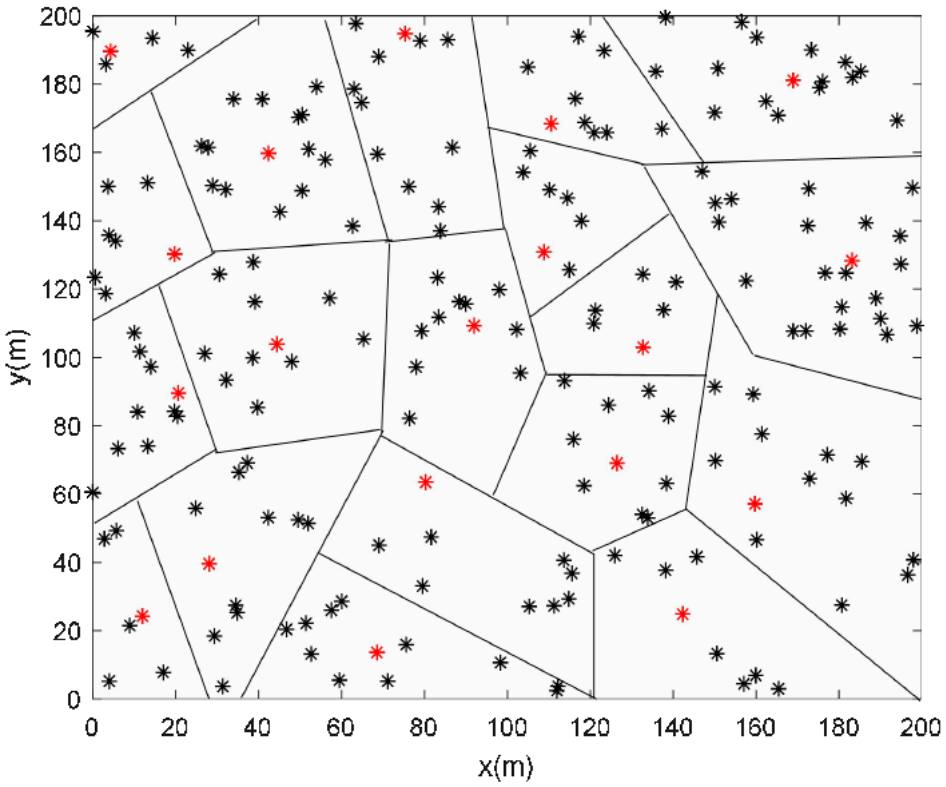


Figure 15. Improved K-means clustering.

The clustering effect of the K-means algorithm in experimental scenarios shows that the improved K-means algorithm has a better clustering effect than the traditional K-means, LEACH-C, and EEUC algorithms. This conclusion also demonstrates the universality of the improved K-means algorithm in optimizing the clustering effect of WSN networks.

2.5. BP Neural Network

A BP (backpropagation) neural network is a multi-layer feedforward neural network trained using the error backpropagation algorithm, and is the most widely used neural network. As shown in Figure 16:

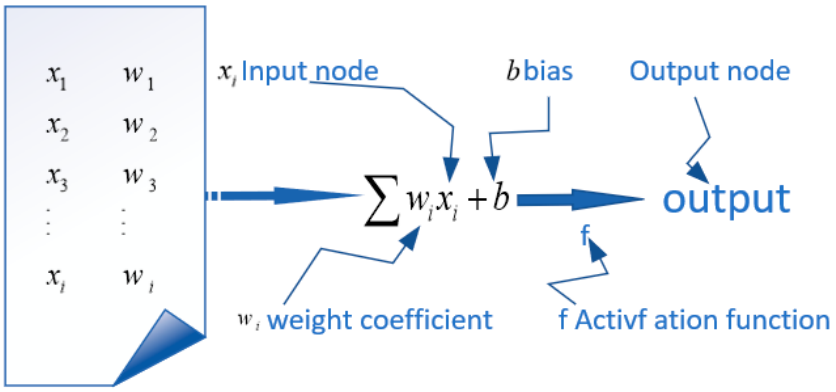


Figure 16. BP (backpropagation) neural network formula.

Perceptron—a single node in a BP neural network, consisting of input items, weights, biases, activation functions, and outputs.

Input node: $x_1, x_2, x_3, \cdots, x_i$;

Weight: $w_1, w_2, w_3, \cdots, w_i$;

Bias: b;

Activation function: f;
Output nodes: $outpu$.
A BP neural network is a typical nonlinear algorithm.

A BP neural network consists of an input hidden layer (also known as an intermediate layer) and an output layer, with one or more hidden layers. Each layer can have several nodes. The connection status of nodes between layers is reflected through weights.

There is only one hidden layer in traditional shallow neural networks. There are multiple hidden layers in neural networks for deep learning, as shown in Figure 17:

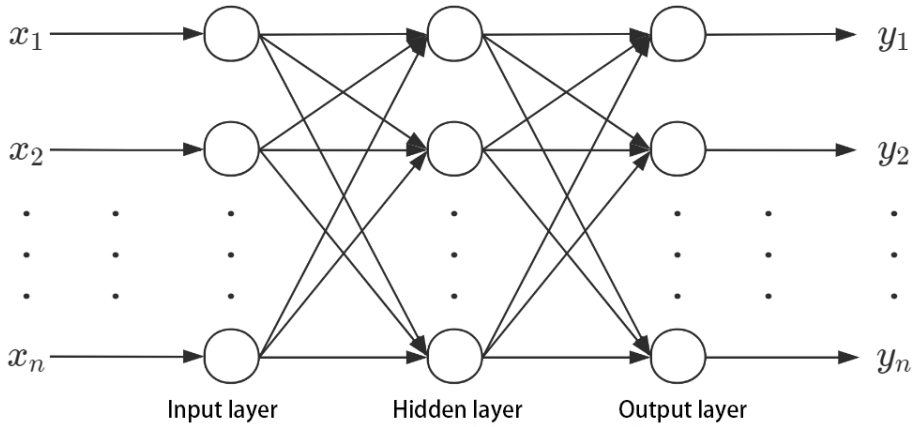


Figure 17. BP neural network three-layer structure.

Input Layer: The input layer is the starting layer of a neural network, responsible for receiving external input data or features. The number of neurons in the input layer usually corresponds to the number of features in the input data.

Hidden Layers: Located between the input and output layers, hidden layers are a crucial part of neural networks for processing information. The hidden layer can have one or more layers, each consisting of a certain number of neurons. Neurons are connected through weights and biases, and nonlinearity is introduced through activation functions.

Output Layer: The output layer is the last layer of a neural network, responsible for generating the final output of the network. The number of neurons in the output layer usually corresponds to the number of target variables that need to be predicted. The output layer usually uses the softmax function to convert the output into a probability distribution. For regression problems, the output layer may directly output predicted values.

Activation Function: Each neuron in the hidden layer uses an activation function to process its input. Common activation functions include the sigmoid function, ReLU (Corrected Linear Unit) function, Tanh function, etc. The choice of activation function has a significant impact on the performance of the network, as they determine how neurons respond to their inputs.

Weights and Biases: Weights and biases are learnable parameters in neural networks used to control the strength of connections and bias terms between neurons. During the training process, these parameters are updated through backpropagation algorithms to minimize the prediction error of the network. The weights and biases are derived from the solution space position and fitness values of the whale algorithm.

Backpropagation Algorithm: The backpropagation algorithm is the core algorithm of a BP neural network used to calculate the prediction error of the network and update weights and bias parameters. It adjusts the parameters of the network layer by layer through the backpropagation of error signals, making the output of the network gradually approach the true value.

In the process of forward propagation, there is a step that compares whether the expected result is satisfactory or not. In this step, there is an error between the actual output result and the expected output result. In order to reduce this error, this is converted into an

optimization process. For any optimization problem, there is always an objective function, which is the loss function.

The purpose of repeated training is to find the minimum value of this function in order to minimize the error between the actual output result and the expected output result. Each iteration will generate a weight update, and then, the updated weights will be propagated forward with the training samples. If the results are not satisfactory, backpropagation will be performed and the iteration will continue. This process is repeated until a satisfactory result is obtained.

