# PLOS ONE

# An intelligent algorithm for energy efficiency optimization in software-defined wireless sensor networks for 5G communications

**Kemal Gökhan Nalbant[1], Suliman A. Alsuhibany[2], Asma Hassan Alshehri[3], Maha Hatira[4,5]\*, Bong Jun Choi[6]\***

1 Faculty of Engineering Architecture, Department of Software Engineering, Istanbul Beykent University, Sariyer, Istanbul, Turkey, 2 Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia, 3 Department of Computer Sciences, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al Kharj, Saudi Arabia, 4 Faculty of Engineering and Information Technology, Taiz University, Taiz, Yemen, 5 Islamic University Centre for Scientific Research, The Islamic University, Najaf, Iraq, 6 School of Computer Science and Engineering, Soongsil University, Seoul, South Korea

\* maha.hatira.yemen@outlook.com (MH); david.choi@soongsil.ac.kr (BJC)

## Abstract

Wireless communications have lately experienced substantial exploitation because they provide a lot of flexibility for data delivery. It provides connection and mobility by using air as a medium. Wireless sensor networks (WSN) are now the most popular wireless technologies. They need a communication infrastructure that is both energy and computationally efficient, which is made feasible by developing the best communication protocol algorithms. The internet of things (IoT) paradigm is anticipated to be heavily reliant on a networking architecture that is currently in development and dubbed software-defined WSN. Energy-efficient routing design is a key objective for WSNs. Cluster routing is one of the most commonly used routing techniques for extending network life. This research proposes a novel approach for increasing the energy effectiveness and longevity of software-defined WSNs. The major goal is to reduce the energy consumption of the cluster routing protocol using the firefly algorithm and high-efficiency entropy. According to the findings of the simulation, the suggested method outperforms existing algorithms in terms of system performance under various operating conditions. The number of alive nodes determined by the proposed algorithm is about 42.06% higher than Distributed Energy-Efficient Clustering with firefly algorithm (DEEC-FA) and 13.95% higher than Improved Firefly Clustering IFCEER and 12.05% higher than another referenced algorithm.

## 1 Introduction

Fifth Generation (5G) technology represents a significant advancement over all prior mobile generation networks and is a cornerstone of the digital transformation. Three new services, including Extreme Mobile Broadband (eMBB), are available to end users using 5G. In addition to many other features, it provides increased bandwidth, ultraHD streaming movies, virtual

reality and augmented reality (AR/VR) media, high-speed internet access, and minimal latency. Massive machine type communication, or eMTC, offers broadband and long-range machine-type communication at a very low cost and with minimal power usage. For Internet of Things applications, eMTC offers mobile carriers a high data rate service, low battery consumption, and wider coverage with fewer complicated devices. With its exceptionally fast speed, high throughput, low latency, improved reliability and scalability, and energy-efficient mobile communication technology, 5G offers a limitless internet connection at your leisure, anytime, anywhere [1,2].

Software-defined network (SDN) is a strategy to enable programmatically efficient network design and to facilitate network administration, hence improving network performance and monitoring [3]. It opens up a world of new creative applications, from traffic engineering to data centre virtualization, fine-grained access control, and more, by separating the data and control planes [4]. It is a strong option for 5G networks as well because it has a demonstrated benefit in several commercial networks [5,6].

A wireless sensor network (WSN) typically uses a base station and multiple sensor nodes. Due to its low deployment cost and great benefits, WSN has gained rapid development and wide popularity [1–3]. However, WSN suffers from limited network resources, inflexible network management, and partial efficiency of routing algorithms [4–6]. To solve the above problems, SDN based Wireless Sensor Networks (SDWSN) [7–10] apply software-defined technology to WSNs by separating data and control planes. The SDWSN improves network flexibility through centralized management and programmability, supports high-speed heterogeneous connectivity, flexible and efficient detection, dynamic and reliable routing, and improves WSN utilization [11–13].

Network energy efficiency is still a major obstacle to WSN deployment in SDN [14–17]. Since the original routing method is no longer fully suitable for WSNs, it is important to develop a routing strategy that considers the characteristics of SDWSNs [18–20]. Various routing algorithms such as Low Energy Adaptive Clustering Hierarchy (LEACH) [21], Distributed Energy-Efficient Clustering (DEEC) for heterogeneous WSNs [22], Stable Election Protocol (SEP) [23], and other clustering routing algorithms are deployed in SDWSN. The SDWSN exhibits strong scalability and robustness because its cluster routing algorithm divides sensor nodes into multiple clusters. Each cluster has a cluster head that collects information from its members and sends it to the base station. LEACH randomly selects cluster heads to balance network load. However, polling the cluster heads frequently consumes power [24]. When choosing cluster heads, DEEC considers the remaining energy of the nodes, prolonging the network lifecycle and increasing the data transmission delay.

The clustering step of the clustering routing method, an approach to solving difficult optimization problems by modeling the behavior of groups, has recently been improved by applying some clever optimization algorithms. The Artificial Bee Colony Algorithm (ABC) [25], the Firefly Algorithm (FA) [26], and the Genetic Algorithm (GA) [24] are examples of intelligent optimization algorithms. The GA local search performance is poor despite its fast convergence time. The FA has fewer parameters than GA. Global search and convergence speed are better for the ABC method. Methods that are easy to integrate with clustered routing algorithms include the FA and ABC algorithms. Intelligent optimization techniques suffer from poor solution accuracy, convergence to locally optimal solutions, and limited search space.

The main contributions of this paper are as follows.

1. Establish the network model and energy consumption model of SDWSN.

2. A hybrid optimization algorithm (optimized algorithm based on the firefly algorithm, gravity search algorithm, and biogeography optimization, FGB) was designed. To

improve the capability of a worldwide search for the best solution, a location update approach based on population diversity is first presented and is based on FA. Then, based on the hybrid gravitational search algorithm (GSA) and biogeographic-based optimization (BBO), the ability of local search to find the optimal solution is optimized. The algorithm improves the performance of optimization procedures that find approximate global solutions.

3. A distributed high-efficiency entropy energy-saving clustering routing algorithm for SDWSN is designed (DHEEC). This algorithm considers energy entropy when selecting cluster heads, which is more suitable for energy updates in SDWSN. At the same time, FGB is used for cluster head selection to speed up the rotation of cluster heads.

4. Through experimental simulation and analysis, compare FGB with other algorithms to verify its performance on standard functions. Comparing DHEEC with other algorithms, it prolongs the network lifetime and improves energy utilization.

The remaining of this this paper is organized as follows. In Section 2, the literature review is discussed in detail. In Section 3, the proposed system model is described and the analytical formulations are discussed. In Section 4, the hybrid optimization algorithms are explained. In Section 5, the high entropy energy saving clustering routing algorithm is described. In Section 6, the simulation experimentations and comparative performance evaluation is discussed. Finally, Section 7 concludes the paper.

## 2 Related work

This section mainly focuses on two aspects of the network clustering routing algorithm and the intelligent optimization algorithm.

A three-stage heterogeneous network model for WSN is proposed in [27]. The model uses a threshold function and weighted election probabilities to select cluster heads and capture network heterogeneity. Based on the evolutionary approach for LEACH protocol optimization (LEACH-O) [28], both WSN lifetime and energy efficiency are improved. The Enhanced Adaptive Distributed Energy Saving Cluster (EADEEC) algorithm [29] based on DEEC improves cluster stability, node survival time, and data transmission. The authors in [30] uses particle swarm optimization (PSO) to estimate the relative positions of cluster heads and base stations, developed mobile paths for base stations based on the ABC traversal path approach, and reduced sensor node consumption power.

The clustering phase of the clustering routing method uses a population optimization algorithm. In [31], an energy-efficient heterogeneous WSN routing technique based on Improved Firefly Clustering (IFCEER) was developed. However, it has poor solution accuracy and slow convergence are also problems. The WSN routing method based on FA for optimizing fuzzy C-means (FFACM) is a good reference to prevent the algorithm from tuning to local optima [32]. This is due to incorrect initial placement of the clustering centers. The authors in [33] considered the Bayesian PSO algorithm that is based on the Bayesian principle probability strength function for parameter tuning. However, especially complex multi-minimal problems face local optima. The Hybrid Firefly and Particle Swarm Optimization Algorithm (HFPSO) [34] checks the current global best fit value and decides to start the search process. In [35], a hybrid firefly algorithm with particle swarm optimization (HFAPSO) is proposed. This algorithm uses the HFPSO for optimal cluster head selection on LEACH-C and improves the FA's global search ability. Reference [36] applied FA to a mixed-state logic model with an energy parameter to detect distributed denial-of-service (DDoS) attacks. Reference [37] used FA and BBO for feature selection in software product lines. In [38], the Gray-Wolf optimization

(GWO) algorithm was combined with PSO to significantly reduce the computational time required to implement the algorithm.

The LEACH-G protocol was proposed in [39], where the calculation of the ideal cluster head size is based on area and number of nodes. The enhanced LEACH protocol should theoretically outperform the original LEACH in terms of network robustness and power consumption. The R-LEACH is a clustering strategy designed based on the remaining energy of nodes and the maximum number of cluster heads [40]. The probability threshold is redefined by the ideal number of cluster heads to improve the network speed and reduce the average power consumption. To achieve network load balancing, reference [41] provides a power-aware routing algorithm that creates timeouts according to the remaining power of a node and controls how often a node is elected as cluster leader. Each node determines its own attribute value based on the distance from the base station and the amount of energy left over, and based on this, it competes with neighboring nodes to obtain the best cluster head selection. Reference [42] suggested a distributed energy-saving clustering routing method. In [43], the authors found that if each cluster region remains fixed after establishment, the cluster head election is performed in each cluster region based on the residual energy factor, and if the cluster reaches the head energy, proposed an improved low-energy fixed clustering algorithm. If the threshold is below, a node from the cluster her members is randomly selected for the next voting round. This algorithm is efficient and prolongs network life. Reference [44] proposes a modified DEEC (MDEEC) algorithm that extends the network lifecycle but increases the amount of data received by the base stations. Additionally, a low-power communication mechanism was established within the cluster to allow nodes with residual power above a threshold to compete as cluster head candidates. Reference [45] proposed an improved protocol called LEACH-LOMUC, which selects candidate cluster heads based on remaining energy, size of cluster heads and distance between clusters, cluster head and base station transmission line, which effectively saving network energy. This protocol overcomes the shortcomings of the LEACH algorithm, which does not take into account the residual energy, the distribution of the cluster head, and the direct communication between the cluster head and the base station. Reference [46] proposed a distributed clustering routing protocol based on dynamic partition load balancing to solve the "hot zone" problem caused by uneven load on the nodes. Distributed clustering routing protocol determines different cluster radius based on the distance between the cluster head and base station, thus prolonging the life of the network. Reference [47] suggested that a highly efficient relay and relay node selection scheme designed for farmland environments to solve the problem of high power consumption caused by direct communication between the cluster head node is located far away from the base station and the receiver node. The node's residual energy, the density of the network architecture, and the distance are carefully considered when choosing a cluster leader. The forwarding methods in different communication models are described one by one, starting with the distance between the candidate forward node and the sink node during the relay node selection process. The energy efficiency of the network is increased when energy balance is achieved. By setting the fit function to remaining energy and the distance from the base station, references [48,49] improved the weights of the first three wolves in the original GWO based on price suitable treatment. They also designed an improved algorithm based on the Physical Interest-Based Advanced GWO (FIGWO), which allows for optimal election of cluster leaders. Reference [50] has improved the particle swarm optimization algorithm based on adaptive learning factors, determined the adaptive function based on the node energy factor and the position equilibrium factor, and proposed two transmission methods. The throughput (single-hop and multi-hop) is optimized according to the distance between the cluster head and base station, while balancing the energy consumption of

