Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 4 (2023)

Cluster Based Improved Particle Swarm Optimization for Optimum Cluster Head Election for Energy Efficient Routing in Wireless Sensor Networks

[1]Mrs. T. Nivetha, [2]Dr. K. Prabhavathy

[1]M.Sc(CS).,M.Phil., (Ph.d), Research Scholar,
Rathinam College of Arts and Science, Coimbatore nivisathya23@gmail.com
[2]Assistant Professor Sree Saraswathi Thyagaraja College, Pollachi

Abstract: The proposed methodology addresses critical challenges in Wireless Sensor Networks (WSN), focusing on optimizing cluster head and forwarding node selection. Leveraging an enhanced Particle Swarm Algorithm (PSO), the approach prioritizes residual energy and spatial balance in node selection. It efficiently assigns cluster head nodes to ordinary nodes and selects forwarding nodes within clusters. The algorithm incorporates proximity principles to ensure balanced positioning of nodes. Through iterative iterations, the method refines node selections, favoring candidates with higher residual energy and improved spatial distribution. This approach optimizes WSN performance, enhancing data transmission efficiency and network longevity by minimizing energy consumption. Moreover, it reduces communication overhead through piggybacking and ensures dynamic node adaptation for evolving network conditions.
Keywords - Wireless Sensor Networks, Optimization Algorithm, Genetic Algorithm, Cluster Head Election, Particle Swarm Optimization.

1. Introduction

In Wireless Sensor Networks (WSNs), conventional clustering routing protocols aim to reduce network traffic and energy consumption by consolidating data through cluster head nodes and transmitting it to the base station. However, this approach tends to accelerate the energy depletion of cluster head nodes due to their heavy responsibilities, thereby shortening the network's lifespan. To address this challenge and prevent cluster head nodes from becoming excessively distant from the base station, leading to increased energy consumption, existing clustering routing protocols often opt for selecting forwarding nodes as close as possible to the base station. In this approach, the cluster head node forwards the fused data to the selected forwarding node, which subsequently relays the data to the base station. Nevertheless, if the number of available forwarding nodes is limited, the workload on these nodes becomes burdensome. Alternatively, when employing a single-hop method to transmit data to the base station, the communication overhead of nodes surges when the distance between the node and the base station exceeds a certain threshold. Thus, the optimization of cluster head node and forwarding node selection in WSN routing protocols, along with the enhancement of the transmission path for forwarding nodes, stands as a critical task for mitigating network communication overhead, achieving balance, and extending the overall network's operational lifespan.

In response to the above problems faced by WSN routing protocols, Literature [1] uses random clustering strategy and periodic cluster head rotation to maintain the energy balance of nodes; Literature [2] uses a hierarchical autonomous system based on hop count and energy to achieve reasonable routing selection. ; Literature [3] uses relay nodes to share the data transmission tasks of cluster heads to reduce the energy consumption of cluster heads; Literature [4] uses a tree routing protocol based on virtual cross areas to reduce data transmission delays; Literature [5] uses equal spacing The non-uniform clustering method of ring division and equal-angle sector division ensures the shortest communication distance between the source node and the base station; Literature [6] uses a cluster head selection method based on the signal strength and residual energy of the node receiving base station to avoid frequent replacement of cluster heads; Literature [7] adopted a clustering routing protocol based on compressed sensing to determine the communication radius of clusters to extend network life.

Particle Swarm Optimization (PSO) algorithm is a population-based swarm intelligence optimization algorithm. It has the characteristics of simple implementation, fast convergence speed and high search accuracy. It has great advantages over other algorithms in solving combinatorial optimization problems [8]. Literature [9] uses an improved particle swarm algorithm that dynamically adjusts the number of nodes in the cluster to reduce cluster head node energy consumption; Literature [10] uses a recoding method based on particle position and speed to increase network coverage area; Literature [11] optimizes network coverage However, these three algorithms do not consider the impact of relevant factors in the particle swarm algorithm on optimal clustering. Literature [12] uses the discrete particle swarm algorithm to select cluster heads and relay nodes to build the optimal cluster structure; Literature [13] avoids the algorithm from falling into local optimality by adjusting the inertia weight coefficient; Literature [14] uses fuzzy and network coding based improved particle swarm algorithm constructs the optimal transmission path from node to base station without considering forwarding

In summary, current cluster head node selection methods primarily rely on assessing residual energy. While some approaches take into account the distance between cluster head nodes and the base station, they often overlook the importance of achieving a balanced distribution of cluster head nodes. Having cluster head nodes evenly distributed across the network can significantly reduce the total communication distance and minimize network communication overhead. Secondly, most existing clustering routing protocols tend to choose a limited number of forwarding nodes, which can lead to increased energy consumption among these nodes. Some protocols that attempt to increase the number of forwarding nodes typically employ random selection methods, which may result in redundancy with cluster head nodes and fail to consider factors like energy levels and node positions. Lastly, in current protocols, forwarding nodes utilize a combination of single-hop and multi-hop transmission methods. In particular, nodes located closer to the base station often employ a single-hop mode for data transmission. When using the multi-hop mode, the distance of adjacent forwarding nodes is mainly considered, and the point energy and directionality toward the base station are not considered, resulting in limited energy and a reasonable increase in path overhead. The protocol based on the particle swarm algorithm fails to fully utilize the advantages of the particle swarm algorithm and optimize the shortcomings of the algorithm. For example, the randomness of particle initialization can easily lead to uneven node distribution and fall into local optimality; in the speed update calculation, fixed the learning factor and inertia weight cannot balance the ability of local search and global search, making the algorithm converge slowly and making it difficult to obtain a high-quality cluster head node set, etc.

The remaining part of the paper is organized as follows, section 1 provides the brief introduction and literature survey about the optimization based cluster head election protocols. Section 2 discusses the system model of the proposed protocol. Section 3 describes the proposed protocol in details and the simulation results are tabulated in section 4 and section 5 concludes the paper with quoted references.

2. System Model

2.1 Network Model

First, make the following conventions on the network model.

1) The shape of the WSN experimental area is a flat regular figure. The sensor nodes are randomly distributed and fixed in the monitoring area. Each node is identified by a globally unique ID.
2) The energy of the base station is not limited, and the energy of all other sensor nodes is limited, but the initial energy of each node is the same, and the processing capabilities and communication capabilities are equal.
3) The wireless transmission power of the node can be self-regulated and the transmission power can be selected independently.