As shown in Figure 18, a BP neural network is used to calculate the global optimum, not the local optimum. Ant colony algorithm optimization is used to calculate the global optimum.

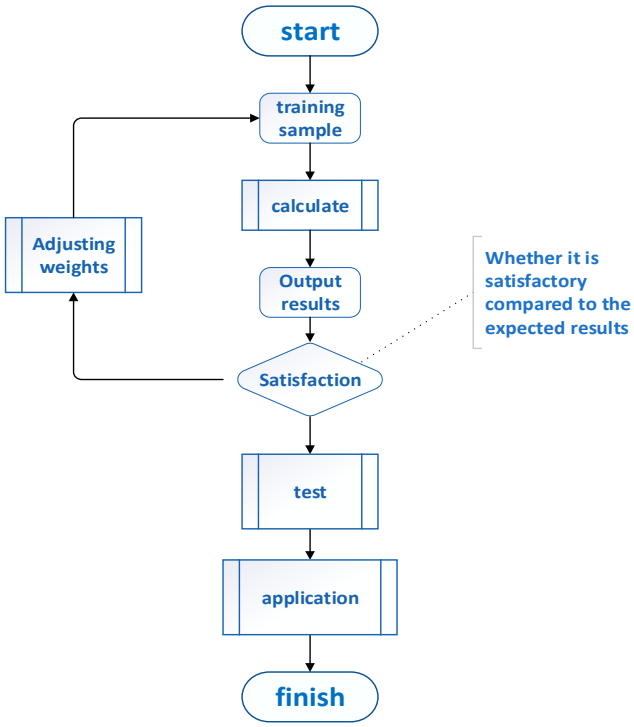


Figure 18. BP calculation flowchart.

The whale algorithm (WOA) combined with a BP neural network model is used to calculate the global optimum, as shown in Figure 19:

2.6. WOA-BP Algorithm

The WOA-BP neural network model is prone to convergence to local optima due to the randomness of the initial weights and thresholds in traditional BP neural networks, which affects the final prediction performance. To solve the above problems, the WOA is used to improve the BP neural network. The specific process is as follows.

(1) Collect the data obtained from numerical models, divide the dataset, and normalize it.
(2) Initialize the BP neural network, set the training frequency, learning rate, and learning accuracy, and determine the neural network structure.
(3) Initialize the WOA parameters and use the mean squared error of the training and testing samples as the fitness function.
(4) Calculate the fitness value of whales using Formulas (12) and (13) and record the optimal individual position.
(5) Use Formulas (16) and (17) to choose different ways to update the whale position.
(6) Determine whether the termination conditions are met. If the maximum number of iterations is reached, the optimal initial weights and thresholds are output. Otherwise, repeat steps (4)–(5).

(7) Assign the optimal initial weights and thresholds to the BP neural network.
(8) Calculate the output error of the BP neural network.
(9) Determine whether the termination conditions are met. If the output error reaches the preset learning accuracy or maximum training frequency, the prediction result can be output after reverse normalization processing. Otherwise, update the weights and thresholds first, and then repeat steps (8)–(9).

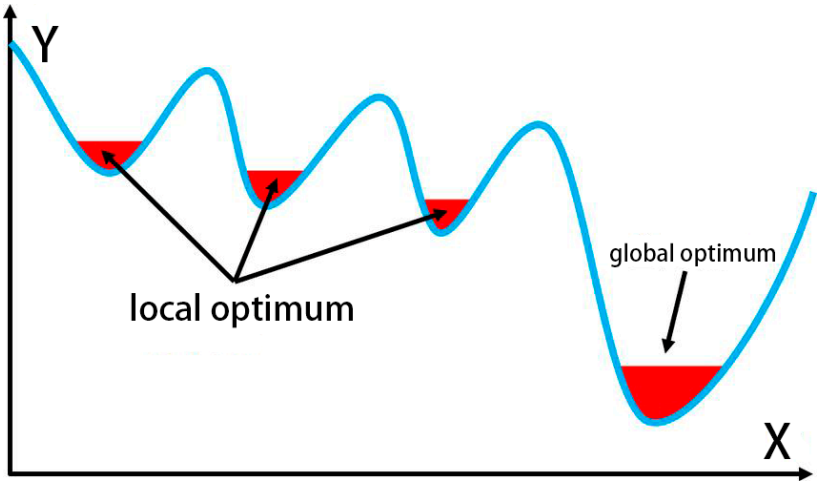


Figure 19. Global optimum graph.

The process of improving the BP neural network using the WOA algorithm is shown in Figure 20.

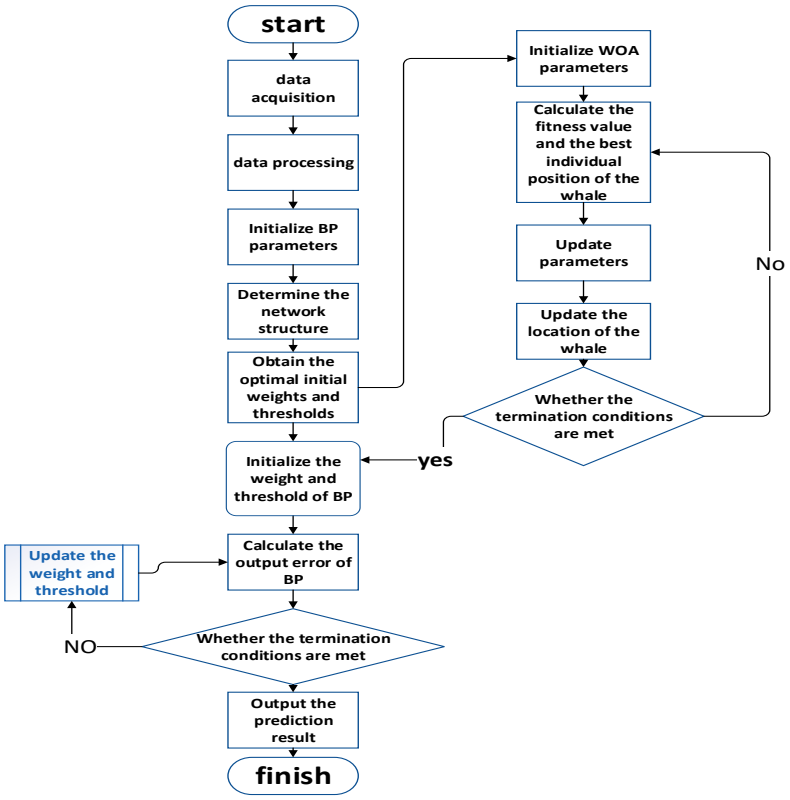


Figure 20. WOA-BP.

The simulation data come from the sensor battery health status dataset, which is a multi-input dataset and simulated using the MATLAB R2023b tool.

From the Figure 21, it can be seen that as the number of training sessions increases, the error becomes smaller and smaller.

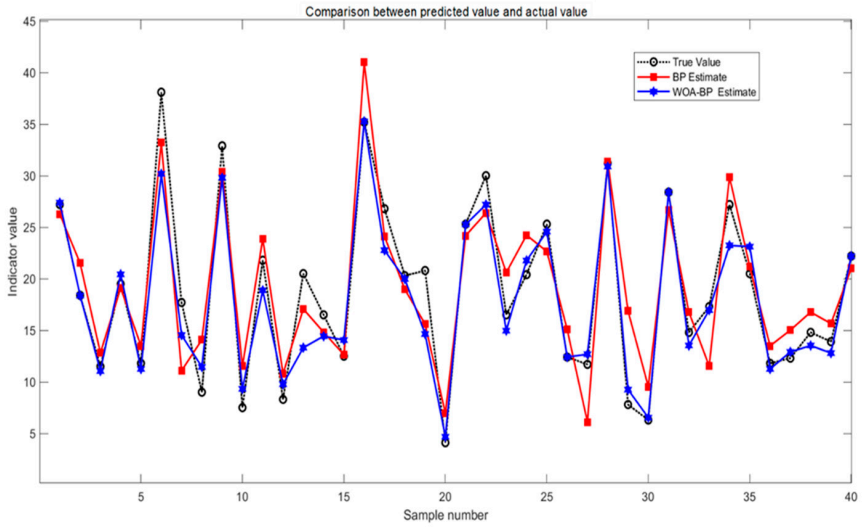


Figure 21. Comparison between predicted value and actual value.