the nodes. Reference [51] proposed a heterogeneous clustering technique based on ant colony optimization to solve the problem of heterogeneity. This algorithm used the concept of pheromone update primacy to optimize the path quality and significantly improve the network quality. The optimal cluster head is selected by evaluating the relationship between the random number and the attribute and the weight parameters are optimized based on the GA and the simulated incubation algorithm. Reference [52] proposed a low-power routing protocol, where the node attribute is defined around the distance between the node and the base station and the remaining power factor. Reference [53] proposed a hybrid heterogeneous clustering routing protocol based on fuzzy logic theory and ant colony optimization. Based on the ant colony optimization method, the heterogeneous cluster regions are created, allowing efficient transmission of monitoring data from the cluster head to the base station, thereby effectively reducing the power consumption of the network. For the purpose of balancing the energy consumption of the network, showing the relationship between the cluster head ratio and the network size, and designing the suitable function for the particles of the swarm optimization algorithm on the energy remaining and node density, the improved filtering based on the PSO algorithm is proposed in [54], which greatly prolongs the life of the network. Using a GWO and a unique fit function, reference [55] proposed a new clustering protocol to predict the actual power consumption of multiple clustering schemes, giving allows reuse of a particular clustering method. A two-way communication network is established between the peripheral cluster leader and the receiving node to prevent premature death of nodes far from the base station. This network effectively balances the power consumption between the cluster head and the relay node, while reducing the overall network cost.

Reference [56] proposed a centralized clustering technique (LEACH-C), which would improve the existing LEACH protocol. The base station selects the best cluster head in the cluster head election step using a simulated annealing process. To solve the problem of random selection of cluster leaders using the LEACH protocol, the LEACH-C takes into account the node's position and remaining energy when selecting the cluster leader. However, due to one-hop communication between the cluster heads and base stations, the energy consumption of cluster heads far from the base station will increase, eventually leading to their premature collapse and network lifetime. Reference [57] proposed a centralized cluster routing protocol in which the base station groups the nodes for each round of cluster leader election using averaging technique and the GWO selects the best cluster leader for each cluster. By improving the cluster structure, it effectively increases the lifetime of the network. However, because this protocol does not take into account the routing between clusters, the cluster head located far from the base station will die prematurely due to high power consumption, reducing the lifetime of the network. References [58] proposed a centralized clustering routing system based on K-Means. The base stations use K-Means clustering to create the network clusters during network startup. The nodes disperse the cluster heads and remaining energy in each cluster. The K-Means clustering technique is sensitive to the starting point and simple to converge to the local optimal point, which has an impact on the clustering effect. However, it prolongs the life of the network to some extent. Furthermore, the protocol does not take routing between clusters into account, and the single-hop technique increases the energy consumption of the cluster head when it is far from the base station. This places constraints on the system. The ABC-SD protocol, which uses the artificial swarm method to solve the clustering and routing problems, has been presented in [59]. By taking into account the remaining energy of the nodes and neighboring data, this protocol creates an efficient matching function that is used to evaluate the results of the artificial swarm method. The clustering problem can be solved depending on the quality of the solution. The protocol builds routing between clusters based

on the loss function after considering the energy efficiency and the number of hops in the routing hop between the clusters. The ABC-SD effectively extends the network lifetime and increases the network throughput by implementing centralized clustering and distributed route mix generation techniques. However, the load balancing between cluster heads, which affects network performance, is not taken into account by this protocol. A technical improvement based on the clustering routing protocol of the PSO algorithm has been proposed in [60]. A new matching function is designed by considering the redundant energy and the geographical location of the node in the grouping step to select the group leader who is near the base station and has a high level of excess energy. This fitness function is then used to evaluate the quality of the solution generated by the swarm algorithm. Each cluster head is paired with a relay node in the routing step between the clusters to balance the network energy consumption. The network lifetime is effectively extended by this protocol. However, there is no guarantee that the selected forward and cluster leaders are the current optimal option due to some features of the swarm optimization process itself. Reference [61] suggested a software-defined network routing method based on fuzzy logic. It was used as the basis for implementing the Fuzzy Topology Discovery Protocol (FTDP). It prolongs the life of the network and reduces the packet loss rate. To increase the lifetime of the network and reduce the latency, reference [62] developed an intelligent routing method based on the ABC algorithm that maps potential routing links as food sources and takes into account energy and delay variables. The Fuzzy Ant Colony Optimization Routing (FACOR) was developed by [63] by incorporating the ant colony algorithm into the routing protocol, evaluating the link viability by using fuzzy calculations based on energy, distance and connectivity, then choose the optimal link. A metaheuristic search routing protocol based on GA and mutation operator is presented in [64]. To develop effective routing and load balancing strategies, reference [65] introduced the load balancing and routing strategies using the Glowworm swarm optimization approach (LBR-GSO). Using the properties of a dynamic network, reference [66] suggested an improved ant colony algorithm (IACO), which can be used to generate sensor and rule node propagation functions. The network is clustered heterogeneously in [67], and the smaller the distance from the sink node, the smaller the cluster size. This reduces communication power loss in the cluster and provides the cluster main node with more power to transfer data between clusters, balancing the network power consumption and successfully avoiding the main problem. Reference [68] proposed a WSN hierarchical routing technique based on fuzzy cluster C-means and swarm intelligence (FCM-SI). The optimal cluster leader is selected using an artificial swarm algorithm, and then establishes a multi-hop path to the base station using the ant colony method. Although the power consumption and load balancing of the network are taken into account, the effect of the random starting cluster center of the fuzzy C-means algorithm is not taken into account. This can easily lead to locally optimal clustering, which will affect the entire network. An energy-aware multi-hop routing (EAMR) method is proposed in [69]. During the setup phase, a permanent cluster is created by selecting the initial cluster leader, selecting the first relay nodes, and determining the cluster membership of the remaining sensor nodes. The data is passed through the relay node during this phase, thus balancing the load on the cluster. However, the power consumption of the fixed burst far away from the base station is significantly higher than that of the fixed burst near the base station as the amount of data transmission increases. The clusters cannot evenly distribute network load. Reference [70] proposed effective and reliable routing algorithm based on dempster-shafer proof theory (DS-EERA) in which the attribute index is established as proof, using weighted method and the number of entropy to determine the weight of the index. Then, apply the merge rule of the theoretical DS proving to the underlying probability distribution function of each merged index value and the next hop chosen to propagate the data. Using the fuzzy C-

**Table 1. Performance comparison of algorithms.**

| Reference | Target detection | Feature selection | Dataset | Machine learning technique | SDN | IoT | UDP |
|---|---|---|---|---|---|---|---|
| [51] | DoS, R2L, Probe, U2R | Information gain | NSL-KDD | Support vector machine | X | X | ✓ |
| [52] | DoS, R2L, Probe, U2R | Mutual information | NSL-KDD | Support vector machine | X | X | ✓ |
| [53] | DDoS | - | NSL-KDD99 | Support vector machine | X | ✓ | X |
| [54] | DoS, R2L, Probe, U2R | Information gain | NSL-KDD99 | XGBoot | X | X | ✓ |
| [55] | DoS, R2L, Probe, U2R | Regularized random forest | NSL-KDD99 | Naïve Bayes | X | X | ✓ |
| [56] | Normal and attack | Stacked based feature selection | NSL-KDDCup99 | Gated Recurrent Unit | X | X | ✓ |
| [57] | DDoS | - | GSB, IBRL | Random Forest | X | X | ✓ |
| [58] | DDoS | Principal component analysis | PDG, LUCE | Q-Learning | X | X | ✓ |
| [59] | DDoS | Information gain | KDD99 | Online Naïve Bayes | X | ✓ | ✓ |
| [60] | DoS, R2L, Probe, U2R | Stacked based feature selection | KDD99 | Random Forest | X | X | ✓ |
| [61] | Spike, drift, constant, noise | - | NSKL-DD | Linear Weighted Projection Regression | X | X | ✓ |
| [62] | Spike, drift, constant, noise | Information gain | Simulated data | Ensemble Learning | X | X | ✓ |
| [63] | Normal and attack | Chi-square information gain | KDD99 | Passive Aggressive Algorithm | X | X | ✓ |
| [64] | DDoS | Mutual information | NAMOS, GSB | Stack Ensemble Learning | X | ✓ | ✓ |
| [65] | DoS | Information gain | KDD99 | Leat-Square SVM | X | X | ✓ |
| [66] | DoS | Regularized random forest | UNSW-NB15 | Random Forest | ✓ | X | ✓ |
| [67] | DoS | Mutual information | KDD99 | Q-Learning | ✓ | X | X |
| [68] | DoS | - | Simulated data | Ensemble Learning | ✓ | X | ✓ |
| [74] | DoS | Information gain | NSL-KDD | Naïve Bayes | X | ✓ | ✓ |
| [75] | Spike, drift, constant, noise | - | KDD99 | Support Vector Machine | X | X | ✓ |
| [76] | DDoS | Mutual information | KDD99 | FCM | ✓ | ✓ | X |
| [77] | Spike, drift, constant, noise | - | NSL-KDD | Random Forest | X | X | ✓ |
| [78] | DoS, R2L, Probe, U2R | - | KDD99 | ANN | X | ✓ | ✓ |
| [79] | DDoS | Regularized random forest | KDD99 | SVM | X | ✓ | ✓ |
| [80] | DoS | - | GSB, PDG | Reinforcement learning | X | ✓ | X |
| [81] | Normal and attack | - | KDD99 | Q-Learning | X | ✓ | X |
| Proposed | DDoS | Population diversity | GSB, LUCE, IBRL, PDG, NAMOS | FA, GSA | ✓ | ✓ | ✓ |

means algorithm for clustering and the particle swarm optimization algorithm to optimize the initial center of the cluster, reference [71] proposed a clustering routing algorithm that ignores the fuzzy C-means and sensitivity to the center of the original cluster.