2.2 Energy Consumption Model

In wireless sensor networks (WSN), sensor node energy consumption encompasses communication energy usage (including data sending and receiving) and data processing energy (like data fusion). Communication energy is influenced by factors such as the communication environment, transmission distance, and data packet size. This paper employs a low-power adaptive clustering and hierarchical energy consumption model, choosing between free space and multi-path attenuation models based on distance (with "d" representing distance and "d0" the threshold). "Efs" and "Emp" are power amplifier factor parameters, "m" stands for data packet size in bits, "Eelec" signifies the energy consumption per 1-bit data transmission, and "EDA" indicates the energy consumed for each 1-bit data fusion. The paper outlines the computation methods for sending energy consumption "ETX(m,d)" and receiving energy consumption "ERX(m,d)" when nodes separated by distance "d" transmit "m" bits of data, along with the calculation method for fusion energy consumption "EDA(m,d)" when fusing "m" bits of data.

2.3 Cluster Head Election

- ***Cluster Head Initialization***

To simplify the energy-intensive calculations associated with particle swarm algorithms, the clustering and routing computations based on this algorithm are offloaded to base stations equipped with abundant energy resources. During the initialization phase, all nodes transmit their remaining energy, locations, and numerical information to the base station, which receives and stores this data. Subsequently, the base station conducts clustering calculations using the particle swarm algorithm, and the resulting outcomes are broadcasted. Each surviving node, upon receiving this broadcasted information, obtains precise location details of the elected node and routing node. This approach enables the base station to perform the computational workload of the particle swarm algorithm, thereby optimizing the election and clustering processes. Consequently, individual nodes are relieved from performing intricate computations, effectively conserving node energy.The particle swarm algorithm relies on random particle selection, similar to many traditional clustering routing protocols that use random methods for cluster head selection. However, this randomness can often lead to local optimization issues and result in uneven node distribution within clusters. Additionally, if a selected node possesses insufficient energy, it may struggle to fulfill the demanding responsibilities of a cluster head node. To mitigate these challenges, there are energy limitations imposed on nodes participating in the cluster head node selection process.

Assume that there are N surviving nodes in the WSN, the energy of node i is E(i), and the base station calculates the average remaining energy of all nodes in the WSN as to ensure that the chosen cluster head nodes possess adequate energy for data processing within the cluster, the base station initiates the process by forming a set, denoted as EA, comprising all nodes with energy levels greater than or equal to Eavg. Subsequently, a random selection method is employed to pick K nodes from this set, designating them as candidates for cluster head positions. These selected nodes collectively represent one particle. After establishing this initial group of candidate cluster head nodes, non-cluster head nodes join the nearest cluster head nodes to create the initial clustering structure.The process continues by randomly selecting K nodes from the EA in a total of M iterations. Eventually, this results in M sets of initial cluster head nodes, forming M particles, each corresponding to a distinct cluster configuration. In essence, M different clusters are generated through this procedure.

- **Fitness Function**

Due to the high energy consumption of cluster head nodes, using node residual energy as an evaluation index is conducive to selecting higher energy cluster head nodes; the more balanced the distribution of cluster head locations in the network, the closer the cluster head is to non-cluster head nodes and base stations. The smaller it is, the smaller the communication overhead will be. In order to select the optimal candidate cluster head node set from the M groups of initial cluster head node sets as the cluster head node set, the remaining energy and position of the node are used as evaluation indicators, and a fitness function is constructed to evaluate each candidate cluster head node set.

The fitness function needs to comprehensively evaluate all candidate cluster head nodes in the initial cluster head node set, which needs to consider the average energy and equilibrium position of all candidate cluster head nodes. The larger the fitness function value is, the better the selected cluster head node set is.

1) Energy Factor

The energy factor, denoted as f1, is determined as the ratio of the average remaining energy among all candidate cluster head nodes in the candidate cluster head node set to the average remaining energy among all non-cluster head nodes. This factor provides insight into the relative energy levels between candidate cluster head nodes and non-cluster head nodes in the Wireless Sensor Network (WSN).

Here's the formula for calculating the energy factor f1:

f1 = (1/K) * Σ [CHer(i)] / [(1/(N-K)) * Σ [NCHEr(j)]]
Where:
K represents the number of candidate cluster head nodes in the candidate cluster head node set. N stands for the total number of surviving nodes in the WSN.
CHer(i) signifies the remaining energy of candidate cluster head node CHi in the r-th round. NCHEr(j) represents the remaining energy of non-cluster head node NCHj in the r-th round.
In essence, f1 captures the energy balance between the candidate cluster head nodes and the non- cluster head nodes in each round of clustering calculations.

2) Position Balancing Factor

The location balance factor, denoted as f2, characterizes the distribution of candidate cluster head nodes within the Wireless Sensor Network (WSN) based on communication distances. It assesses the relative distances between various nodes, including the distances between candidate cluster head nodes and the base station, as well as the distances between each candidate cluster head node and the non-cluster head nodes within its cluster. The sum of these distances is inversely related to the aggregate distance between all non-cluster head nodes and the base station.
Here's the formula for calculating the location balance factor f2:
f2 = [(1/ΣΣ d(NCHi, CHj))] / [(Σ d(NCHi, BS) + Σ d(CHj, BS)) * (1/N(N-1)) * ΣΣ d(NCHi, CHj) + ΣΣΣ d(NCHi, CHk)]
Where:
N represents the total number of surviving nodes in the WSN. K denotes the number of candidate cluster head nodes.
N-K represents the number of non-cluster head nodes.
d(NCHi, BS) represents the distance between non-cluster head node NCHi and the base station BS. d(CHj, BS) represents the distance between cluster head node CHj and the base station BS.
d(NCHi, CHj) signifies the distance from non-cluster head node NCHi to its corresponding candidate cluster head node CHj.
d(NCHi, CHk) represents the distance between non-cluster head node NCHi and other candidate cluster head nodes CHk.