From the Figure 22, it can be seen that the maximum error is 8.688 and the sample size is 1.

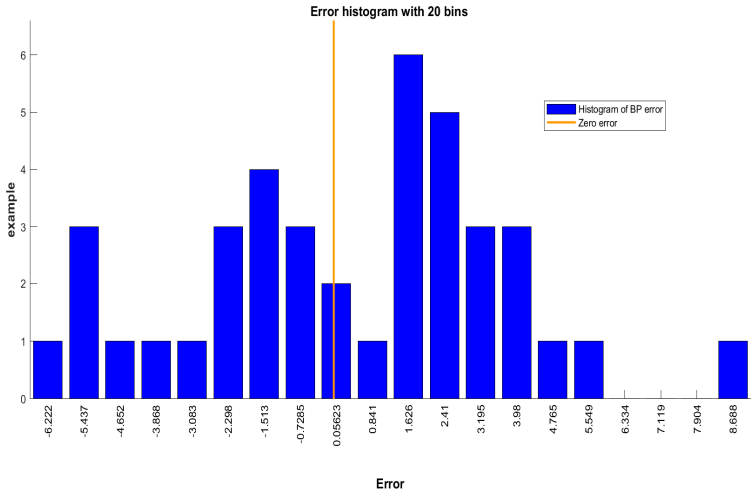


Figure 22. BP error histogram with 20 bins.

From the Figure 23, it can be seen that the maximum error is 2.352 and the sample size is 6.

We can then determine the optimal hidden layer node.

As shown in Figure 23, The optimal number of hidden layer nodes is 8, with a mean squared error of 0.018178.

In Table 4, a comparison of errors between BP neural network and WOA-BP neural network is made from five dimensions. The average absolute error of BP Neural Network is 2.9143, which is greater than the average absolute error of WOA-BP Neural Network of 1.7029; The mean square error of BP Neural Network is 12.1307, which is greater than the mean square error of WOA-BP Neural Network, which is 6.5033; The root mean square error of BP Neural Network is 3.4829, which is greater than the root mean square error of WOA-BP Neural Network of 2.5501; The average absolute percentage error of BP Neural

Network is 21.3418%, which is greater than the average absolute percentage error of WOA-BP Neural Network of 9.7347%; The prediction accuracy of BP Neural Network is 78.6582%, which is lower than the prediction accuracy of WOA-BP Neural Network of 90.2653%. Therefore, the accuracy of WOA-BP Neural Network is 11.6071% higher than that of BP Neural Network.

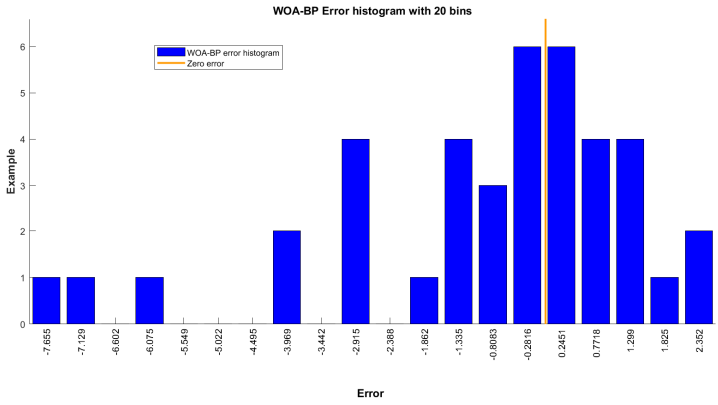


Figure 23. WOA-BP has an error histogram with 20 bins.

Table 4. Comparison between BP neural network and WOA-BP neural network.

Error Term	BP Neural Network	WOA-BP Neural Network
Mean absolute error (mae)	2.9143	1.7029
Mean squared error (mse)	12.1307	6.5033
Root mean squared error (rmse)	3.4829	2.5501
Average absolute percentage error (mape)	21.3418%	9.7347%
Prediction accuracy	78.6582%	90.2653%

3. Results

Performance evaluation:

Initialize 50 SNs, initialize the data for transmission distance, node MAC address, and battery energy, and use the ant colony algorithm to find the best route for transmitting data, as shown in Figure 24:

In Figure 24, each node is a wireless mobile sensor consisting of multiple nodes to form a topology diagram. Each node contains information such as the transmission distance of the network topology, node MAC address, battery level, and communication link. The diagram shows the network structure models of 20 wireless mobile sensors.

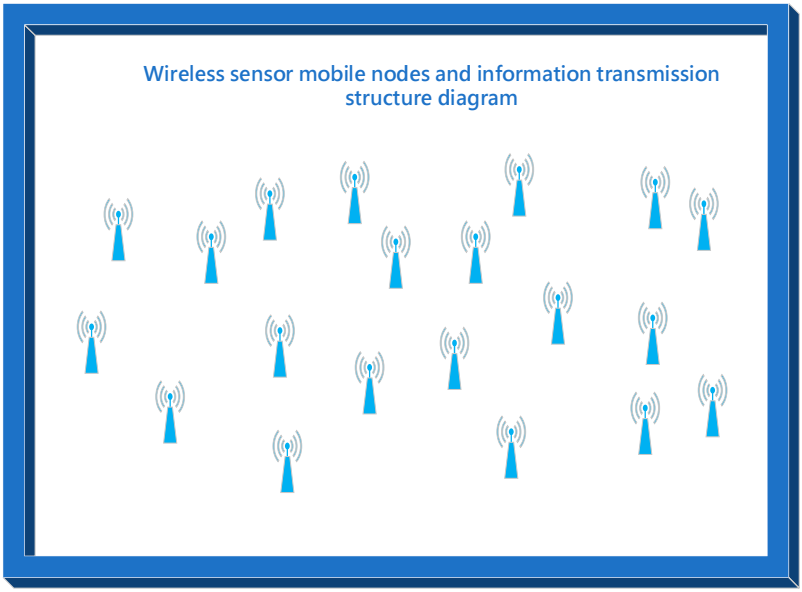


Figure 24. Wireless sensor mobile node and information transmission structure diagram.

Figure 25: Wireless sensor mobile node and information transmission structure diagram.

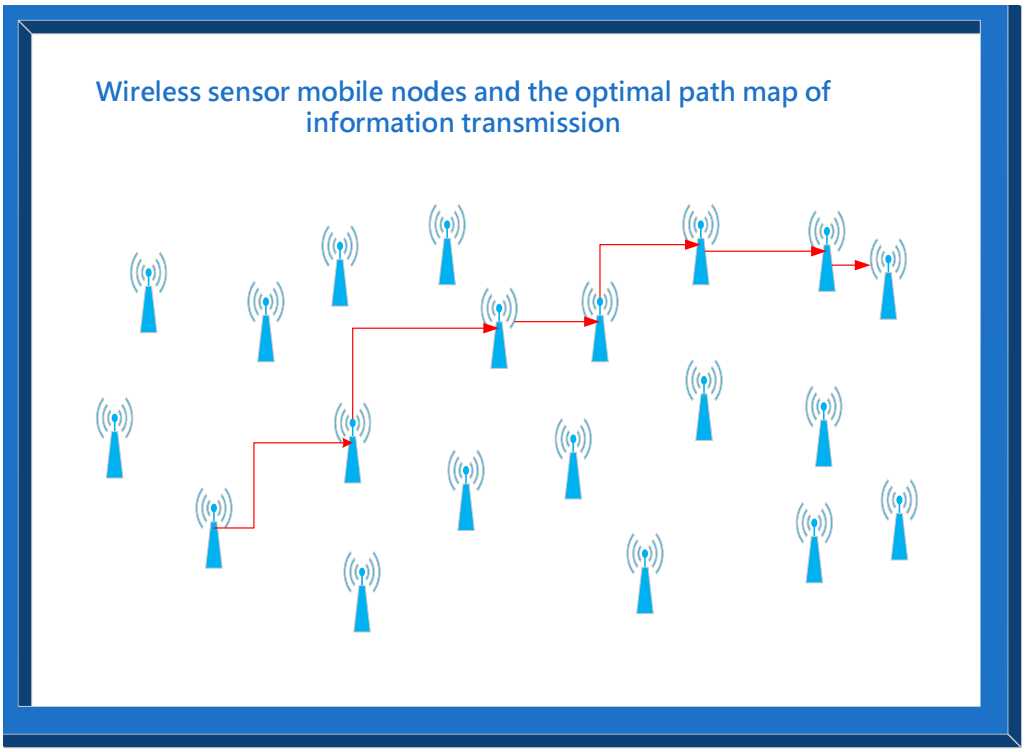


Figure 25. Wireless sensor mobile nodes and optimal path map of information transmission.