Table 1 summarizes the performance of various state-of-the-algorithms and the proposed method in terms of important metrics.

## 2.1 Problem statement

In summary, the improved algorithm based on the traditional clustering routing protocol usually selects the cluster heads based on variables such as residual energy and distance, but it has the disadvantage of alternately selecting the clusters based on the probability and determine the number of key clusters spontaneously. The impact of cluster head position on node power consumption is often overlooked, despite the fact that some researchers try to delay the cluster head rotation as much as possible by setting a threshold.

Aiming at the above problems, in order to improve the efficiency of the optimization process for identifying an approximative global solution, a hybrid optimization algorithm that combines the global optimization powers of FA with the local optimization capabilities of BBOGSA is created. A distributed high-efficiency entropy energy-saving clustering routing system is created to speed up the rotation of the cluster heads. In order to show how well the hybrid optimization approach performs, it is compared to similar algorithms in terms of convergence accuracy, running time, and stability through experimental simulation and analysis. It is compared to competing techniques to show how the distributed high-efficiency entropy energy-saving clustering routing algorithm may lengthen network life and reduce data transmission energy usage.

## 3 System model

### 3.1 Network model

In the architecture of SDWSN, the base station is located at the control layer, including the controller module that calculates the route of the entire network. The sensor nodes are located in the data layer and realize the function of forwarding data according to the flow table [72–74]. According to the SDWSN architecture, the SDWSN model is shown in Fig 1, and the control layer has the following functions: to obtain information such as network topology, the remaining energy of sensor nodes, and location. Calculate the cluster head rotation table according to the energy, the optimal path to meet the demand, and load the DHEEC and cluster head rotation table into flow table information. The cluster head located in the data layer only completes receiving, merging, and forwarding data [75,76].

The network model is based on the following assumptions:

1. The base station is situated in the middle of the $M \times M$ area, and the sensor nodes $s_i$ ($i = 1,2, \ldots, N$) are dispersed at random [77–79];
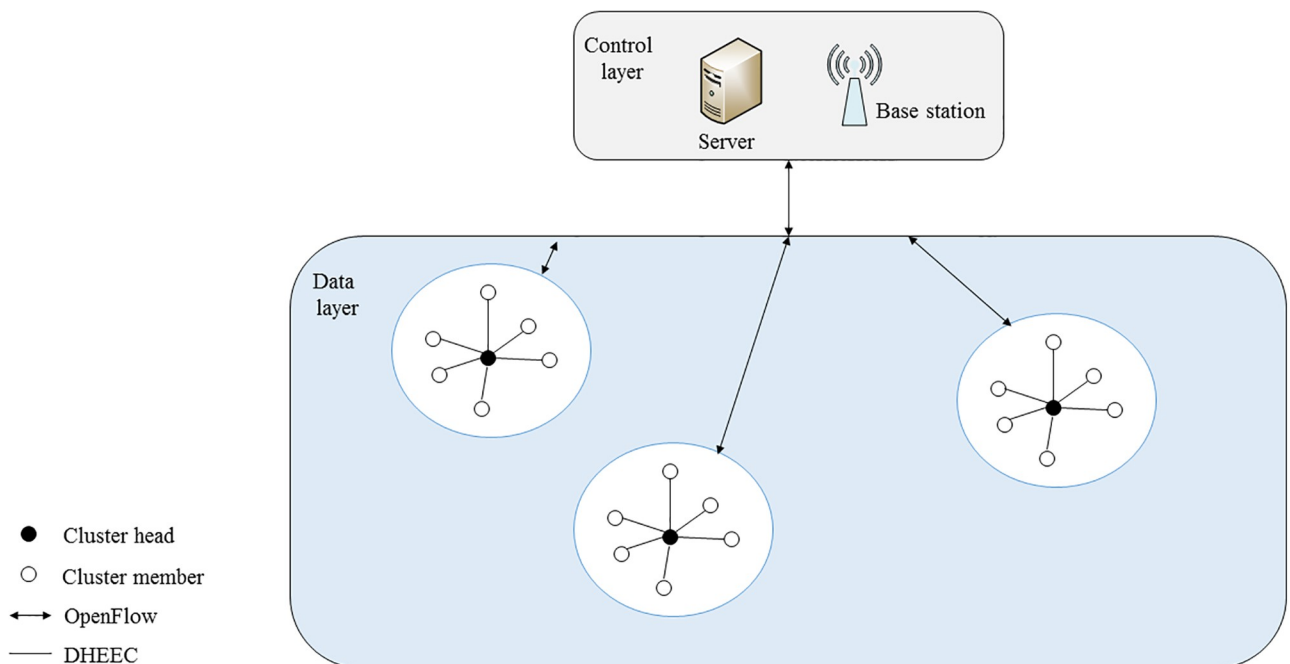


**Fig 1. Proposed network model of SDWSN.**

https://doi.org/10.1371/journal.pone.0301078.g001

2. Data communication between the cluster head and base station using the OpenFlow protocol [80–82];

3. Node software and hardware configurations are the same, each has a unique ID, and the location is fixed;

4. The links of the network are symmetrical;

5. The base station's data transmission and calculation speeds are faster than those of the sensor nodes.

## 3.2 Energy consumption model

The radio network energy consumption model is shown in Fig 2.

According to the energy consumption model, the energy $En_{Tx}$ of transmitting a $l$ bit packet is:

$$En_{Tx} = \begin{cases} l En_{elec} + dis^2 l \varepsilon_{fs}, & dis < dis_{break} \\ l En_{elec} + dis^4 l \varepsilon_{mp}, & dis \geq dis_{break} \end{cases} \tag{1}$$

Among them, $En_{elec}$ is the unit energy consumption, $\varepsilon_{fs}$ and $\varepsilon_{mp}$ are the power amplification loss, dis is the distance [83,84], and $dis_{break}$ is the threshold distance.

The energy $En_{Rx}$ required to receive a $l$ bit data packet is:

$$En_{Rx} = l En_{elec} \tag{2}$$

Therefore, the total energy consumed in one iteration is:

$$En_{Round} = En_{Tx} + En_{Rx} = l_D \left( 2N En_{elec} + N En_{DA} + k \varepsilon_{mp} dis_{HB}^4 + k \varepsilon_{fs} dis_{MH}^2 \right) \tag{3}$$

Among them, $k$ is the number of clusters [85,86], $dis_{HB}$ is the average distance between the cluster head and the base station, and $dis_{MH}$ is the average distance between the cluster member node and the cluster head node:

$$\begin{cases} dis_{HB} = \dfrac{0.765}{2} M \\ dis_{MH} = \dfrac{M}{\sqrt{2\pi k}} \end{cases} \tag{4}$$
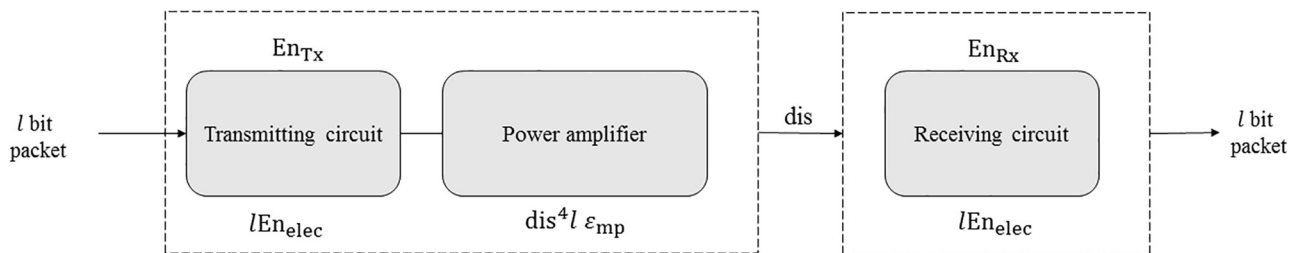


**Fig 2. Energy consumption model.**

https://doi.org/10.1371/journal.pone.0301078.g002

## 4 Hybrid optimization algorithm

### 4.1 Improved firefly algorithm

FA is an algorithm inspired by the flickering behavior of fireflies, which iterate their positions according to the brightness to find the optimal solution. The following rules are used:

1. Fireflies attract one-way according to the brightness;

2. The attraction of fireflies is proportional to the brightness;

3. Fireflies with low brightness are attracted to fireflies with high brightness;

4. The firefly with the highest brightness moves randomly.

**Definition 1**: The relative fluorescence brightness $I$ of fireflies $i$ and $j$ is:

$$I = I_0 e^{-\beta r_{ij}} \tag{5}$$

Among them, $I_0$ is the maximum fluorescence brightness of fireflies, $\beta$ is the light intensity absorption coefficient [87,88], $r_{ij}$ is the Euclidean distance, and the calculation formula is:

$$r_{ij} = \|c_i - c_j\| = \sqrt{\sum_{w=1}^{D} \left(c_{iw} - c_{jw}\right)^2} \tag{6}$$

Among them, $c_i$ and $c_j$ represent the spatial positions of fireflies $i$ and $j$, and $c_{iw}$ and $c_{jw}$ are the $w$-th dimension coordinates of fireflies $i$ and $j$.

**Definition 2**: Mutual attraction $\gamma$ between fireflies. $\gamma$ corresponds to the relative fluorescence brightness between fireflies:

$$\gamma = \gamma_0 e^{-\beta r_{ij}^2} \tag{7}$$

Among them, $\gamma_0$ is the maximum attractiveness.

However, the convergence speed of FA is slow, and the global search lacks randomness, which may lead to convergence to a locally optimal solution [89,90]. The average distance between firefly individuals and ideal firefly individuals is used in this study to quantify population diversity and to suggest a location update technique based on population diversity.

**Definition 3**: Diversity index $\beta^n$. Reflects the diversity of the $n$-th generation population:

$$\beta^n = \frac{1}{|s|} \sum_{i=1}^{|s|} \sqrt{\sum_{j=1}^{N} \left(C_{ij} - \bar{C}_Y\right)^2} \tag{8}$$

where $s$ stands for population size, and $\bar{C}_Y$ denote the population's mean center.

Further adjust the position update strategy of fireflies, as shown in formula (9):

$$\begin{aligned}
c_i^{n+1} &= c_i^n + \gamma_0 e^{-\beta r_{ij}^2}\left(c_j - c_i\right) + \alpha \times \text{rand} \\
&= c_i^n + \gamma_0 e^{-\beta r_{ij}^2}\left(c_j - c_i\right) + \rho\left(c_i^n - C_{\text{best}}^n\right) + \alpha \times \text{rand}
\end{aligned} \tag{9}$$

Among them, $c_i^n$ represents the spatial position of firefly $i$; $\alpha \in [0,1]$ represents the step factor; random parameter rand $\in [0, 1]$. $C_{\text{best}}^n$ represents the position of the brightest firefly individual in the current iteration n rounds [91–93]; $\rho$ is a weight that changes according to $\beta^n$ and

the number of iterations, and its calculation formula is:

$$\rho = \begin{cases} -k\sigma, \ \beta^n \leq \sigma\beta^0 \\ 0, \beta > \sigma\beta^0 \end{cases} \tag{10}$$

Among them, $\beta^0$ is the initial population diversity index, k is the current iteration number, $\sigma$ is a linear decreasing function, and its calculation formula is:

$$\sigma = \frac{\text{maxiterate} - k}{\text{maxiterate}} \tag{11}$$

Among them, maxiterate is the maximum number of iterations.