This formula captures the degree of location balance among candidate cluster head nodes, taking into account their distances from both the base station and the non-cluster head nodes within the cluster.Indeed, in the context of the formula for the location balance factor f2, when a non-cluster head node NCHi is situated within the cluster led by cluster head node CHj, it contributes to a more balanced distribution of candidate cluster head node positions in the Wireless Sensor Network (WSN). This is particularly evident when the candidate cluster head node set is closer to the base station, as it results in shorter distances between non-cluster head nodes and their respective cluster head nodes in each sub-cluster. Consequently, this configuration reduces network communication distances and yields larger values for the location balance factor f2, signifying a more evenly distributed candidate cluster head node set within the WSN.

Based on the energy factor and position balance factor, the fitness of the candidate cluster head node set is calculated in a weighted manner. The calculation method of the fitness value function F1 is:

$$F1 = af1 + (1-a)f2$$

In the provided context, the weight coefficient "a," falling within the range (0,1], serves as a weighting factor in the fitness function. Depending on the specific requirements of the Wireless Sensor Network (WSN) regarding residual energy and position balance, this weight can be adjusted accordingly.

In the fitness function, a larger residual energy within the candidate cluster head node set corresponds to a more balanced position distribution, and consequently, a higher fitness value. This indicates that a candidate cluster head node set with greater residual energy and balanced positions is considered better in terms of fitness.

To determine the global optimal position, the following steps are taken with M groups of initial cluster head node sets:

1. Calculate the fitness function for each group of candidate cluster head node sets.
2. Record the position with the maximum fitness within each candidate cluster head node set as the position for that group.
3. Initially, each candidate cluster head node set is considered its own local optimal position.
4. Among the initial M groups of cluster head node sets, the candidate cluster head node set with the maximum fitness function value is designated as the global optimal position.

This process helps identify the candidate cluster head node set with the best fitness and position distribution, ensuring the selection of an optimal configuration for the WSN.

- **Speed and Position Update Method**

During the iterative optimization process, which follows the initial fitness calculation and the determination of local and global optimal positions, the positions of the candidate cluster head node sets are updated, and their fitness is recalculated. To facilitate this position update and achieve optimization results, a velocity parameter is introduced to control the rate of change in positions. This velocity is represented as a vector.

Specifically, the velocity components for each candidate cluster head node in the x and y directions are denoted as vxid and vyid, respectively, while the position components are represented by xxid and xyid. Initially, the calculation of the two velocity components is random. However, in each subsequent iteration round, these velocity components are determined based on the velocity components from the previous iteration round, the local optimal position (pxid, pyid), and the global optimal position (pxgd, pygd).

The specific calculation method for updating the velocity components and positions is defined based on the relationships between these parameters and is further elaborated in reference [15]. This method guides the iterative adjustments in the positions of candidate cluster head nodes, ultimately leading to optimization results in the Wireless Sensor Network (WSN).

In the iterative optimization process, the positions of candidate cluster head node sets are updated, and their fitness is recalculated. A velocity parameter, represented as a vector with components vxid and vyid for the x and y directions, controls the rate of position change. Initially random, in subsequent iterations, the velocity components are determined based on the previous round's velocity, the local optimal position (pxid, pyid), and the global optimal position (pxgd, pygd). These updates are influenced by factors such as the inertia weight (w), indicating the impact of the previous round's velocity, cognitive learning factor (c1) and social learning factor (c2), representing acceleration based on proximity to local and global optimal positions, and random numbers (r1 and r2) for introducing variability. The specific calculation method for updating position components is described in reference [15] and guides the iterative adjustments, ultimately optimizing the positions of candidate cluster head nodes in the Wireless Sensor Network (WSN).

$$\begin{cases} x_{xid}(t) = x_{xid}(t-1) + v_{xid}(t) \\ x_{yid}(t) = x_{yid}(t-1) + v_{yid}(t) \end{cases}$$

- **With adaptive Learning Factors and Inertial Weights**

In the above speed update calculation, the learning factors and inertia weights in traditional routing protocols based on particle swarm optimization are usually set to fixed values, which cannot balance local search capabilities and global search capabilities. The convergence speed of the algorithm is relatively slow, and it is difficult to obtain high quality. The set of cluster head nodes.

In the selection process of the optimal cluster head node set, the early iterations focus on local optimal search, and the later iterations focus on global optimal search. If the search range of local optimal search is small, the algorithm will easily fall into the local optimal solution. Therefore, it is necessary to expand the local search range to find the optimal candidate cluster head node set as much as possible to enhance the diversity of the group; global search needs to speed up the algorithm Convergence, maintaining a balance between convergence speed and search effect. In order to achieve this goal, the cognitive learning factor c1 and the social learning factor c2 are set to change dynamically. In the stepwise iterative process, c1 changes from large to small and c2 changes from small to large.

In order to meet the above changing rules of the two learning factors, a dynamically changing learning factor is constructed based on the fixed value setting of the traditional learning factor. The calculation method is:

$$\begin{cases} c_1 = 2.5 - 2\left(\dfrac{t}{t_{\max}}\right)^2 \\ c_2 = 0.5 + 2\left(\dfrac{t}{t_{\max}}\right)^2 \end{cases}$$

Among them, t is the number of iterations in this round, and tmax is the maximum number of iterations. As the number of iterations changes, c1 and c2 dynamically change to meet their changing rules, allowing the algorithm to adaptively expand the local search range in the early iterations and accelerate the global convergence speed in the later iterations.

During the iteration process, the inertia weight can affect the search range of this round based on the speed of the previous round. At the end of each round of iteration, the fitness function is calculated for the selected candidate cluster head node set, and the inertia weight is dynamically adjusted based on the fitness value result, so that the cluster head node set selected in this round of iteration has a more balanced position. Here, the nonlinear adaptive inertia weight strategy [16] is used to calculate the inertia weight, the method is

$$w = \begin{cases} w_{\min} + \dfrac{(w_{\max} - w_{\min})(f_i - f_{\min})}{f_{avg} - f_{\min}}, f_i \leqslant f_{avg} \\ w_{\max}, f_i > f_{avg} \end{cases}$$

Among them, w_{max} and w_{min} are the maximum and minimum inertia weights set respectively, f_i is the fitness value of the candidate cluster head node CH_i, f_{min}, f_{max} and f_{avg} respectively represent the minimum, maximum and average fitness values of the candidate cluster head node set in this round. When $f_i > f_{avg}$, the speed of this candidate cluster head node mainly refers to the speed of the previous round, increasing the activity of the candidate cluster head node set; conversely, the speed of this candidate cluster head node mainly refers to the local optimal position and global optimal position accelerates the candidate cluster head node set to move closer to the advantage space.