The red arrows in Figure 25 represent path planning in the wireless sensor, Figure 25 simulates each wireless sensor node ant using the ACOD. Each ant searches for food in the topology diagram. If there is no food, it will check for pheromones. Each ant has a probability of making mistakes and moving toward more pheromones. If there are no pheromones, it will continue to move in its original direction, accompanied by random interference and memory. If an obstacle blocks the direction of each ant's movement, the ant will randomly choose another direction. The closer to the food it moves, the more

pheromones are sown, and the less pheromones are kept away from the food. We can use the ant colony algorithm to find the best route for transmitting data.

Through 1927 computational iterations, the optimal path for transmitting data was found in Figure 26. The best routing result graph is shown in [31].

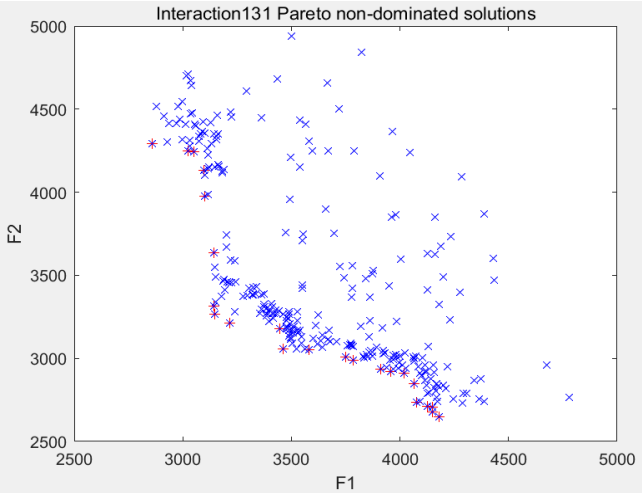


Figure 26. Using the ant colony algorithm to calculate the optimal transmission path graph.

Each point in Figure 26 represents a wireless sensor node, with a total of 300 nodes in the matrix and an activity range of 5000 × 5000. For ease of display, the cutting range is 2500 × 5000 on the horizontal axis and 2500 × 5000 on the vertical axis. Each h point in the figure displays the transmission time and distance of the wireless sensor node, and calculates the optimal path using the ACOD. The 131st calculation produced the Figure 26.

The movement range of 300 wireless SNs in Figure 27 is a matrix of 5000 × 5000. For ease of display, the cutting range is as follows: the horizontal axis is 1800~3600 and the vertical axis is 1500 × 4500. Using the ACOD, we can calculate the optimal transmission path through 500 iterations, as shown in Figure 27, Each "*" in the figure represents a wireless sensor node, and a line connected by several "*" is represented as the optimal transmission path.

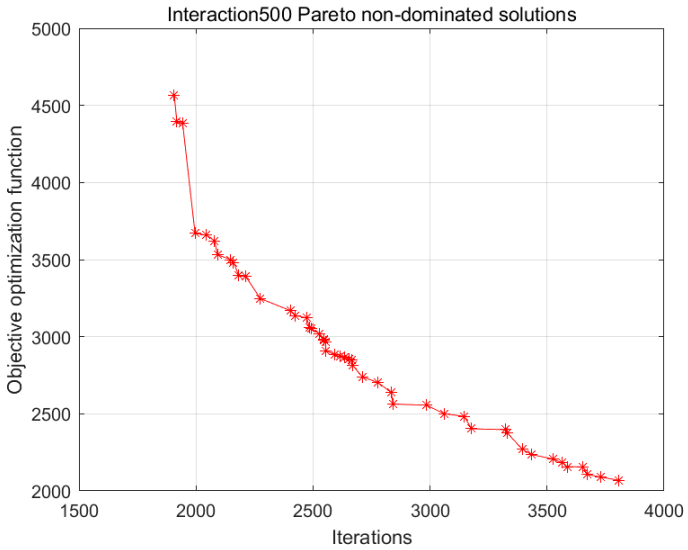


Figure 27. Calculating 500 times using the ant colony algorithm to form the optimal path map.

As shown in Figure 28, Based on the data generated by the whale optimization algorithm (WOA), we input these into a backpropagation (BP) network to construct the following network:

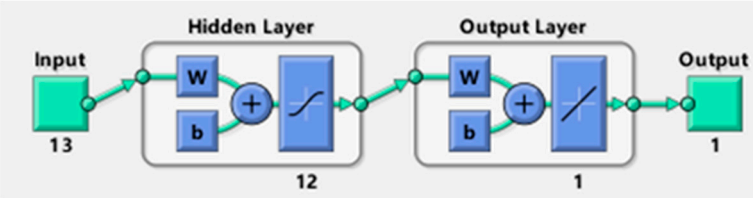


Figure 28. Combining BP neural network with whale optimization algorithm (WOA).

As shown in Table 5, The number of hidden layer nodes is 13, and only 10 are listed in the table. The smaller the mean squared error of the training set, the higher the accuracy. Among the 4 to 13 hidden layer nodes, the 12th hidden layer node has the highest accuracy, with a mean squared error of 0.015. In addition, the 13th has a mean squared error of 0.027, indicating a decrease in accuracy.

Table 5. Determining the optimal hidden layer node.

Serial Number	Number of Hidden Layer Nodes	Mean Squared Error of Training Set
1	4	0.032
2	5	0.024
3	6	0.026
4	7	0.027
5	8	0.032
6	9	0.022
7	10	0.032
8	11	0.027
9	12	0.015
10	13	0.027

Figure 29 uses the whale optimization algorithm (WOA), with the horizontal axis representing the number of iterations and range, and the vertical axis representing the mean squared error and range. As the number of iterations increases, the mean squared error gradually decreases and the accuracy gradually improves. Figure 7 shows that the number of iterations for the mean squared error significantly decreases, while the change in the mean squared error is minimal.

The data generated based on the whale optimization algorithm (WOA) are input into a backpropagation (BP) neural network to predict the wireless SN.

In Figure 30, red represents the BP prediction error, green represents the WOA-BP prediction error, the vertical axis represents the range of prediction error values, and the horizontal axis represents the sample size and value range. From the graph, it can be seen that the red BP has the highest prediction error value, while the green WOA-BP has a smaller prediction error.

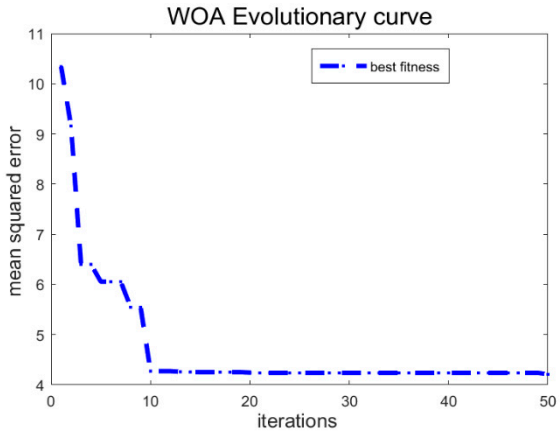


Figure 29. Using whale optimization algorithm (WOA) to reduce mean squared error.