According to formula (11), the initial value of $\sigma$ is 1, and when $k = 1$ in the first iteration, the value of $\rho$ obtained by formula (10) is negative. Move in random directions away from the optimal solution, thus enabling a wider search [94]. Fireflies will do local random searches near the ideal solution to provide a more accurate local search as the number of iterations rises and the diversity index falls. In order to prevent the algorithm from converging to the local optimal solution, such a dynamic adjustment mechanism can balance the issues of global optimization in the early stages of the algorithm and local optimization in the later stages.

The fixed step size may make it difficult to reliably converge to the optimal point as the number of iterations rises since most fireflies would do local optimal searches in the vicinity of the optimal point. As a result, the step size factor is dynamically modified in the manner shown below:

$$\alpha = \alpha_0\sigma \tag{12}$$

where $\alpha$ is a constant value in FA. $\alpha_0$ represents the initial step size factor.

The specific steps of firefly algorithm are shown in Algorithm 1, and its steps are described as follows.

**Algorithm 1. Improved firefly algorithm.**

**Input:** population size $N$, initial step factor $\alpha$, initial population diversity index $\beta^0$, maximum number of iterations maxiterate
**Output:** optimal solution $C_{\text{best}}$
```
 1: Determine objective function f(x)
 2: Create initial population C₀
 3: Generate light intensity
 4: while k < maxiterate do
 5: Determine attraction γ
 6: for i ∈ [1, N] do
 7: for j ∈ [1, N] do
 8: Determine σ, β
 9: if Iⱼ > Iᵢ then
10: Move firefly i towards j
```
11: $c_i^{n+1} = c_i^n + \gamma_0 e^{-\beta r_{ij}^2}\left(c_j - c_i\right) + \rho\left(c_i^n - C_{\text{best}}^n\right) + \alpha \times \text{rand}$
```
12: end if
13: Change attraction
14: Search again and modify light intensity
15: end for
16: end for
17: Determine the best solution C_best
18: end while
```
Algorithm 1 is explained below stepwise.

Step 1: Initialize parameters, population size $N$, initial step factor $\alpha$, initial population diversity index $\beta^0$, maximum iteration number maxiterate, etc.

Step 2: Construct the initial population $C_0$, and calculate the brightness [95], sort according to the brightness, calculate the fitness $I$ and the current global optimal solution $C_{best}$ according to the maximum fluorescence brightness $I_0$.

Step 3: Update the distance between fireflies according to formula (9), update the attraction between fireflies according to formula (7), and update the global optimal solution $C_{best}$ at the same time.

Step 4: Add 1 to the number of iterations, and update the step factor according to formula (12).

Step 5: If the number of iterations reaches maxiterate, the algorithm stops and outputs the current global optimal solution $C_{best}$, otherwise, execute steps 2 to 4 in a loop.

## 4.2 Gravity search and biogeography hybrid algorithm

BBO regards the solutions in the population as habitats, assigns a habitat suitability index (HSI) to each habitat to reflect the overall suitability, and updates the optimal solution through the population migration operation. Habitats are similar to firefly individual locations in FA [96], and the general rules used by BBO are as follows:

1. Brightness is proportional to the HSI value of the habitat;

2. Individuals in habitats with high HSI values migrate to habitats with low HSI values;

3. Habitats with low HSI values attract individuals from habitats with high HSI values;

4. Habitats face random variation among individuals.

**Definition 4**: Habitat $H$. Habitat provides a solution within the search space of a numerical optimization problem.

**Definition 5**: Habitat Suitability Index HSI. Reflect the overall suitability of the habitat.

**Definition 6**: Suitability index variable (SIV). The HSI is influenced by additional coefficients, such as rainfall, vegetation diversity, and temperature.

**Definition 7**: Emigration index $\mu$. Controlling habitat migration, the probability of a species leaving a habitat is proportional to the number of species in the habitat.

**Definition 8**: Immigrant index $\lambda$. Controlling for habitat migration, the migration index is maximized when the habitat is free of species.

The BBO starts from a random population with population size $N$ and dimension $D$, and each individual $H_{ij}$ is generated according to formula (13).

$$H_{ij} = a_j + rand \times \left( b_j - a_j \right) \tag{13}$$

Among them, $i = \{1, 2, \ldots, N\}$, $j = \{1, 2, \ldots D\}$, $a_j$ and $b_j$ are the upper and lower bounds of the $j$th dimension of the solution vector respectively.

To calculate HSI, sort population individuals $H_k$ in order from good to bad, and calculate emigration index $\mu_k$ and immigrant index $\lambda_k$ according to formula (14).

$$\begin{cases} \mu_k = E \times S_k/S_{max} \\ \lambda_k = (1 - S_k/S_{max})I \end{cases} \tag{14}$$

Among them, $S_k$ denotes the number of individuals in the current population, $S_{max}$ represents the number of individuals in the largest population, $E$ represents the maximum immigrant emigration rate, and $I$ represents the maximum immigrant immigrant rate. The larger $\mu_k$ is, the smaller $\lambda_k$ is, and the higher the HSI is. The smaller $\mu_k$ is, the larger $\lambda_k$ is, and the lower the HSI. In each iteration, the better solution is propagated to the worse solution. The poorer solution has a high probability of learning from the better solution. Overall, as the number of iterations increases, all solutions approach the optimal solution.

Habitats with higher and lower HSIs are prone to mutation, while solutions in the middle are left untouched. The rate of variation is:

$$m_i = M \times \left[ \frac{1 - P_i}{P_{max}} \right] \tag{15}$$

Among them, $M$ is the maximum mutation rate, $P_i$ is the species probability, and $P_{max}$ is the maximum species probability, where $P_{max} = N$.

Variation is caused by unexpected events that lead to changes in the number of species and HSI in the habitat, and the variation rate of species depends on the probability of species. When the number of species is large or small, the species probability is low and the variation rate is high; when the number of species is moderate, the species probability is high and the variation rate is low. The purpose of the mutation is to increase the diversity of the population so that the habitats with low HSI values are improved, the habitats with high HSI values are improved, and the habitats with high HSI values may change to low HSI values. The mutation is conducive to expanding the search space and jumping out of local optimum.

The gravitational force $F$ between objects and the acceleration $a$ of the individual are respectively:

$$\begin{cases} F = G\frac{M_1 M_2}{R^2} \\ a = \frac{F}{M} \end{cases} \tag{16}$$

Among them, $M_1$ and $M_2$ are the mass of the object, $G$ is the gravitational constant, $R$ is the Euclidean distance, and $M$ is the mass.

According to GSA, the gravitational force $f_{ij}$ between population individuals $i$ and $j$ and the acceleration ai of individuals are respectively:

$$a_i = \frac{F_{ij}}{M_{ii}} \tag{17}$$

Among them, $M_{aj}$ and $M_{pi}$ represent the active gravitational mass of population individual $i$ and the passive gravitational mass of population individual $j$ respectively, and $M_{ii}$ is the inertial mass of population individual $i$.

The inertial mass of population individuals is related to fitness. The larger the inertial mass, the greater the probability of producing an optimal solution. The mass $M_i(t)$ of the population at time $t$ is calculated as follows:

$$\begin{cases} M_{ai} = M_{pi} = M_{ii} = M_i, \ i = 1, 2, \ldots, N \\ m_i(t) = \text{fit}_i(t) - \text{worst}(t)/\text{best}(t) - \text{worst}(t) \\ M_i(t) = m_i(t)/\sum_{j=1}^{n} m_j(t) \end{cases} \tag{18}$$

Among them, $\text{fit}_i(t)$ is the fitness value, best$(t)$ is the optimal solution, worst$(t)$ is the worst solution, and the calculation formula is:

$$\begin{cases} \text{best}(t) = \max_{i \in \{1,2,\ldots,N\}} \text{fit}_i(t) \\ \text{worst}(t) = \min_{i \in \{1,2,\ldots,N\}} \text{fit}_i(t) \end{cases}$$

The solution set of the BBO algorithm is:

$$m_{ti} = [(I_{i1} + E_{i1}), (I_{i2} + E_{i2}), \ldots, (I_{iD} + E_{iD})] \tag{19}$$

BBO avoids falling into the local optimal solution and premature algorithm through migration and mutation operations, and improves the overall stability of the population; that is, it improves the value of HSI. According to HSI, the fitness function is created as follows:

$$\begin{aligned} \text{fit}_i(t) &= \text{HSI} = \text{SIV}_i + N + \lambda_k - \mu_k \\ &= \text{SIV}_i + N + \left[\frac{1 - S_k}{S_{\max}}\right] I - \frac{ES_k}{S_{\max}} \\ &= \text{SIV}_i + N + \left[\frac{1 - S_k(I + E)}{S_{\max}}\right] \\ &= \text{SIV}_i + N + \left[\frac{1 - S_k M_t}{S_{\max}}\right] \end{aligned} \tag{20}$$

Update formula (18) to get:

$$m_i(t) = \frac{S_{\max}[\text{SIV}_i + N + I - S_k M_t - \text{worst(t)}]}{S_{\max}[\text{best(t)} - \text{worst(t)}]} \tag{21}$$

Update the speed and position of the individual according to formula (17):

$$v_i^d(t + 1) = \text{rand}_i \times v_i^d(t) + a_i^d(t) \tag{22}$$

$$x_i^d(t + 1) = v_i^d(t) + v_i^d(t + 1) \tag{23}$$

The pseudocode of BBOGSA is shown in Algorithm 2, and its steps are described as follows.

**Algorithm 2. BBOGSA algorithm.**

**Input:** population size $N$, dimension $D$, maximum immigrant emigration rate $E$, maximum immigrant immigrant rate $I$, maximum mutation rate $M$, maximum number of iterations $r_{\max}$

**Output:** Optimal solution $G_{\text{best}}$
```
 1: Determine objective function f(x) = C_best
 2: Generate initial population C_0
 3: while k < r_max do
 4: Determine HIS
 5: for i ∈ [1, N] do
 6: for j ∈ [1, N] do
 7: Calculate μ_k, λ_k, M
 8: if M_j > M_i
 9: Move ith habitat toward jth
10: HSI = SIV_i + [ (I−S_kM_t)/S_max ]
11: end if
12: Change HIS
13: Search and modify again gravitational
```

```
14: end for
15: end for
16: Determine the optimal solution Gbest by sorting HIS
17: end while
```

The steps of the Algorithm 2 are explained below.