- **Location Mapping Strategy**

Following each iteration, the candidate cluster head node set's positions undergo updates, and it's possible that the updated node positions may not correspond to any surviving nodes within the Wireless Sensor Network (WSN). In such cases, location mapping processing [11] becomes necessary. The fundamental concept behind this process is to apply the proximity principle, mapping the updated locations to the nearest surviving nodes in terms of spatial proximity. Utilizing Xxid and Xyid as the updated node coordinates and CMnx and CMny as the coordinates of the nearest surviving node CMn within the network, the position mapping process is executed as follows:

$$\mathrm{CM}_n = \left\{ (\mathrm{CM}_{nx}, \mathrm{CM}_{ny}) \middle| \min \sqrt{(\mathrm{CM}_{nx} - X_{xid})^2 + (\mathrm{CM}_{ny} - X_{yid})^2} \right\}$$

The location mapping strategy is employed to address discrepancies arising from the discrete distribution of network nodes, which can lead to updated positions not aligning with the actual locations of surviving nodes. When multiple nodes share the same updated position coordinates, a flag is utilized during node updates. In subsequent updates and mapping procedures, nodes first check if the flag designates them as cluster head nodes. If so, they sequentially select the closest node positions for mapping.

Following position mapping, each candidate cluster head node set, post-update, is treated as an optimization outcome, and their fitness values are computed. Based on these calculations, updates are made to the local optimal position for each candidate cluster head node set and the global optimal position for the M groups of cluster head node sets in this round. If the iteration is ongoing, the process continues with position updates and mapping for candidate cluster head nodes. When the iteration concludes, the candidate cluster head node set at the global optimal position becomes the optimal cluster head node set, marking the culmination of the cluster head election process.

After the cluster head election, the base station proceeds to calculate the distances from non- cluster head nodes to each cluster head node. These nodes are divided into cluster head nodes and ordinary nodes, where ordinary nodes exclusively handle data transmission, fuse their own monitored data, and forward it directly to the cluster head node. The cluster head node receives data from ordinary nodes, performs data fusion, and subsequently transmits it to the appropriate forwarding node. This forwarding node, chosen from among ordinary nodes, receives data from the cluster head and determines the best path to the base station, ultimately forwarding the data to the base station.

Throughout the clustering calculations based on the particle swarm algorithm, it is essential to retrieve the remaining energy of each node. To minimize communication overhead and facilitate subsequent rounds of clustering and node communication interactions, a piggyback technology is implemented. Under this approach, each node, after completing data collection and processing, adds its remaining energy to the data and transmits it to the cluster head node. The cluster head node performs data fusion and appends its own remaining energy to the fused data, sending it to the base station via the forwarding node. In a similar manner, the forwarding node incorporates its remaining energy during transmission. Upon receiving the data, the base station records the received remaining energy information for each node, alongside the data. This data serves as the basis for a new round of cluster elections, where the remaining energy information carried by each node is the current node energy minus the energy expended during data transmission.

2.4 Forwarding Node Election and Multi-Hop Transmission

- **Election of Forwarding Nodes**

Following the completion of the cluster head node election, the subsequent step involves selecting the appropriate forwarding node for each cluster head node. When the network lacks an adequate number of forwarding nodes, multiple cluster head nodes sending data to the base station through a limited number of forwarding nodes can lead to increased energy consumption for these forwarding nodes. To address this issue, existing Wireless Sensor Network (WSN) routing protocols typically assign one forwarding node per cluster head node [11] to augment the number of forwarding nodes. However, these protocols often use a random selection approach among all nodes without considering the selected node's remaining energy or spatial balance.

In this paper, when selecting forwarding nodes, the improved particle swarm algorithm used for cluster head node election, as mentioned earlier, is employed to elect a forwarding node for each cluster head node from the pool of ordinary nodes within its cluster. This ensures that the selected forwarding node exhibits an optimal energy and location relationship, mitigating the problem of having too few forwarding nodes in the WSN and preventing accelerated energy consumption.

In the calculation and evaluation of forwarding nodes, a distinction is made from the cluster head node election regarding the energy factor and position balancing factor calculation method.

Similarly, in this context, "N" represents the count of surviving nodes in the WSN. After electing "K" cluster head nodes, the WSN is divided into "K" clusters. During initialization, the higher remaining energy nodes in each cluster, as per the cluster head node screening method, are identified. These candidate high-energy nodes are grouped to form a candidate forwarding node set, with each candidate set containing "K" nodes.Post-initialization, the number of ordinary nodes is calculated as "N-2K." Here, "ERN(i)" represents the remaining energy of candidate forwarding node "RNi" in the "rth" round, and "ECN(j)" represents the remaining energy of ordinary node "CNj" in the "rth" round. The energy factor, denoted as "fit1," is calculated using the following formula:

$$\mathrm{fit}_1 = \frac{\frac{1}{K}\sum_{i=1}^{K} E_{\mathrm{RN}}^{r}(i)}{\frac{1}{N-2K}\sum_{j=1}^{N-2K} E_{\mathrm{CN}}^{r}(j)}$$

Certainly, in this context:

- "d(CNk, CHj)" signifies the distance from the ordinary node "CNk" to the corresponding cluster head node "CHj."
- "d(RNi, BS)" represents the distance between the forwarding node "RNi" and the base station "BS."
- "d(RNi, CHj)" denotes the distance from the forwarding node "RNi" to the associated cluster head node "CHj."
- "d(RNi, RNm)" represents the distance between forwarding nodes "RNi" and "RNm."
- The calculation method for the forwarding node position balancing factor, referred to as "fit2," is determined as follows:

$$\mathrm{fit}_2 = \frac{\frac{1}{N-2K}\sum_{k=1}^{N-2K} d(\mathrm{CN}_k,\mathrm{CH}_j)}{\frac{1}{K^2}\left(\sum_{i=1}^{K} d(\mathrm{RN}_i,\mathrm{BS}) + \sum_{i=1}^{K} d(\mathrm{RN}_i,\mathrm{CH}_j) + \sum_{i=1}^{K-1}\sum_{m=i+1}^{K} d(\mathrm{RN}_i,\mathrm{RN}_m)\right)}$$