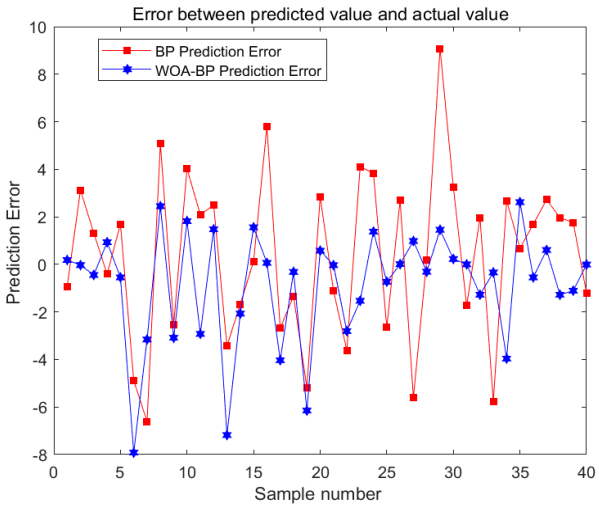


Figure 30. Comparison of BP prediction error and WOA-BP prediction error.

In Figure 31, * stands for multiplication, the circle in Figure 31 represents the real data, the dashed line represents the real regression curve, and the blue line represents the fitted curve after BP training. R is the regression correlation coefficient, representing the correlation between the predicted output and the target output. The closer the R value is to 1, the closer the relationship between the prediction and the output data, and the higher the accuracy. The smaller the value, the poorer the proximity and accuracy between the predicted and output data. R = 0.90877 in Figure 32.

In the WOA-BP regression plot, $R = 0.96009$, indicating that the accuracy of the WOA-BP algorithm is higher than that of the BP algorithm.

After 1000 cycles, the BP prediction accuracy is 81.3735%, and the WOA-BP prediction accuracy is 90.2353%.

Therefore, this improved whale algorithm enhances the node search capability, accelerates the global search speed, and optimizes network performance by increasing the node coverage in wireless sensor networks. This also solves the major problems that disrupt the overall efficiency of wireless sensor networks, such as low energy efficiency and a shortened network lifespan.

In addition, we adopted the ACOD to establish the optimal movement trajectory for the moving convergence point, intelligently considering the distance between CHs. The enhancement in the distance heuristic factor generated thereby avoids local optima, accelerates convergence speed, improves network performance over time, and reduces the latency of Sink data retrieval.

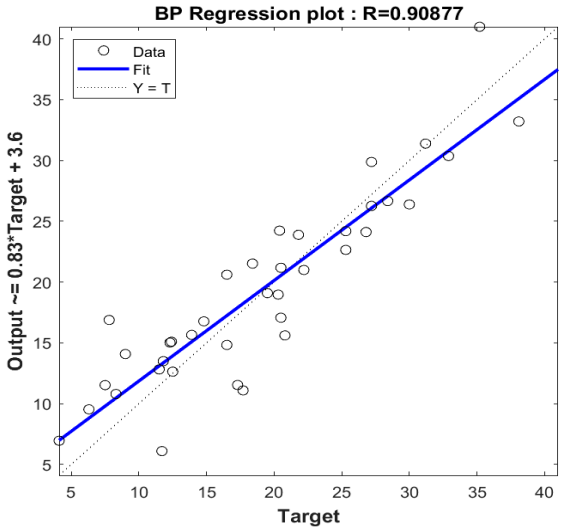


Figure 31. BP regression plot.

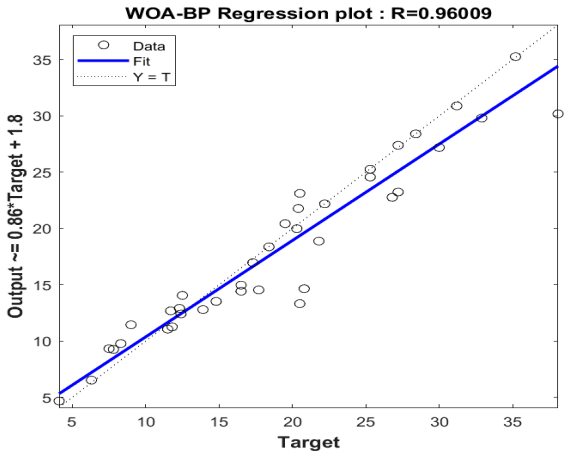


Figure 32. WOA-BP regression plot.

Finally, in order to effectively compare and evaluate the performance level of our proposed model and existing multi-objective evolutionary algorithms (MOEAs), we adopted two evaluation metrics, namely the hyper volume (HV) metric and the Epsilon (ε) metric. The HV index evaluates the convergence and diversity of the obtained solutions, while the Epsilon index measures the minimum transformation factor required to ensure non-dominance. In both cases, our proposed model exhibits superior performance, highlighting its potential in optimizing wireless sensor networks.

A. Simulation settings

For analysis, we simulated a WSN (WSN) consisting of N SNs (designated as CHs) and a base station. Considering the actual usage scenario, we assume that all nodes are fixed after deployment and their positions remain unknown. Each node is assigned a unique ID, with ID 0 assigned to the BS.

In order to obtain the most realistic simulation results, we followed the EC model based on specific features of the Chipcon CC2420 radio transceiver data table. The total energy consumption E (Sn) of any sensor was calculated by adding up the energy consumed by the sensor during different operating states (i.e., sleep, reception, or transmission) and the energy consumed during transitions between these states. In these calculations, the corresponding time spent in these states and the number of transitions for each sensor were considered.

The entire simulation experiment was divided into two main parts. Firstly, the effectiveness of the standard whale optimization algorithm in solving the WSN coverage problem was evaluated. Subsequently, in an identical simulation environment, the effectiveness of the enhanced whale optimization algorithm was measured based on the previously tested basic version. This two-stage simulation process is crucial for ensuring a comprehensive understanding of the comparative advantages of each algorithm.

The parameters for the experiment have been carefully selected to accurately reflect common challenges and constraints in real-world scenarios. Specifically, we carefully considered factors such as the average EC of each node, the throughput of the base station, and execution time when setting up the simulation.

By incorporating these considerations into our performance evaluation strategy, our goal is to create a highly credible simulation setup that is highly relevant to the actual deployment of WSNs. The ultimate goal is to effectively represent the impact and potential of the proposed optimization strategy on WSN performance using this setting.

B. Compare network performance indicators

To quantify the performance of the proposed network optimization strategy, it is necessary to evaluate several important network metrics [32]. One of the most important indicators is network lifespan. It is defined as the point at which the first sensor node in the network stops running due to energy depletion [33]. Comparing our method with the network lifetime of standard ant colony optimization and mobile aggregation ant colony optimization M proves the advantages of our model. Specifically, our network did not begin node depletion until after approximately 2600 rounds, while ant colony optimization M began node depletion after 2330 rounds. This indicates that our proposed algorithm reduces the energy consumption of CHs by using mobile aggregation nodes to directly communicate with CHs within a short communication range, thereby greatly extending the lifespan of the network [34].

In addition, the proposed method exhibits enhanced performance on different numbers of nodes, as demonstrated by the extensive simulation results. These indicate that our network optimization strategy always performs better than ant colony optimization M by improving the network lifespan and reducing the total energy consumption [35].

The final key performance indicator involves determining the optimal trajectory for the mobile convergence point through the ACOD. Traditional ACOA allows ants to deviate from the target over time. Our model enhances the global search ability, improves the convergence speed, and avoids becoming trapped in the local optimal solution by considering the distance between CHs. By doing so, our model reduces latency, enhances data collection, and thus improves the overall network performance of wireless sensor networks.

C. Performance evaluation of proposed optimization strategies

Adopting the ant colony optimization algorithm and further considering heuristic distance factors can improve the global search ability, avoid becoming stuck in local optimal solutions, and improve the convergence speed. Therefore, compared with traditional ant colony optimization models, the proposed model effectively solves the problem of low energy efficiency and extends the lifespan of the network.