Step 1: Set parameters such as the initial population size $N$, dimension $D$, maximum immigrant emigration rate $E$, maximum immigrant immigration rate $I$, maximum mutation rate $M$, and maximum number of iterations $r_{max}$.

Step 2: Construct the initial population $C_0$, calculate the suitability index HSI, sort and update the current global optimal solution $G_{best}$.

Step 3: Calculate the migration index $\mu_k$ and the immigration index $\lambda_k$ of the habitat according to the formula (14), and perform the migration operation, and perform the mutation operation according to the formula (15).

Step 4: Calculate the inertial mass of each population according to formula (21), and update the individual position according to formula (23).

Step 5: If the number of iterations reaches $r_{max}$, the algorithm stops and outputs $G_{best}$, otherwise, execute steps 2 to 4 in a loop.

## 4.3 Hybrid optimization algorithm design

FA tends to converge to the local optimal solution in the final iteration, which makes it unable to perform a proper local search. The local search ability of BBOGSA makes up for the defect of FA. Therefore, FGB is designed to combine the global search ability of FA and the local search ability of BBOGSA to improve the performance of the optimization process for finding approximate global solutions.

FGB has two phases. In the first stage, the improved FA performs preliminary optimization to search the design search space by performing a limited number of iterations, and the optimal solution found in this stage is $C_{best}$. In the second stage, a more refined search of $C_{best}$ is performed using BBOGSA algorithm for a limited number of iterations.

The initial population constructed by the improved FA is randomly distributed in the whole space, while in BBOGSA, $C_{best}$ is directly transformed into the optimal initial population.

The FGB process is shown in Fig 3.

The pseudocode of FGB is shown in Algorithm 3.

```
Algorithm 3. FGB algorithm.
Input: L, N, α, β⁰, maxiterate, D, E, I, rmax
Output: Optimal solution Gbest
 1: Determine objective function f(x)
 2: Create initial population C0
 3: Generate light intensity
 4: while k < maxiterate do
 5: Determine γ
 6: for i ∈ [1, N] do
 7: for j ∈ [1, N] do
 8: Determine β, σ
 9: if Ij > Ii then
10: Move ith firefly towards jth
```
11: $c_i^{n+1} = c_i^n + \gamma_0 e^{-\beta r_{ij}^2} \times \left( c_j - c_i \right) + \rho \times \left( c_i^n - c_{best}^n \right) + \alpha \times \text{rand}$

```
12: end if
13: Change γ
14: Determine light intensity again and modify it
15: end for
16: end for
17: Determine the optimal solution Cbest
18: end while
19: Determine objective function f(x) = Cbest
20: Create initial population C0
21: while k < rmax do
22: Determine HIS
23: for i ∈ [1, L] do
24: for j ∈ [1, L] do
25: Determine μk, λk, M
26: if Mj > Mi
27: Move ith habitat towards jth
```
28: Calculate $\text{HSI} = \text{SIV}_i + N + \left[\frac{I - S_k M_t}{S_{\max}}\right]$
```
29: end if
30: Change the variable HIS
31: Modify gravitational by searching again
32: end for
33: end for
34: Determine the optimal solution Gbest by sorting HIS
35: end while
```

## 5 High entropy energy saving clustering routing algorithm

The running time sequence of DHEEC is shown in Fig 4, with rounds as the execution cycle, including the start-up state and steady state. Before the first round of iteration, the base station sends flow table information and forwards it to all sensor nodes, while at the same time indicating the start. In the initial stage of a startup, SDWSN has two important functions: (1) selecting the cluster head; (2) establishing a cluster. In (1), the temporary cluster head is selected based on the threshold calculation formula, and the node information is sent to the base station through the cluster head. Then, based on the FGB, the temporary cluster head is exchanged with the members of the cluster, so that the node with the highest energy becomes the cluster head. The cluster head sends a message to the adjacent nodes; the common node selects the cluster head to join the cluster and becomes a member node in the cluster; and the cluster is formally established. The SDWSN enters the stable stage and transmits data according to inter-cluster routing.

In DHEEC, entropy is used to describe the probability of a node predicting to become a cluster head, the information uncertainty of the node's initial energy and residual energy, the smaller the entropy, the greater the residual energy of the node. Each sensor node automatically completes its own entropy selection by obtaining the initial energy and remaining energy levels.

### 5.1 Threshold formula

Based on the assumption of the system model, the initial energy of the sensor node is randomly distributed in the interval $[\text{En}_0, (1 + a_{\max})\text{En}_0]$, so the sensor node $s_i(i = 1, 2,..., N)$ has $(1 + a_i)$ $\text{En}_0$ initial energy, where $a_i \in [0, a_{\max}]$. The initial total energy $\text{En}_{\text{total}}$ of the network is shown
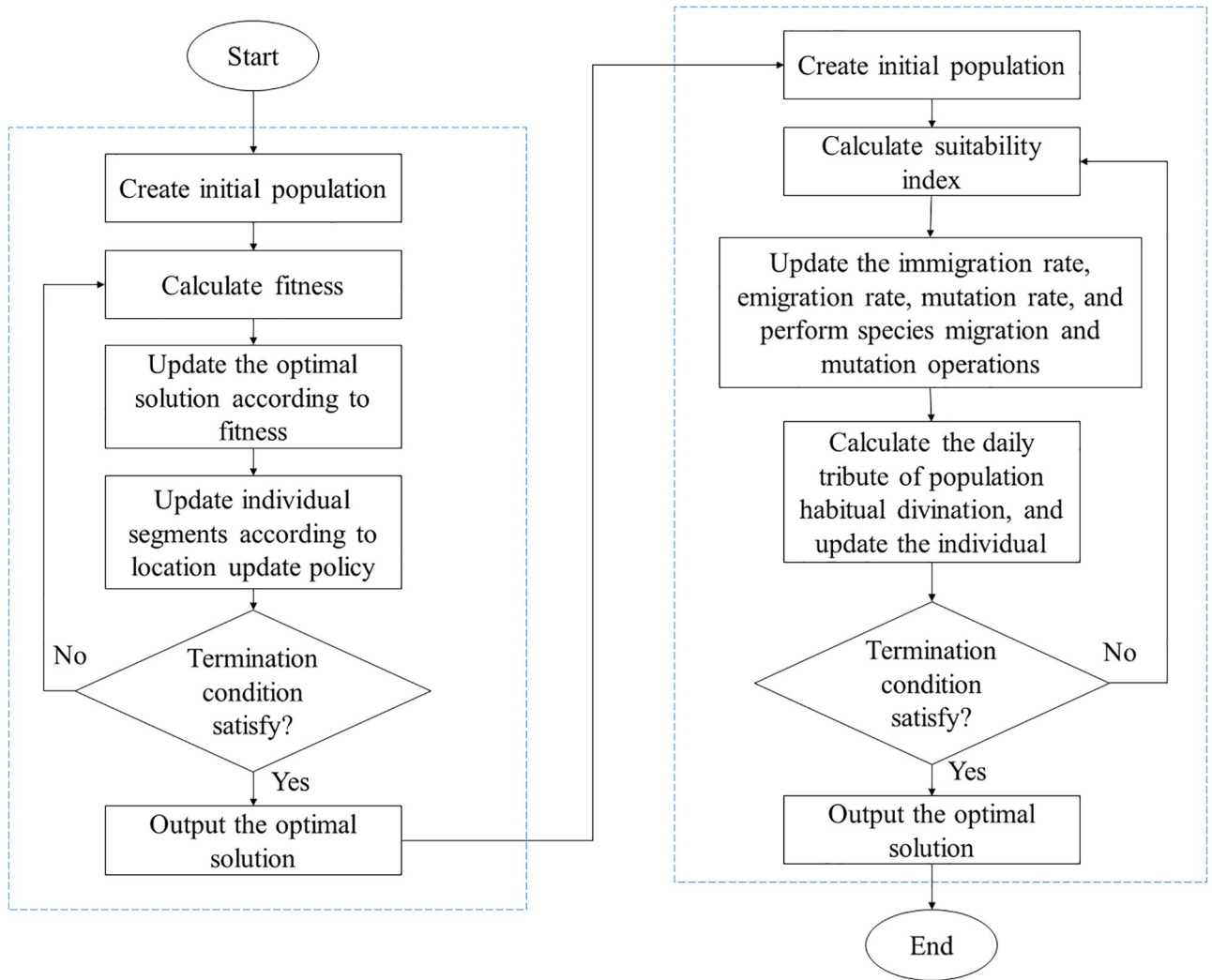
**Fig 3. Flowchart of FGB mechanism.**

in formula (24):

$$\text{En}_{\text{total}} = \sum_{i=1}^{N} (1 + a_i)\text{En}_0 = \text{En}_0\left[N + \sum_{i=1}^{N} a_i\right] = \text{En}_0(N + A) \tag{24}$$

**Definition 14**: The relative energy entropy $H(\text{En}_i)$ of the node is:

$$H(\text{En}_i) = -\sum_{i=1}^{n} k \times \text{En}_i(r)\ln \bar{E}(r) \tag{25}$$

Among them, $\text{En}_i(r)$ is the remaining energy of the sensor node $s_i$ in $k$ rounds, and $\bar{E}(r)$ represents the average energy of the r round network:

$$\bar{E}(r) = \frac{1}{N}\sum_{i=1}^{N} \text{En}_i(r) \tag{26}$$

The relative energy entropy consists of three parts: the predictable number of cluster heads, the remaining energy, and the average energy, and oscillates around in the direction of the two
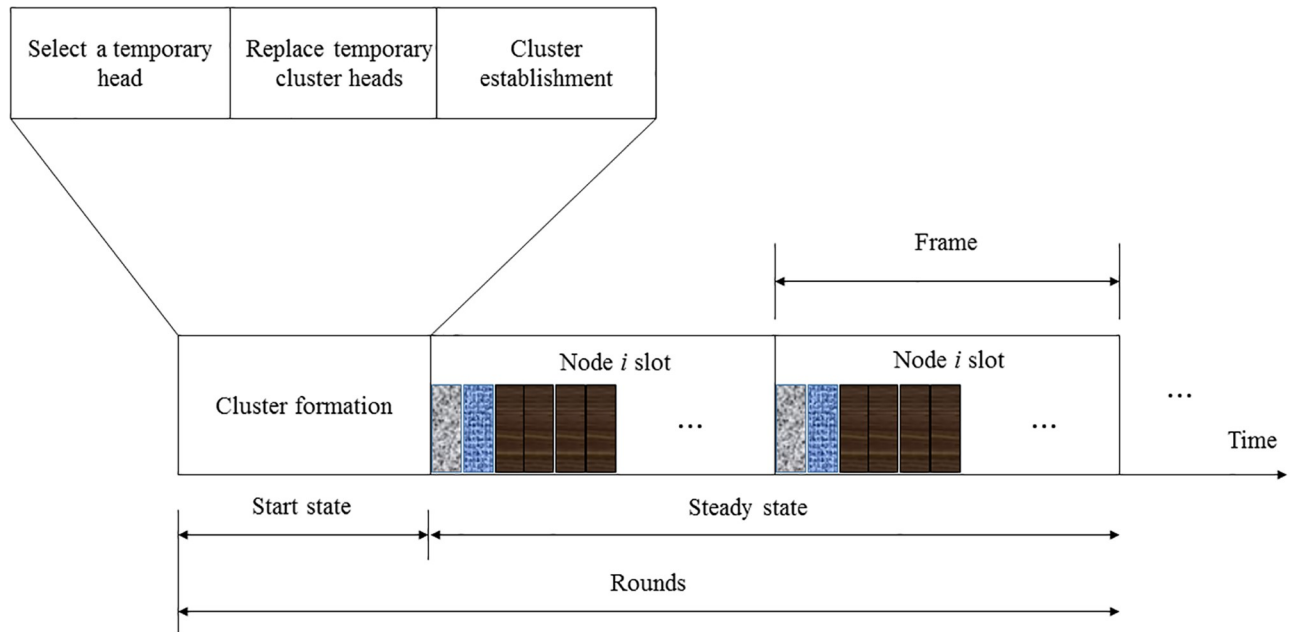
**Fig 4. Running time sequence of DHEEC.**