In this context, it's important to note that the ordinary node "CNk" is situated within the cluster led by the cluster head node "CHj," and the candidate forwarding node "RNi" corresponds to the cluster head node "CHj." The proximity principle is applied, meaning that the closer the candidate forwarding node set is to the base station, the shorter the distances between cluster head nodes and their corresponding forwarding nodes within each cluster. As a result, the distances between the forwarding nodes in the candidate forwarding node set become smaller, leading to a more balanced distribution of positions among the candidate forwarding nodes. Consequently, the value of "fit2" increases as the position distribution of the candidate forwarding node set becomes more balanced.

$$F_2 = b\,\mathrm{fit}_1 + (1-b)\,\mathrm{fit}_2$$

In this context, the parameter "b," falling within the range (0, 1], represents the weight coefficient. The fitness value of the candidate forwarding node set is influenced by both the residual energy of the candidate forwarding node set and the balance in their positions. Specifically, a higher residual energy and a more balanced position distribution contribute to a greater fitness value for the candidate forwarding node set. This increase in fitness value signifies that the candidate forwarding node set is more advantageous.

During multiple iterations of the candidate forwarding node set, the selection of the optimal forwarding node set employs a method similar to the cluster head node election, involving speed updates and position mapping to determine the most suitable candidates.

- **Multi-hop Transmission of Forwarding Nodes**

Based on the LEACH energy consumption model [1], if the base station is far away from the WSN and the distance d between the forwarding node and the base station is greater than the threshold distance d0, the sending energy consumption level is d4; but if d≤d0, the energy consumption of data transmission between nodes is greatly reduced. Its energy consumption level is d2. In practical applications, the base station is usually far away from the WSN. In existing WSN routing algorithms, forwarding nodes often use a single-hop method to forward data to the base station [13], which consumes too much energy; while algorithms using multi-hop methods mainly select multiple hops based on the shortest distance between forwarding nodes [17], without Considering the remaining energy of the forwarding node; and without considering the direction of the base station, the multi-hop path may not be the shortest distance.

After selecting the optimal set of forwarding nodes, this paper will determine whether the forwarding node uses a single-hop or multi-hop mode to transmit data to the base station based on the distance d between the forwarding node and the base station. If d>d0, then the forwarding node uses a multi-hop mode to transmit data; Otherwise, the forwarding node uses single-hop transmission.

In the multi-hop path selection of forwarding nodes, this paper uses the method of constructing a minimum spanning tree to search for the shortest path for multi-hop transmission of forwarding nodes, while taking into account the remaining energy of the nodes to maintain the balance of forwarding node consumption while reducing network energy consumption.

The route between forwarding nodes takes the base station as the root of the tree. At the beginning, each forwarding node is abstracted as a point, and the forwarding nodes are connected with edges to construct a weighted connected graph G=(V,E), where V Including all forwarding nodes, E includes the set of edges between any two nodes in V. In order to search for the optimal path from a certain forwarding node to the base station, the distance and remaining energy between two adjacent forwarding nodes in each hop need to be comprehensively considered.

Assume that the forwarding node RNi is used as the starting point to search for the next hop node passed by the base station; if the forwarding node RNj is a neighbor forwarding node of RNi, it is necessary to evaluate whether the node RNj can be used as the next hop node. Here, based on these 2 It is measured by the weight of the edge of two nodes. The weight is determined by the distance between the two nodes and the remaining energy, and is represented by wi,j. If the distance between node RNj and the base station is greater than or equal to the distance between node RNi and the base station, then node RNj cannot be used as the next hop node, and wi,j is set to ∞; otherwise, wi,j is calculated based on the distance of two nodes and the remaining energy. The method is

$$w_{i,j}=\begin{cases}\dfrac{E_{\mathrm{TX}}(m,d_{i,j})}{E_{\mathrm{RN}}^{r}(i)E_{\mathrm{RN}}^{r}(j)}, d_{j,\mathrm{BS}}<d_{i,\mathrm{BS}}\\ \infty, d_{j,\mathrm{BS}}\geqslant d_{i,\mathrm{BS}}\end{cases}$$

Among them, di,j represents the distance between the forwarding node RNi and RNj, di,BS represents the distance between the forwarding node RNi and the base station, RNEr (i) and RNEr (j) represent the r-th round of forwarding nodes RNi and RNj respectively. remaining energy. If the distance between two forwarding nodes is larger and the remaining energy is smaller, the weight of the two nodes is larger, and the probability of the neighbor forwarding node being selected as the next hop is smaller. If the forwarding node RNi has multiple adjacent forwarding nodes, calculate the weights of the forwarding node RNi and other adjacent nodes respectively, and select the forwarding node with the smallest weight as its next hop node.

After each round of forwarding node election, the above-mentioned minimum spanning tree method will be used to establish a multi-hop path from the forwarding node to the base station. The specific process of the multi-hop path establishment method based on the minimum spanning tree is as follows. In the weighted connected graph G=(V, E) constructed above, base station v0 is added to V as the root node of the tree, U records the node set of the minimum spanning tree, and W records the distance between the forwarding node to be selected and its neighbor nodes. The set of weights that constitute the edge, T records the set of weights of the edges in the minimum spanning tree.

Step 1: Initially add the root node v0 to the set U, and T and W are empty.

Step 2: According to the set threshold distance d0, calculate the distance di,0 from a certain node vi in V except v0 to v0 in sequence. If di,0≤d0, then vi will transmit data to the base station in a single hop mode, and add vi in U, set the weight wi,0=0 of the edge between vi and v0, add wi,0 to T, and go to step 5; otherwise, go to step 3 to calculate and establish a multi-hop path from vi to v0.

Step 3: Calculate the weight of the edge from vi to all other forwarding nodes according to equation (13), and add it to W.

Step 4: Select the smallest weight wi,k in W. At this time, the distance from vi to v0 is di,0, and the distance from vk to v0 is dk,0. Calculate the distance di,k from vi to vk. If di,0≤di,k, or ETX(m, di,BS)<(ETX(m, di,j)+ETX(m, di,BS)), then the overhead of sending data from vi to v0 via vk is larger, at this time, vi will directly send data to v0, add node vi to U, set the weight of the edge between vi and v0 wi,0=0, add wi,0 to T, and set W to be empty; Otherwise, add node vi to U, add wi,k to T, and set W to be empty.