The K-means clustering method is used to determine the optimal configuration of network elements, including the number of CHs, the clustering nodes, and the link quality between the CM and CHs. In addition, it maintains a balanced use of resources and extends the lifespan of the network.

The improved whale algorithm uses a BP neural network to optimize the initial population distribution, thereby accelerating global search and subsequently increasing node coverage. It effectively alleviates hot issues and improves the overall performance of wireless sensor networks.

Meanwhile, monitoring the remaining energy of nodes and the direct communication between mobile aggregation nodes and CHs within a short communication range can help reduce EC.

Our simulation results indicate that our proposed performance optimization strategy provides an energy-saving, robust, and scalable solution, significantly optimizing network performance by reducing average EC, improving base station throughput, and minimizing execution time compared to traditional methods. Therefore, its effectiveness in maximizing the performance of wireless sensor networks has been validated.

The following section can be divided into subheadings. It will provide a concise description of the experimental results, their explanations, and the conclusions that can be drawn.

4. Discussion

In this section, the results obtained from the proposed model will be evaluated and compared with other relevant models, and the impact of the proposed model on WSNs will be discussed.

Information transmission between wireless sensor nodes is affected by power supply, quality, and terrain, which are based on probability, dispersion, and liquidity; the proposed model has strong global search ability, is not continuous, and is not based on location. The following is shown in Table 6:

Table 6. Optimization algorithm feature comparison.

Feature	Particle Swarm Optimization	Ant Colony
Inspiration source	A school of birds or fish	Ants foraging for food
Information transmission mechanism	Location and speed information	Pheromone
Search method	Based on location	Based on probability
Mathematical model	Simple	Complex
Applicable problem types	Continuous optimization	Discrete optimization
Convergence	Easy to fall into local optima	Has a strong global search ability
Computational complexity	Low	High

Compared with traditional models, the test results of the proposed model show a significant improvement in network performance. This improvement stems from the integration of ant colony optimization-based routing, K-means clustering, and the improved whale algorithm to achieve maximum coverage. Observations indicate that using ant colony optimization-based routing can significantly reduce the travel distance and energy consumption of mobile aggregation nodes. In addition, the application of the ACOD helps to determine the most effective network layout, thereby improving the robustness and lifespan of the network.

By comparing the proposed model with traditional methods, it is evident that our model outperforms the latter in all aspects. A traditional WSN is often affected by issues related to low energy efficiency, a shortened network lifespan, and hotspots. The proposed model solves these problems through energy-saving routing, optimal node layout, an efficient data transmission mode, and the improved whale algorithm to achieve better network coverage.

In terms of impact, the proposed model provides a promising approach for efficient management of WSNs. This model can address the common challenges in this field and can be widely implemented to improve overall network performance. If applied properly, this WSN-based performance optimization integrated model can make progress in military operations, environmental monitoring, public safety, healthcare, and other

fields. The potential of this model lies in its ability to meet the growing demand for more efficient and reliable wide-area sensor networks. In addition, the scalability and EE of the proposed model can extend the lifespan of WSNs, making it suitable for long-term on-site deployment.

Therefore, it has broad application prospects in improving the efficiency and expected lifespan of wireless sensor networks. Future work can focus on further improving algorithms and application technologies to further enhance network performance and adaptability to different network conditions.

A. Interpretation of evaluation results

The evaluation results obtained from the simulation of our proposed model confirm the hypothesis that integrating ant colony optimization and advanced whale algorithms into WSNs can significantly optimize network performance.

In our simulation environment, we set parameters related to the proposed algorithm and ant colony optimization M algorithm for comparative analysis. We uniformly deployed all SNs and aggregation nodes in a $2 \times 200 \times 200$ m^3 area and conducted multiple simulations to determine the appropriate values for parameter changes. The results indicate that our improved ACOD always performs better than its corresponding algorithm, ant colony optimization M.

An important indicator used in network performance evaluation is network lifespan, defined as the duration of time before the first node runs out of energy and stops running. Our model indicates that there is a significant improvement in extending the network lifespan due to the direct communication between mobile convergence points and CHs over relatively short distances.

In terms of energy utilization, our algorithm selects the node with the highest remaining energy as the CH. The CH will only rotate when its remaining energy is below the preset energy threshold. The EC rate in our scheme has always been lower than that of ant colony optimization M, indicating that the energy performance in WSNs has been enhanced.

In addition, we have deployed the ACOD to determine the optimal movement trajectory of the moving convergence point. Our improved ant colony optimization algorithm is different from the traditional algorithm. The traditional algorithm usually makes the ants move to the next node according to the current information, but it will deviate from the target over time. By considering the distance between CHs to improve the distance heuristic factor, we have improved the global search ability, avoided becoming stuck in local optima, and improved the convergence speed.

Finally, our method significantly enhances the data collection performance of WSNs. Compared to ant colony optimization M, the amount of data collected within a given time range is higher. In summary, the simulation results provide convincing evidence that our proposed model can significantly improve the performance of WSNs in multiple aspects.

Finally, the comparison with the whale swarm optimization algorithm shows that the proposed model has the superior ability to improve network coverage with fewer iterations, which helps to optimize the overall network performance.

In summary, the proposed model combines ant colony optimization algorithm-based routing coverage maximization with an enhanced whale algorithm and exhibits robust comparative performance compared to the well-known WSN protocol. It provides an energy-saving, reliable, and scalable solution for WSN performance optimization, outperforming existing methods in terms of average EC, base station throughput, and execution time per node.

B. The meaning of the proposed model

The impact of the proposed model is multifaceted, including both theoretical and practical aspects of wireless sensor networks.

Firstly, the model indicates that the EC in WSNs can be significantly optimized. The proposed ant colony optimization routing model proposes an energy-saving mechanism. It

maintains the number of active nodes at the lowest level by carefully selecting the CH, and follows an optimized trajectory for moving aggregation nodes.

In addition, the model also demonstrates the possibility of improving network data reliability by establishing dedicated routing trees. It optimizes the total throughput of the base station, thereby enhancing the consistency of data transmission. This is particularly important in applications such as infrastructure monitoring, industrial sensing, and telemetry, where correct and timely data transmission can have a significant impact.

The model also demonstrates the efficiency of the integrated whale algorithm in maximizing coverage. By optimizing the initial population distribution and accelerating the global search, the whale algorithm ensures a wide-area coverage of nodes, which is an ideal feature in wireless sensor networks.

Finally, the proposed model emphasizes the crucial role of effective clustering mechanisms in wireless sensor networks. It can be inferred from the ant colony algorithm that determining the number of CHs, the clustering nodes, and the link quality between the CM and CHs can significantly improve network performance.

In summary, the successful implementation of the proposed model can significantly improve the performance and lifespan of wireless sensor networks, providing a new avenue for research in this field.

A comparison with other WSN algorithms is shown below to Table 7.

Table 7. Simulation of main parameters.

Parameter	Parameter Values
Network area	600 m × 600 m
Number of nodes	360
Packet length	256 byte
Node initial energy	10 J
Energy consumption of data fusion	5 nJ/bit
Communication channel bandwidth	0.5 Mb/S, 1 Mb/S, 1.5 Mb/S, 2 Mb/S,

As shown in Figure 33, as the data transmission rate gradually increases, the network survival time also decreases. However, it can be seen that in all three cases, the improved algorithm has made some progress compared to the original algorithm. When the average network rate is 1 Mbps, the WOA-BP algorithm has the same effect as the LEACH algorithm and the EEUC algorithm. From the simulation experiment results, it can be seen that with a network speed greater than 1.5 Mbps, the WOA-BP algorithm has a short transmission time and a fast transmission rate.