relative energies so that a single node can automatically manage its entropy. In each round of selecting a temporary cluster head, each node evaluates the probability of becoming a cluster head and extracts a random number between 0 and 1 as the conditional probability. If it is less than the conditional probability, the node will choose itself as a temporary cluster head.

The formula for cluster head selection threshold is:

$$T(n) = \begin{cases} H(\text{En}_i)/[1 - H(\text{En}_i)]\left[r\bmod\left[\frac{1}{H(\text{En}_i)}\right]\right], & s_i \in C \\ 0, & \text{other} \end{cases} \tag{27}$$

Among them, $C$ represents the node group that has not become the cluster head, and participates in the selection of the $r$-th round of cluster head according to the threshold $T(n)$. In the last round of algorithm iteration, if si does not become the cluster head, then $s_i \in C$. In this way, the conditional operation based on the relative energy entropy of node energy and initial energy is more suitable for energy update in SDWSN.

## 5.2 Cluster head election using FGB

FGB is used to identify the relative energy entropy in (23), using the following rules:

1. Fireflies attract one-way according to brightness, that is, node energy;

2. The attraction of fireflies is proportional to the node energy;

3. Fireflies with low energy are attracted to fireflies with high energy;

4. The firefly with the highest energy moves randomly.

In DHEEC, fireflies with low entropy, high attractiveness, and good location will attract a large number of fireflies and the attractiveness of fireflies can be predicted by the objective

**Table 2. Node data transmission method.**

| Data transfer method | Condition |
|:---:|:---:|
| Direct | $d <$ dis |
| Cluster head | $d \geq$ dis |

function, namely relative energy entropy. If the attractive forces are balanced, the fireflies will move arbitrarily.

In the algorithm, FGB is used for cluster head selection and is divided into two stages. In the first stage, the improved FA performs preliminary optimization, and the optimal solution found in this stage is $C_{\text{best}}$;

In the second stage, a more refined search of $C_{\text{best}}$ was performed using BBOGSA to find the optimal solution of is $G_{\text{best}}$. The optimal solution is the node with the largest energy in the current iteration times.

## 5.3 Inter-cluster routing

Sensor nodes send node location information, remaining energy information, and topology information to the base station. The node data transmission mode is shown in Table 2, where d is the distance between the node and the base station and 'dis' is the distance threshold, that is, the distance from the node to the cluster head. The base station calculates the network route and controls the sensor nodes. This approach reduces energy consumption and computational overhead. The flow chart of the DHEEC algorithm is shown in Fig 5.
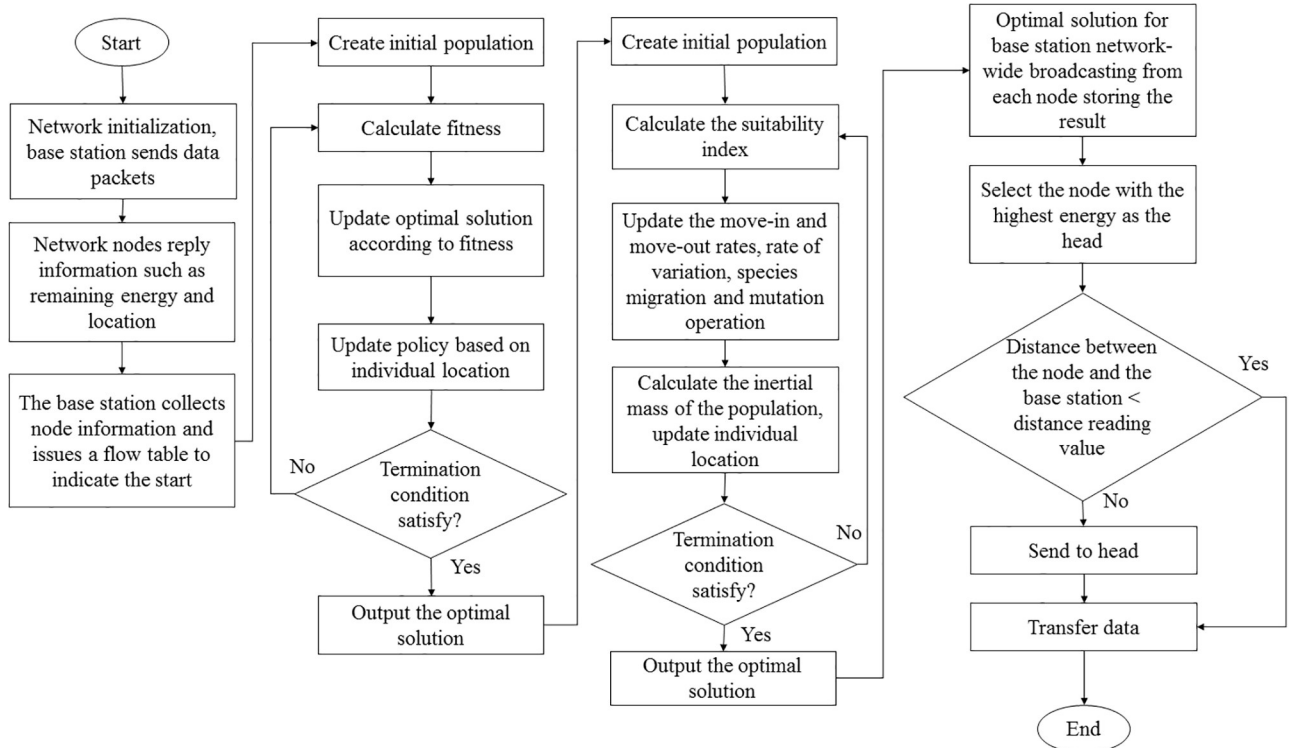


**Fig 5. Flowchart of the proposed DHEEC algorithm.**

# 6 Simulation results

The software used in this section is MATLAB 2018a. First, verify the convergence performance of FGB by comparing the performance of PSOGWO [22], HFPSO [17] and FGB on different test functions; Secondly, by comparing the network lifetime and data transmission volume of IFCEER [15], DEEC-FA [20] and DHEEC, the network performance of DHEEC is verified. PSO is a swarm-based meta-heuristic optimization technique that moves particles in a designated exploration area in an attempt to find the optimum solution to a given issue. Conversely, Grey Wolf Optimization (GWO) is a meta-heuristic optimization approach that draws inspiration from wolves. This article aims to enhance the GWO's advancement by hybridizing it with a PSO approach. A metaheuristic optimization approach called Hybrid Firefly and Particle Swarm Optimization (HFPSO) combines the best aspects of particle swarm and firefly optimization. By examining the prior global best fitness values, HFPSO attempts to accurately establish when the local search process began.

## 6.1 Standard function test experiment

The standard functions used for testing are shown in Table 3, and $f_1$ is a unimodal function to examine the convergence speed and precision of the optimization algorithm. $f_2, f_3$, and $f_{14}$ are multimodal functions, and examine the global search ability of the optimization algorithm. $f_5$ is a discontinuous step function to examine the effectiveness of the optimization algorithm. The problem dimension is 30, and their optimal values are all 0.

**6.1.1 Parameter settings.**   Some parameters of the test standard function experiment are set as follows: $N = 30$, $\alpha = 0.2$, $\beta^0 = 1$, $D = 3$, $E = 1$, $I = 1$.
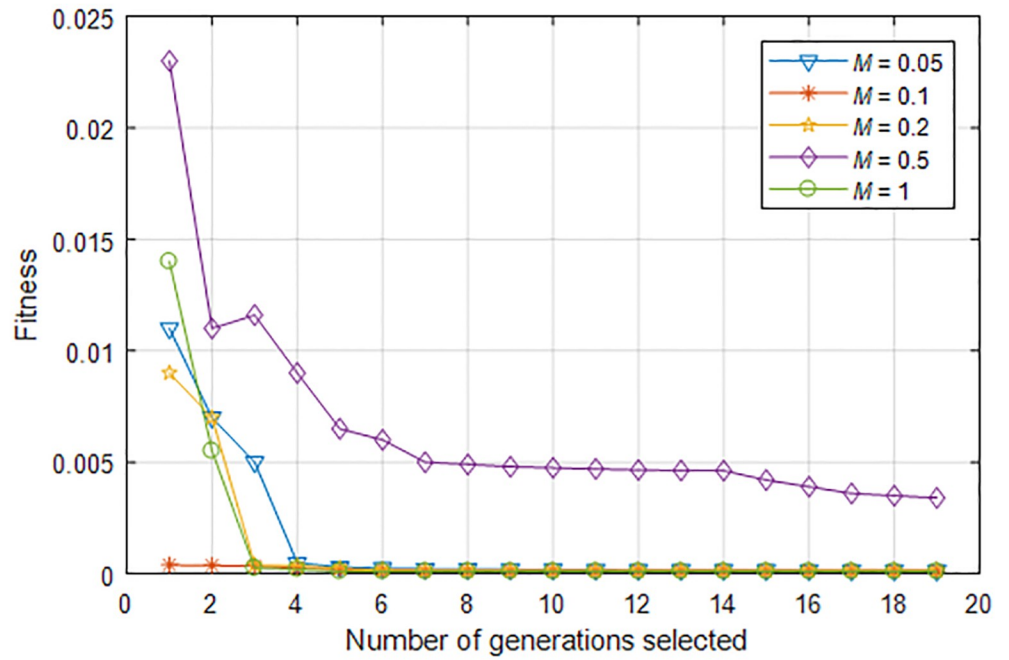
The fitness changes of FGB under different maximum mutation rates and population diversity indices are shown in Fig 6. Fig 6(a) shows the fitness change of FGB when the maximum number of iterations is 20 rounds and the maximum mutation rate $M$ is 0.05, 0.1, 0.2, 0.5 and 1 respectively. The mutation rate increases the diversity of individuals in the population, and a moderate mutation rate promotes the algorithm to quickly converge to the optimal value. Here, the maximum mutation rate is 0.1. Fig 6(b) shows the fitness changes of FGB when the population diversity index β is 0.1, 0.2 and 0.5 respectively. When the population diversity is