Step 5: If U=V, end the search and go to step 6; otherwise, go to step 2.

Step 6: For the weight in T whose weight is 0 and the next node is v0, the previous node of the edge corresponding to the weight is output as a single-hop node; otherwise, the previous node of the edge corresponding to the weight is used as the starting point, and then One node serves as the next hop node and continues to search for the next hop node until the next hop node is v0, forming a multi-hop path output.

3. Simulation Results

WSN is simulated and generated in MATLAB, and the base station is located outside the network. In this environment, a routing protocol based on the improved particle swarm algorithm is implemented and tested. The test computer configuration is 2.3 GHz Intel Core i5, 8 GB memory, 64- bit Windows 10. The relevant experimental environment and parameter settings of WSN initialization, cluster head node and forwarding node election, and data processing are shown in Table 1.

Table 1: Simulation Parameters

Sno	Parameter	Value
1	Number of nodes	100
2	Network size	100m x 100m
3	Maximum number of rounds	3000
4	Number of iterations of the algorithm	100
5	Optimal number of clusters	5
6	Number of particles	20
7	Data packet length/bit	4000
8	Node initial energy (J)	0.5

Considering that WSN is generally relatively stable and the frequency of data collection and data processing is not high, a new round of cluster election is mainly started based on the data processing frequency of WSN, that is, when the cluster head node completes the fusion and merger of node data in the cluster. Send it to the base station. After the base station receives the data, it will start a new round of cluster election. For the case of WSN with fast data processing frequency, this article has not considered it yet, but event-triggered transmission can be used to determine whether to start a new round of cluster election by judging the remaining energy of the cluster head node.

Based on the simulation environment outlined above, we investigated the impact of varying weights in the fitness function on both the network and node energy consumption, as well as the changes in the network's life cycle under the CIPSO protocol. We then conducted a comparative analysis with the LEACH [16] and Genetic Algorithm [17] protocols.

The parameters, such as speed and maximum connections, were extracted from the network simulator, and these parameters exhibited variations corresponding to the nodes are shown in Table 2.

Table 2: Values Set for Parameters

Sl.no	Variable Parameter	Constant Parameter	Values
1	Pause Time (0-120)	Nodes	100
		Max speed	10
		Max Connections	10
2	Nodes (15-60)	Pause Time	0
		Max Speed	10
		Max connection	10
3	Max Speed (5-45ms)	Pause node	0
		Nodes	60
		Max Connections	10

Performance Measures

Several metrics are employed in this study to assess the performance of clustering protocols. Specifically, the following measures are utilized:

1. **Network lifetime:** This metric calculates the time span from the initiation of the sensor network until the final sensor ceases to function.
2. **Stability Period:** This metric determines the duration from the commencement of network operation until the first sensor becomes inactive.
3. **Throughput:** This measure quantifies the overall data transmission rate within the network, encompassing both data sent from Cluster Heads (CHs) to the sink and data transferred from nodes to their respective CHs.
4. Figure 1 illustrates the network lifetime for the proposed protocol as well as for other protocols such as LEACH and Genetic Algorithm.

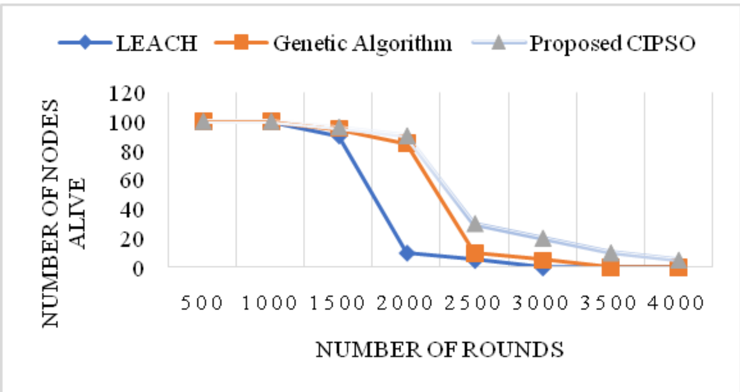

Fig 1: Network Lifetime analysis

The analysis of network lifetime is depicted through a count of rounds. Each round commences with a setup phase, assuming clusters are established, and transitions into a steady-state phase where data is transmitted to the base station. A lower ratio of dead nodes in the same round is indicative of superior algorithm performance. In this study, the network's life cycle concludes when 80% of nodes have ceased to operate, as observed in the experimental results.The network lifetime performance is then compared among LEACH, Genetic Algorithm, and CIPSO optimization, and the results are presented in the following performance Table3:

Table 3: Network Life time Analysis

Number of Rounds	LEACH	Genetic Algorith m	roposed CIPSO
500	100	100	100
1000	100	100	100
1500	90	95	96
2000	10	85	90
2500	5	10	30
3000	0	5	20
3500	0	0	10
4000	0	0	5

The total number of packets sent to Cluster Heads and the Base Station differs significantly between LEACH, GA, and the proposed GWA protocols, with 1.083x10^5 packets and 5.182x10^4 packets, respectively, as indicated in Figure 2. Network throughput, which quantifies how much information a system processes within a given timeframe, has widespread applicability in computer and network systems. Response time, on the other hand, denotes the duration between a user request and the receipt of a response.Comparisons among the three algorithms mentioned in the graph are made. Measuring throughput in networks serves several purposes, primarily assessing the maximum data transmission rate in bits per second along a communication link. For instance, one common method involves transferring a 'large' file from one system to another and measuring the time it takes to complete the file transfer. Throughput is then calculated by dividing the file size by the time elapsed, yielding the throughput in megabits, kilobits, or bits per second.