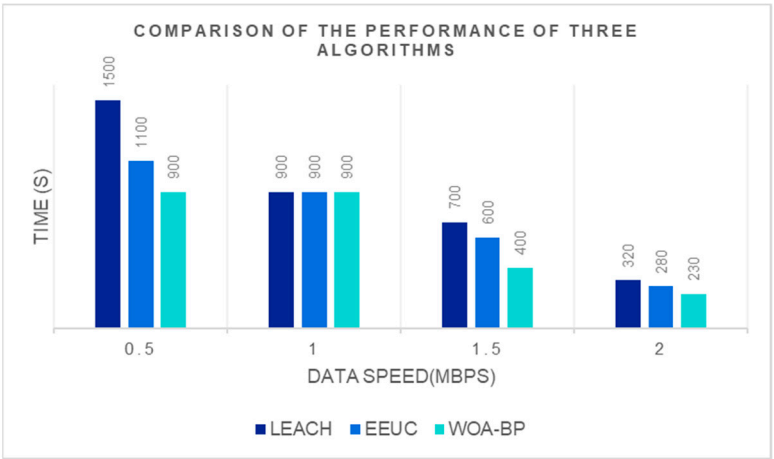


Figure 33. Comparison of performance of three algorithms.

From the above Figure 34, it can be seen that at a runtime of 900, the LEACH algorithm and the EEUC algorithm have a wireless sensor node death rate of 80–100, while the optimized WOA-BP algorithm has a node death rate of 10.

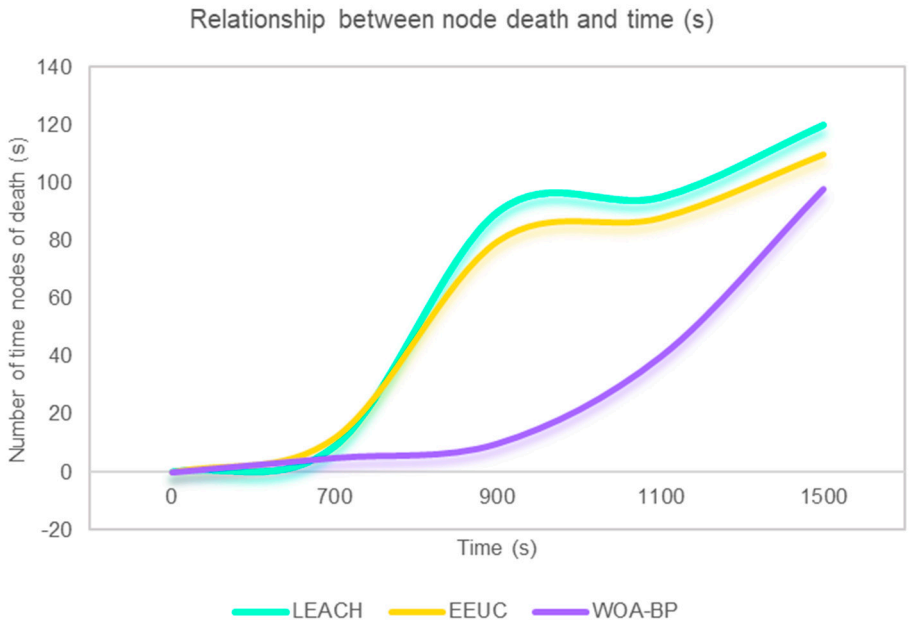


Figure 34. The above figure shows the relationship between node death and time.

5. Conclusions

In the extensive research presented in this article, we propose a comprehensive approach to improve various aspects of network optimization in WSNs. This optimization model revolves around ant colony optimization of routing and the integration of the improved whale algorithm to maximize coverage range.

Through the ant colony optimization method (ACOD), we have adopted a centralized multi-objective solution that is highly effective for clustering and routing problems in wireless sensor networks. For most test cases, this method has better scalability than previous models, mainly due to the use of a larger number of CHs to cover the entire network.

We use the ACOD to optimize routing. This algorithm aims to significantly reduce the average EC of each node and improve the throughput of the base station. Our research findings indicate that reducing the active nodes and limiting communication between clusters within CHs can significantly improve the EE in WSNs. Mobile aggregation nodes collect routing data along the optimal trajectory determined by the ACOD, providing an effective routing method.

When using the K-means algorithm for WSN (wireless sensor network) clustering, the following conclusions can be drawn based on the experiments:

Clustering effect: The K-means algorithm can effectively divide the nodes in a WSN into several clusters, with the center of each cluster determined by the average value of the nodes within that cluster. This clustering method helps to organize the network structure and improve data transmission efficiency and energy utilization efficiency.

Energy optimization: Through reasonable clustering, the communication distance between nodes can be reduced, thereby reducing energy consumption. In WSNs, communication energy consumption is directly proportional to the square of distance, so optimizing clustering through the K-means algorithm can significantly reduce unnecessary long-distance communication and prolong the network lifespan.

Local optimal problem: The K-means algorithm may fall into a local optimal solution, which means the algorithm may not be able to find the globally optimal clustering method.

This is usually related to the selection of initial cluster centers, as different initial cluster centers may lead to completely different clustering results.

Sensitivity to K value: The performance of the K-means algorithm highly depends on the preset number of clusters K. In WSNs, selecting the appropriate K value is crucial for balancing the granularity of clustering and communication efficiency. A K value that is too small may lead to too many nodes in the cluster, increasing communication burden; an excessively large K value may lead to frequent inter-cluster communication, which also reduces efficiency.

Handling outliers: The K-means algorithm is more sensitive to outliers. In WSNs, if the position or data of certain nodes are abnormal, this may have a significant impact on the clustering results. Therefore, it is important to preprocess and clean the data before applying the K-means algorithm for clustering.

Scalability and parallelization: The K-means algorithm has good scalability and can be applied to large-scale WSNs. In addition, due to its iterative nature, the K-means algorithm is also suitable for parallelization to further improve computational efficiency.

In summary, the K-means algorithm has a potential application value in WSN clustering, but it also needs to be adjusted and optimized appropriately for specific application scenarios and needs. For example, the ability to handle large datasets can be improved by improving the selection method of initial cluster centers, introducing kernel methods to handle complex-shaped clusters, or using accelerated K-means algorithms.

We also adopted a combination of the whale algorithm (WOA) and a BP neural network to increase the coverage range of the wireless sensor network.

The WOA can obtain a more stable WOA-BP neural network model by optimizing the initial weights and thresholds of the BP neural network, which can be used in scenarios such as environmental monitoring and energy consumption prediction in WSNs. This combination model can improve the accuracy and robustness of predictions, and support smarter decision-making and more efficient data transmission. The experimental results have shown that it helps to enhance the global search and network coverage.

Author Contributions: Writing—original draft, L.W.; Writing—review & editing, A.Y.D.; Supervision, F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: This study was supported by the China-Laos-Thailand Education Digitization International Joint Research and Development Center of Yunnan Province (Project Number: 202203AP140006).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gorgich, S.; Tabatabaei, S. Proposing an Energy-Aware Routing Protocol by Using Fish Swarm Optimization Algorithm in WSN (Wireless Sensor Networks). *Wirel. Pers. Commun.* **2021**, *119*, 1935–1955. [CrossRef]
2. Computational Intelligence; Researchers at Iranian Research Organization for Science and Technology Target Computational Intelligence. Computational intelligence-based connectivity restoration in wireless sensor and actor networks. *Robot. Mach. Learn.* **2020**, 125–135.
3. Computing—Supercomputing; Research Data from National Institute of Technology Update Understanding of Supercomputing. Centroid-based Routing Protocol with Moving Sink Node for Uniform and Non-uniform Distribution of Wireless SNs. *Comput. Netw. Commun.* **2020**, 3727–3751.
4. Mobile Communications; New Findings from Department of Information Technology Update Understanding of Mobile Communications. A Competent Ad-hoc Sensor Routing Protocol for nergy Efficiency in Mobile Wireless Sensor Networks. *Telecommun. Wkly.* **2020**, 829–844.