**Table 3. Standard functions to test.**

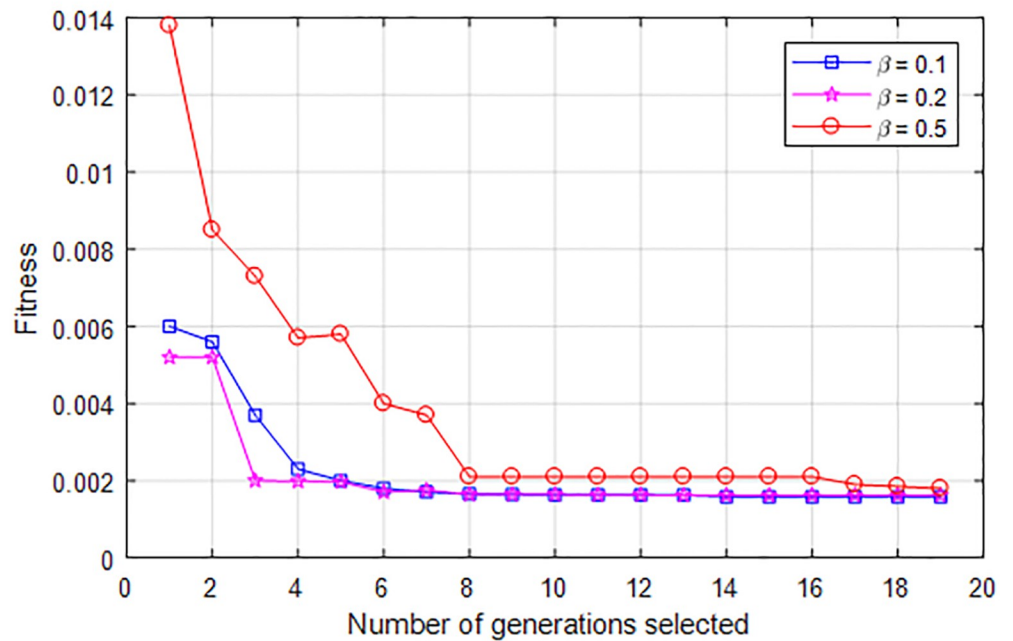| Function name | Expression | Range |
|:---:|:---:|:---:|
| Sphere | $f_1(x) = \sum_{i=1}^{n} x_i^2$ | [-100, 100] |
| Schwefel | $f_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$ | [-10, 10] |
| Rastrigin | $f_3(x) = \sum_{i=1}^{n} \left[ x_i^2 - 10\cos(2\pi x_i) + 10 \right]$ | [-5.12, 5.12] |
| | $f_4(x)$ $$= -20\, e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}}$$ $$-e^{\left[\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right]+20+e}$$ | [-32, 32] |
| Step | $f_5(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$ | [-100, 100] |

(a)



(b)



**Fig 6. Changes in the fitness of FGB under different maximum mutation rates and population diversity indices.** (a) Variation with $M$; (b) Variation with $\beta$.

**Table 4. Comparison of results of algorithms under five standard functions.**

| Function | FGB | | PSOGWO | | HFPSO | |
|---|---|---|---|---|---|---|
| | Optimal value | Running time (s) | Optimal value | Running time (s) | Optimal value | Running time (s) |
| $f_1$ | $9.0824 \times 10^{-17}$ | 1.94 | $1.4335 \times 10^{-26}$ | 2.21 | $4.7795 \times 10^{-9}$ | 2.39 |
| $f_2$ | $3.715 \times 10^{-8}$ | 1.97 | $9.465 \times 10^{-9}$ | 2.05 | $5.6329 \times 10^{-5}$ | 2.30 |
| $f_3$ | 20.8941 | 1.91 | $23.6247 \times 10^{-7}$ | 1.98 | $36.813 \times 10^{-5}$ | 2.11 |
| $f_4$ | $7.8769 \times 10^{-9}$ | 2.27 | 374.867 | 1.85 | $2.8522 \times 10^{-5}$ | 1.86 |
| $f_5$ | 0 | 2.08 | | 1.19 | 0 | 1.47 |

moderate, the species probability is higher and the algorithm is more stable. At this time, the population diversity index is taken as 0.2.

**6.1.2 Results.** The results of the three algorithms under the five standard functions are compared in Table 4. It can be seen from the results that FGB is slightly worse than PSOGWO on $f_1$, but the convergence speed is faster than PSOGWO. The performance of FGB on $f_2$ shows that its search solution range is large. FGB performs well on $f_3$, $f_4$ and $f_5$, which shows its practicability, convergence speed and search ability are stronger than the other two algorithms.

## 6.2 Network simulation experiment

**6.2.1 Simulation parameter settings.** Some parameters of the network simulation experiment are shown in Table 5.

**6.2.2 Network life cycle.** The energy of network nodes decreases as the number of iterations of the algorithm increases, and is exhausted after a certain number of iterations, resulting in the death of the node. The number of dead nodes in the network represents the life cycle of the network.

The comparison of the network node life cycle of IFCEER, DEEC-FA and DHEEC is shown in Fig 7, which shows the impact of the three algorithms of IFCEER, DEEC-FA and DHEEC on the network life cycle. Fig 7(a) shows the changes in the number of dead nodes in the three algorithms of IFCEER, DEEC-FA and DHEEC when the maximum number of iterations is 5000 rounds. It can be seen that the number of death rounds of all nodes of the three

**Table 5. Network simulation parameters.**

| Parameter | Value |
|---|---|
| Network area | 100 m×100 m, 250 m × 250 m |
| Number of sensor nodes | 100 ~ 300 |
| Base station location | (50, 50) |
| Transmission packet | 4000 bits |
| Packet rate | 4 packets/s |
| Number of nodes per cluster | 6 |
| Max battery level of node | 3 J |
| $E_0$ | 0.5 J |
| $E_{elec}$ | 50 nJ/bit |
| $\varepsilon_{fs}$ | 10 nJ/bit.m$^2$ |
| $\varepsilon_{mp}$ | 0.0013 pJ/bit.m$^4$ |
| $E_{DA}$ | 5 nJ/bit.signal |
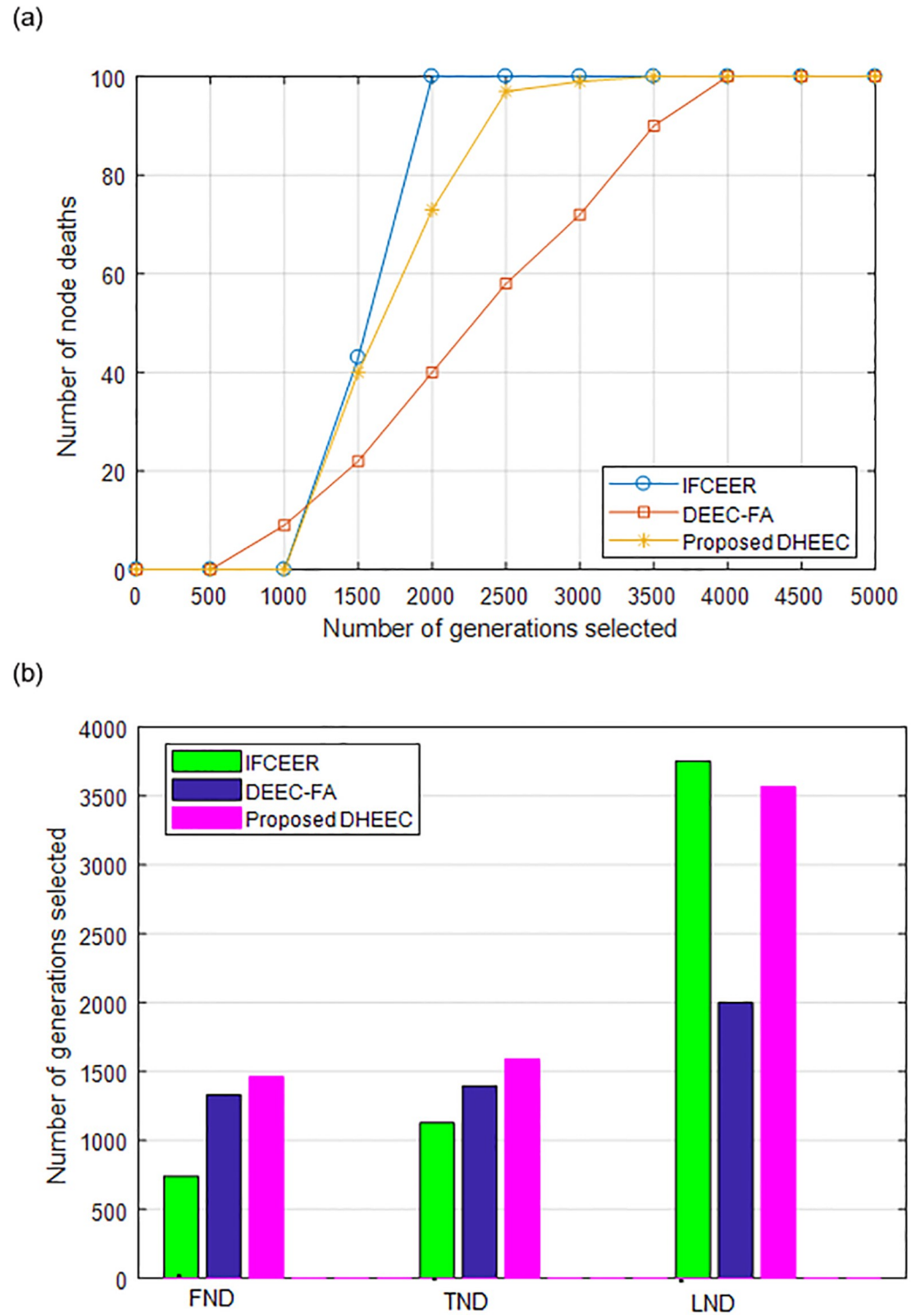
**Fig 7. Comparison of network node life cycle of algorithms.** (a) Number of node deaths. (b) Node death time.

https://doi.org/10.1371/journal.pone.0301078.g007

algorithms of IFCEER, DEEC-FA and DHEEC is 2000, 3754 and 3568 respectively. The first node of the network dies, and the information collected by the base station is no longer comprehensive. With the operation of the algorithm, the number of dead nodes increases, and a small number of nodes that survive in the later stage have low remaining energy and lose communication capabilities. Therefore, in order to measure the life cycle of the network more comprehensively, Fig 7(b) compares the number of rounds of the first node death (first node died, FND), 10% node died rounds (10% node died, TND) and all nodes died rounds (last node died, LND). It can be seen that IFCEER only allows high-energy nodes to participate in the election of cluster heads, resulting in very close FND and LND. The FND of DEEC-FA is much smaller than the other two algorithms, and the network performance drops seriously in the early stage of the algorithm. Considering the balance of the overall energy consumption of the network, and defining TND as the network life cycle, the TND of DHEEC is about 41.05% higher than IFCEER, and about 13.89% higher than that of DEEC-FA.