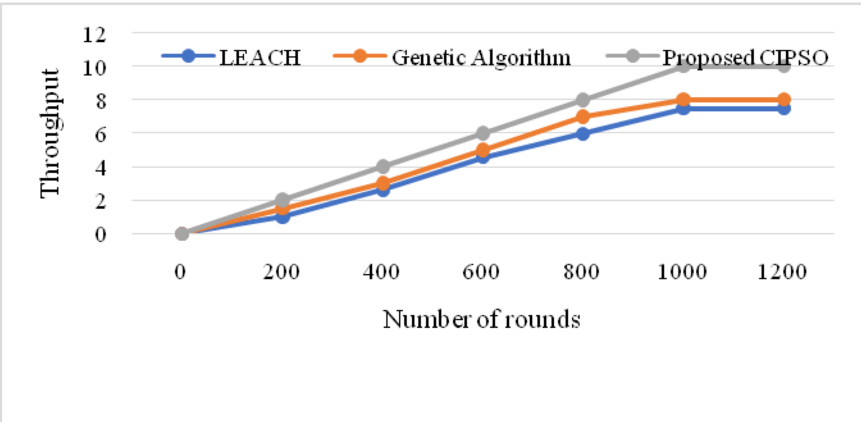

Fig 2: Network Throughput

In Figure 3, the number of rounds is illustrated for LEACH, GA, and the proposed GW optimizer. Figures 4 and 5 highlight that the proposed GW optimizer outperforms other protocols notably in terms of the stability period.

The concept of the First Dead Node (FDN) is significant, as it signifies the number of rounds it takes for the first sensor to cease functioning. This value directly influences the stability period; a larger FDN indicates a longer network stability period. Similarly, the Half Dead Node (HDN) metric indicates the number of rounds it takes for half of the sensor nodes to become inactive. Lastly, the Last Dead Node (LDN) quantifies the number of rounds it takes for all sensor nodes to become inactive.

Table 4: Rounds and Number of Throughput

Rounds	LEACH	Genetic Algorithm	Proposed CIPSO
0	0	0	0
200	1	1.5	2
400	2.6	3	4
600	4.6	5	6
800	6	7	8
1000	7.5	8	10
1200	7.5	8	10

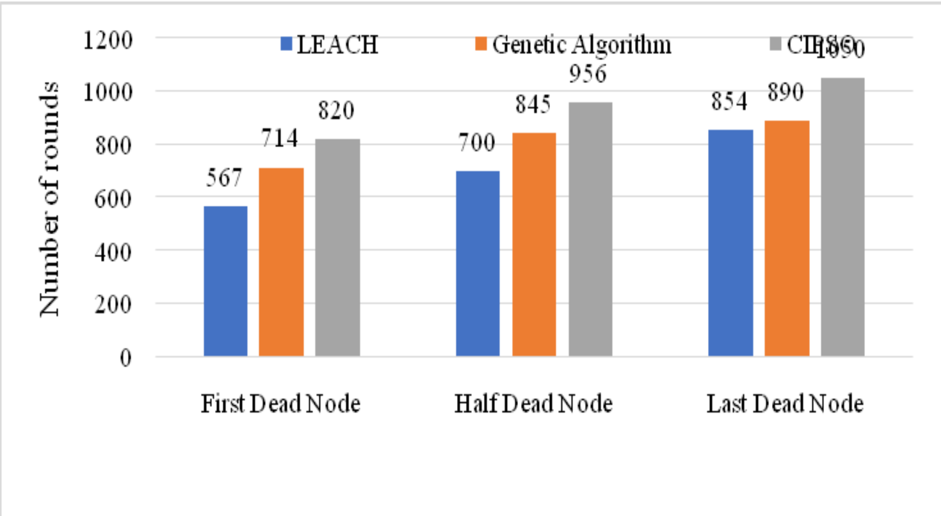

Fig 3: Number of rounds at FDN, HDN, LDN

Figure 4 displays the averaged residual energy of all nodes per round, highlighting the higher residual energy observed after 500 rounds, as examined in this simulation work.

For the simulation setup, 100 nodes are randomly distributed within a 100m x 100m sensing field, with a base station positioned at the center of each field. The initial node starts with an energy level of E0=1J. During this analysis, various parameters such as First Dead Node (FDN), Last Dead Node (LDN), and Half Dead Node (HDN) are employed to characterize the network lifetime, following the approach outlined by Mittal et al. (2016).