5. Networks—Wireless Communications and Networks; New Wireless Communications and Networks Study Findings Recently Were Reported by a Researcher at Chang'an University. Research on simulation technology of wireless sensor in swimming mechanics. *Telecommun. Wkly.* **2020**, 99–105.
6. Technology—Ambient Intelligence and Humanized Computing; Findings from Kalasalingam Academy of Research and Education Provides New Data on Ambient Intelligence and Humanized Computing. Energy Aware Reliable Route Selection Scheme with Clustered Rp Model for Wireless Sensor Networks To Promote transmission efficiency. *Telecommun. Wkly.* **2020**, 58–59.
7. Mobile Communications; Study Results from University of Petroleum Broaden Understanding of Mobile Communications. Arior: Adaptive Ranking Based Improved Opportunistic Routing in Wireless Sensor Networks. *Comput. Netw. Commun.* **2020**, 153–176.
8. Information Technology—Information and Data Encoding and Encryption; Data on Information and Data Encoding and Encryption Published by Researchers at Department of Computer Systems Engineering. NTRU-Like Random Congruential Public-Key Cryptosystem for Wireless Sensor Networks. *Telecommun. Wkly.* **2020**, 184–189.
9. Sahoo, M.B.; Amgoth, T.; Pandey, M.H. Particle swarm optimization based energy efficient clustering and sink mobility in heterogeneous wireless sensor network. *Ad Hoc Netw.* **2020**, 106–107. [CrossRef]
10. Technology—Blockchain Technology; Reports from Lovely Professional University Advance Knowledge in Blockchain Technology. Blockchain Powered Secure Range-free Localization in Wireless Sensor Networks. *Technol. News Focus* **2020**, 143–145.
11. Computing—Supercomputing; Studies from National Institute of Technology Further Understanding of Supercomputing. Dv-hop-based Range-free Localization Algorithm for WSN Using Runner-root Optimization. *J. Math.* **2020**, 281–283.
12. Sensor Research; Researchers from Xi'an Jiaotong University Report Findings in Sensor Research. System Level Design of Wireless Sensor Node Powered By Piezoelectric Vibration Energy Harvesting. *Technol. News Focus* **2020**, 156–160.
13. Electronics; Study Findings on Electronics Discussed by Researchers at Xi'an Jiaotong-Liverpool University. Application of WSN Based on Hierarchical Edge Computing Structure in Rapid Response System. *Electron. Newsweekly* **2020**, 165–167.
14. Robotics; Reports from University of California Advance Knowledge in Robotics. Quadrotor-Based Lighthouse Localization with Time-Synchronized Wireless SNs and Bearing-Only Measurements. *Robot. Mach. Learn.* **2020**, 172–177.
15. Networks—Wireless Communications and Networks; Researchers' from China University Report Details of New Studies and Findings in the Area of Wireless Communications and Networks. Simplified clustering and improved intercluster cooperation approach for WSN energy balanced efficiency. *Netw. Wkly. News* **2020**, 131–133.
16. Computing—Parallel and Distributed Computing; Reports Summarize Parallel and Distributed Computing Findings from University of Padua. Distributed On-demand Clustering Algorithm for Lifetime Optimization in Wireless Sensor Networks. *Comput. Netw. Commun.* **2020**, 99–105.
17. Sensor Research; Reports from Umm Al-Qura University Advance Knowledge in Sensor Research. Wireless Sensor Networks Fault-Tolerance Based on Graph Domination with Parallel Scatter Search. *J. Math.* **2020**, 121–125.
18. Banerjee, P.S.; Mandal, S.N.; De, D.; Maiti, B. i Sleep: Thermal entropy aware intelligent sleep scheduling algorithm for wireless sensor network. *Microsyst. Technol.* **2020**, *26*, 2305–2323. [CrossRef]
19. Carrier Corporation. Patent Issued for System and Method for Adjusting Power in A Wireless Sensor (USPTO 10,667,220). Computers. *Netw. Commun.* **2020**, 172–177.
20. Technology—Ambient Intelligence and Humanized Computing; New Findings on Ambient Intelligence and Humanized Computing from Sona College of Technology Summarized. Cooperative spectrum sensing in wireless sensor networks using forager bee's intelligence. *Telecommun. Wkly.* **2020**, 172–177.
21. Yang, H.; Zheng, J.; Li, Y.; Zhang, Z. UAV Route Planning Based on Fuzzy PID Control Ant Colony Algorithm. In Proceedings of the 42nd China Control Conference, Tianjin, China, 24–26 July 2023; pp. 568–573.
22. Wu, P.; Huang, J.; Li, X.; Chen, B. Multi miscile routes co planning method based on improved ant colony algorithm. In Proceedings of the 42nd China Control Conference, Tianjin, China, 24–26 July 2023; pp. 383–388.
23. Jia, H.; Chen, P.; Wei, Y.; Niu, Z. Application of Particle ACOD in Underwater Planning Path Search. In Proceedings of the 6th International Conference on Smart Education and Artificial Intelligence Development, Barcelona, Spain, 17–19 February 2023; Volume 1, pp. 149–151.
24. Liu, Y. (Ed.) An Optimized Ant Colony Algorithm Combining with Elite Ant System and AHP Analysis for Path Planning. In Proceedings of the 2023 World Transport Conference (WTC2023) Theses Collection, Wuhan, Cina, 14–17 June 2023; Volume 2, pp. 494–498.
25. Gao, L.; Yin, S.; Shao, Y. On the basis of A * Algorithm and improved ant colony algorithm for intelligent robot path planning. In Proceedings of the 35th China Conference on Control and Decision Making, Yichang, China, 20–22 May 2023; pp. 477–481.
26. Ma, L.; Liu, T.; Shen, Z.S. Track planning model of USV based on multi fertility pattern ant colony algorithm. In Proceedings of the 35th China Conference on Control and Decision Making, Yichang, China, 20–22 May 2023; pp. 482–485.
27. Duan, G.; Liu, M.; Kong, W.; Cui, B. (Eds.) Research on Subway Fire Evaluation Path Planning Based on Improved Ant Colony Algorithm. In Proceedings of the 35th China Conference on Control and Decision Making, Yichang, China, 20–22 May 2023; pp. 247–253.
28. Wang, W.; Ji, S.; Song, Q.; Wang, B. (Eds.) An Improved Ant Colony Algorithm Based on Competition Mechanism of SOM Neural Network transmission efficiency. In Proceedings of the 35th China Control and Decision Making Conference Proceedings, Yichang, China, 20–22 May 2023; pp. 254–259.

29. Jin, M.; Fu, G.; Fa, T.; Huang, Z.; Chu, Z. (Eds.) Ant Colony System with Sparse Phenome. In Proceedings of the 2022 21st International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES 2022), Chizhou, China, 14–18 October 2022; pp. 168–171.
30. Wang, S.; Gu, J. (Eds.) Research on garbage truck path planning method based on improved ant colony algorithm Paper. In Proceedings of the 4th International Conference on Computer Engineering, Information Science and Internet Technology (CII 2022), Sanya City, China, 24–25 September 2022; pp. 280–289.
31. Guo, N.; Qian, B.; Na, J.; Hu, R.; Mao, J.L. An Enhanced Ant Colony Algorithm with Variable Neighborhood Descent for Multi department Vehicle Routing Problem with Time Limits. In Proceedings of the 41st China Control Conference, Heifei, China, 27 July 2022; pp. 212–217.
32. Jin, Z.; Shao, Q.; Luan, M.; Zhao, X.; Song, Q. Study on Route Optimization of Shenyang Subway Network Base on Improved Ant Colony Algorithm. In Proceedings of the 34th China Conference on Control and Decision Making, Hefei, China, 21–23 May 2022; pp. 451–453.
33. Lei, L.; Chen, G. (Eds.) Research on Ant Colony Optimization of Oversampling Problem Based on Bootstrap and Maximum Entropy Principles. In Proceedings of the 34th China Conference on Control and Decision Making, Hefei, China, 21–23 May 2022; pp. 32–38.
34. Guo, N.; Shen, Q.Y.; Qian, B.; Na, J.; Hu, R. (Eds.) A Multi dimensional ACOD for Multi objective Multi department Vehicle Routing Problem with Time Windows. In Proceedings of the 33rd China Conference on Process Control, Kunming, China, 22–24 May 2021; p. 522.
35. Wang, G.; Zhao, Y.; Wang, Z.; Zhang, Y. Tea Picking Path Planning Based on Ant Colony Algorithm. In Proceedings of the 41st China Control Conference, Hefei, China, 25–27 July 2022; pp. 196–201.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.