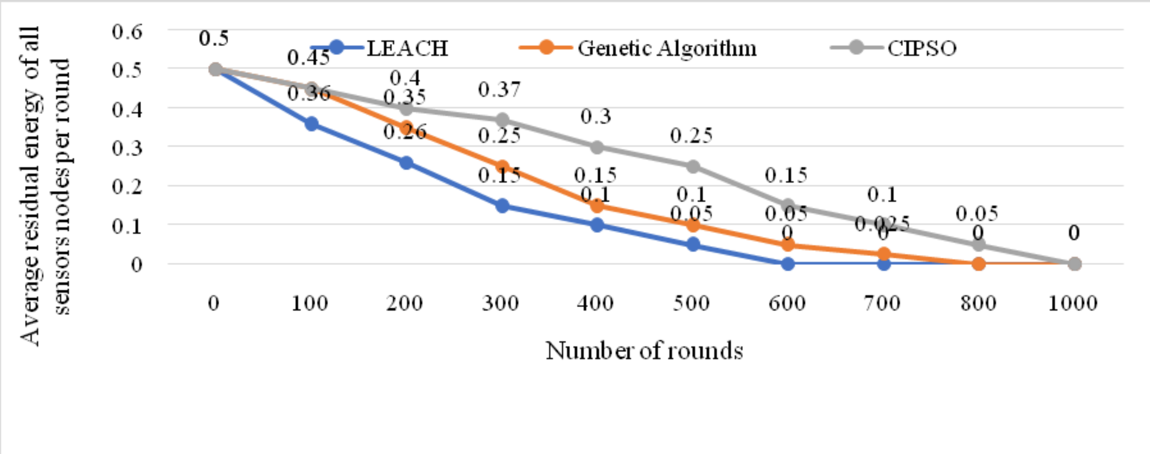

Fig 4: Averaged Residual Energy of all Nodes per Round

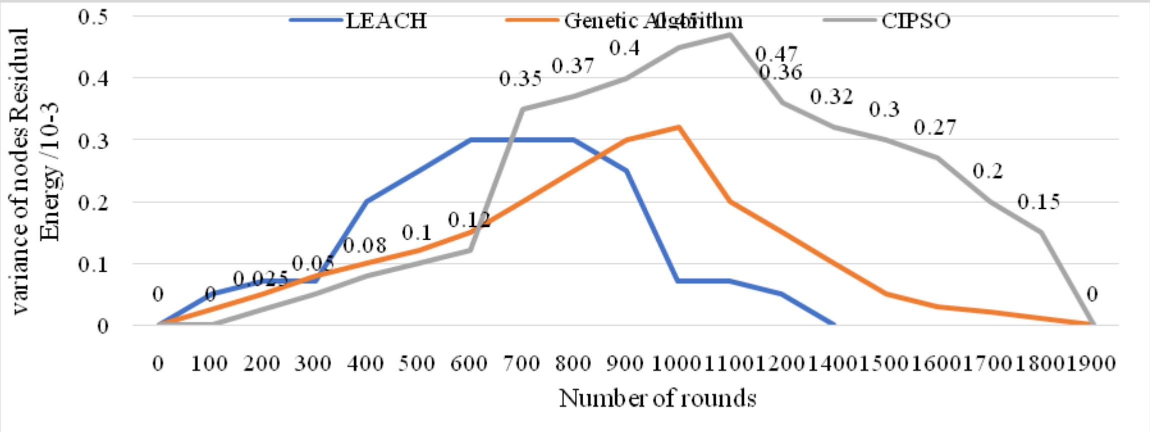

Fig 5: Variance of Nodes Residual Energy along Rounds

This paper introduces a distributed energy-efficient algorithm that involves multiple rounds of abstraction within sensor networks. Figure 5 offers a detailed description of the residual energy and its variance among the nodes, as depicted in the figure. In this process, cluster heads are selected from sensor nodes with higher residual energy through local communication. During the cluster head selection phase, certain nodes are elected, and they engage in computations among themselves to determine the cluster head.

In the initialization phase, the base station aids in this process by broadcasting messages to all nodes at a specific power level. Upon receiving the message, nodes calculate distances based on received signal strength. Table 3 provides a clear view of the optimal cost and energy reduction achieved by each algorithm. The CIPSO algorithm enhances the performance of each sensor node and cluster head, respectively. Communication between nodes and clusters is facilitated through routing, and the optimization process is conducted using the Improved PSO technique within a network simulator, aligning with the specified objectives.

4. Conclusion

This work considers parameters such as pause time, node speed, and maximum connections, which affect node behavior. The results show that the proposed CIPSO protocol outperforms other protocols in terms of stability and energy efficiency. Additionally, figures depict the residual energy of nodes over rounds and the variance of node residual energy.Concludes that the CIPSO algorithm enhances ensor node and cluster head performance and achieves energy-efficient routing in the The optimization process involves improved particle swarm optimization, and communication between nodes and clusters is facilitated through routing. Overall, the study presents a comprehensive analysis of the proposed routing protocol's performance in comparison to standard protocols.

References

[1] Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on wireless communications*, *1*(4), 660-670.

[2] Yu, J., Qi, Y., Wang, G., & Gu, X. (2012). A cluster-based routing protocol for wireless sensor networks with nonuniform node distribution. AEU-International Journal of Electronics and Communications, 66(1), 54-61.

[3] Tarhani, M., Kavian, Y. S., &Siavoshi, S. (2014). SEECH: Scalable energy efficient clustering hierarchy protocol in wireless sensor networks. IEEE Sensors Journal, 14(11), 3944-3954.

[4] Sharma, S., Puthal, D., Jena, S. K., Zomaya, A. Y., & Ranjan, R. (2017). Rendezvous based routing protocol for wireless sensor networks with mobile sink. The journal of Supercomputing, 73, 1168-1188.

[5] Zhao, X., Qu, Z., Tang, H., Tao, S., Wang, J., Li, B., & Shi, Y. (2023). A detection probability guaranteed energy-efficient scheduling mechanism in large-scale WSN. Alexandria Engineering Journal, 71, 451-462.

[6] Jabbar, M. S., Issa, S. S., & Ali, A. H. (2023). Improving WSNs execution using energy-efficient clustering algorithms with consumed energy and lifetime maximization. Indonesian Journal of Electrical Engineering and Computer Science, 29(2), 1122-1131.

[7] Mostarda, L., Navarra, A., & De Leone, R. (2023). Optimal vs rotation heuristics in the role of cluster-head for routing in IoT constrained devices. Internet of Things, 22, 100757.

[8] Sebastian, S., &Jagatheeshkumar, G. Energy Efficient Trustworthy Target Tracking Scheme (3TS) Based on Clustering and Task Cycle Scheduling for Wireless Sensor Networks.

[9] Padmageetha, B., Pramod, K. N., &Mallanagouda, P. (2023). Low Energy Data Centric Algorithm in Wireless Sensor Network. Journal of University of Shanghai for Science and Technology, 25(02), 166-171.

[10] Balasubramanian, D. L., &Govindasamy, V. (2023). Energy aware farmland fertility optimization based clustering scheme for wireless sensor networks. Microprocessors and Microsystems, 97, 104759.

[11] Kaveripakam, S., &Chinthaginjala, R. (2023). Energy balanced reliable and effective clustering for underwater wireless sensor networks. Alexandria Engineering Journal, 77, 41-62.

[12] Bharany, S., Sharma, S., Alsharabi, N., Tag Eldin, E., &Ghamry, N. A. (2023). Energy- efficient clustering protocol for underwater wireless sensor networks using optimized glowworm swarm optimization. Frontiers in Marine Science, 10, 1117787.

[13] Han, B., Li, B., & Qin, C. (2023). A novel hybrid particle swarm optimization with marine predators. Swarm and Evolutionary Computation, 101375.

[14] Solano-Rojas, B. J., Villalón-Fonseca, R., &Batres, R. (2023). Micro Evolutionary Particle Swarm Optimization (MEPSO): A new modified metaheuristic. Systems and Soft Computing, 5, 200057.

[15] Gad, A. G., Houssein, E. H., Zhou, M., Suganthan, P. N., &Wazery, Y. M. (2023). Damping- assisted evolutionary swarm intelligence for industrial iot task scheduling in cloud computing. IEEE Internet of Things Journal.

[16] Gupta, S., &Marriwala, N. (2017, September). Improved distance energy based LEACH protocol for cluster head election in wireless sensor networks. In 2017 4th International conference on signal processing, computing and control (ISPCC) (pp. 91-96). IEEE.

[17] Famila, S., & Jawahar, A. (2020). Improved artificial bee Colony optimization-based clustering technique for WSNs. Wireless Personal Communications, 110(4), 2195-2212.