**6.2.3 Network data transfer volume.** During the operation of the algorithm, the surviving member nodes transmit the collected data to the cluster head, and then the cluster head transmits it to the base station. After the transmission of all nodes is completed, the base station counts the number of data packets received in this round. Therefore, the total number of data packets received by the base station is used to evaluate the energy utilization rate. The more data packets received by the base station, the more balanced the energy distribution will be.

The total number of data packets received by the base station is shown in Fig 8, which shows the change of the total number of data packets received by the base station after 5,000 iterations of the three algorithms of IFCEER, DEEC-FA and DHEEC. It can be concluded that the energy utilization rate of DHEEC is about 31.58% higher than that of IFCEER and about 31.06% higher than that of DEEC-FA.
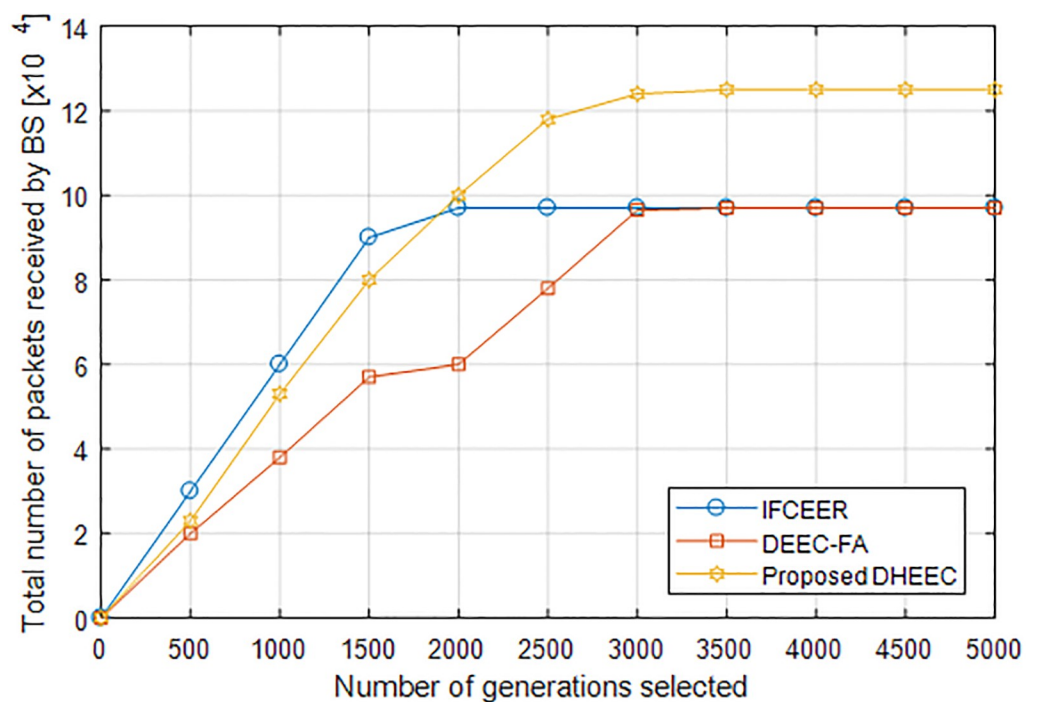


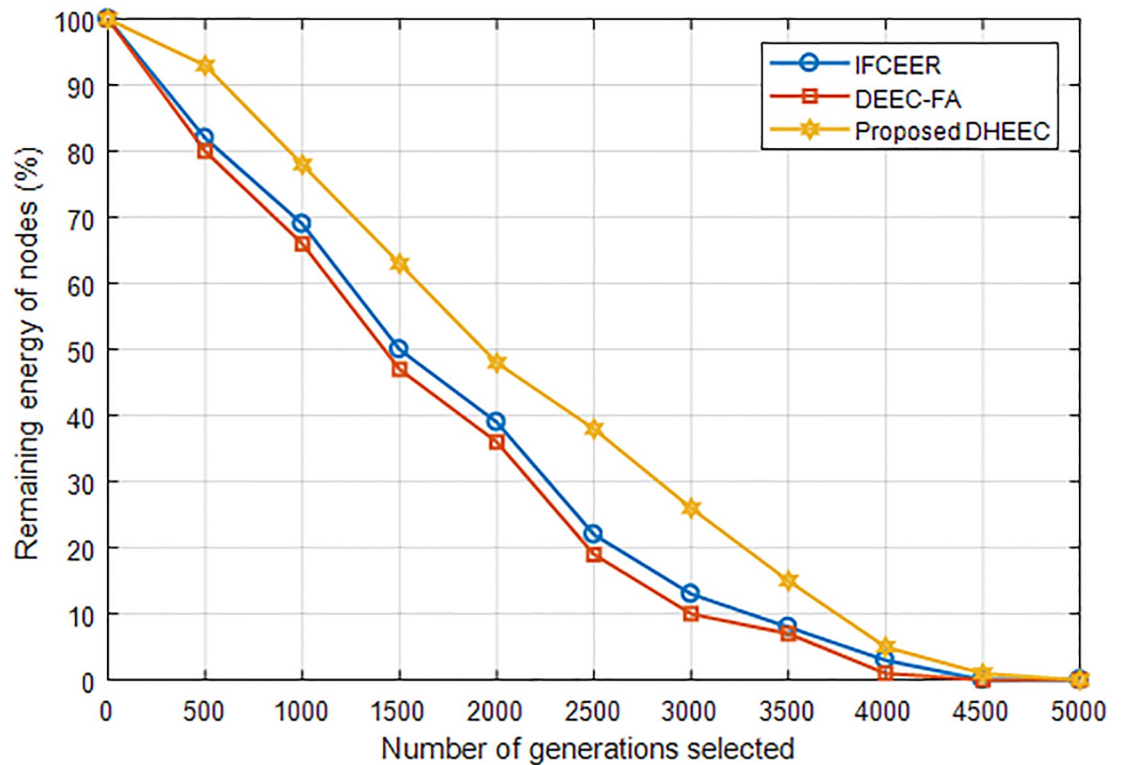**Fig 8. The total number of packets received by the base station.**

**Fig 9. Comparison of average remaining energy of nodes of the algorithms.**

Fig 9 compared the average remaining energy of nodes of the algorithms. As can be seen that the average remaining energy of the proposed algorithm is higher than the existing algorithms which makes it superior and has longer lifetime.

### 6.3 Statistical evaluation

Fig 10 compares the control overhead of algorithms under different topology size. As can be seen from the results, the control overhead of the algorithms increases with increasing topology size. However, the overhead of the proposed method is lower than existing algorithms which validates its effectiveness.

Fig 11 compares the end-to-end delay of the algorithms under increasing number of nodes. It can be seen from Fig 11, the delay increases with increasing number of nodes. However, the proposed method has lower delay than existing algorithms, which further validates its superior performance.

### 6.4 Scalability and performance in larger networks

Fig 12 evaluates the energy consumption of the network with dimension of 250 m×250 m and the number of nodes is 300. As can be seen from Fig 12, the energy consumption of the algorithms increases with increasing number of nodes. However, the proposed method consumes less energy as compared with existing algorithms which makes is energy-efficient.
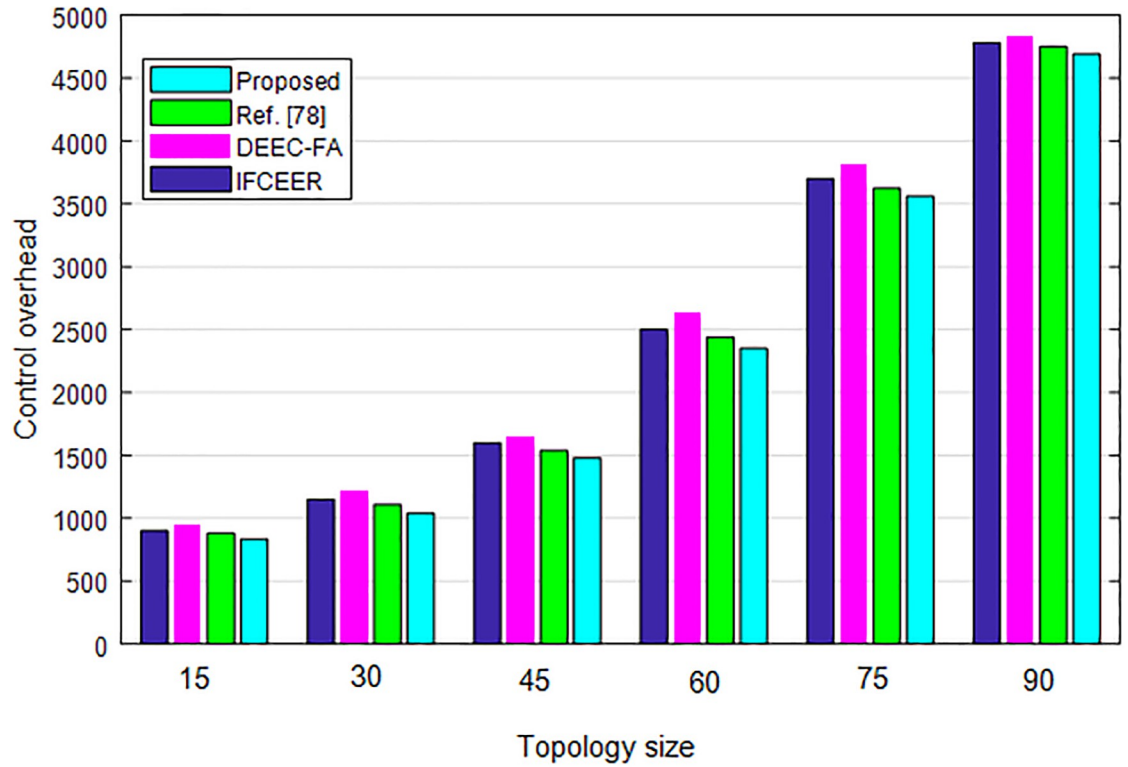
**Fig 10. Control overhead comparison of algorithms under different topology size.**
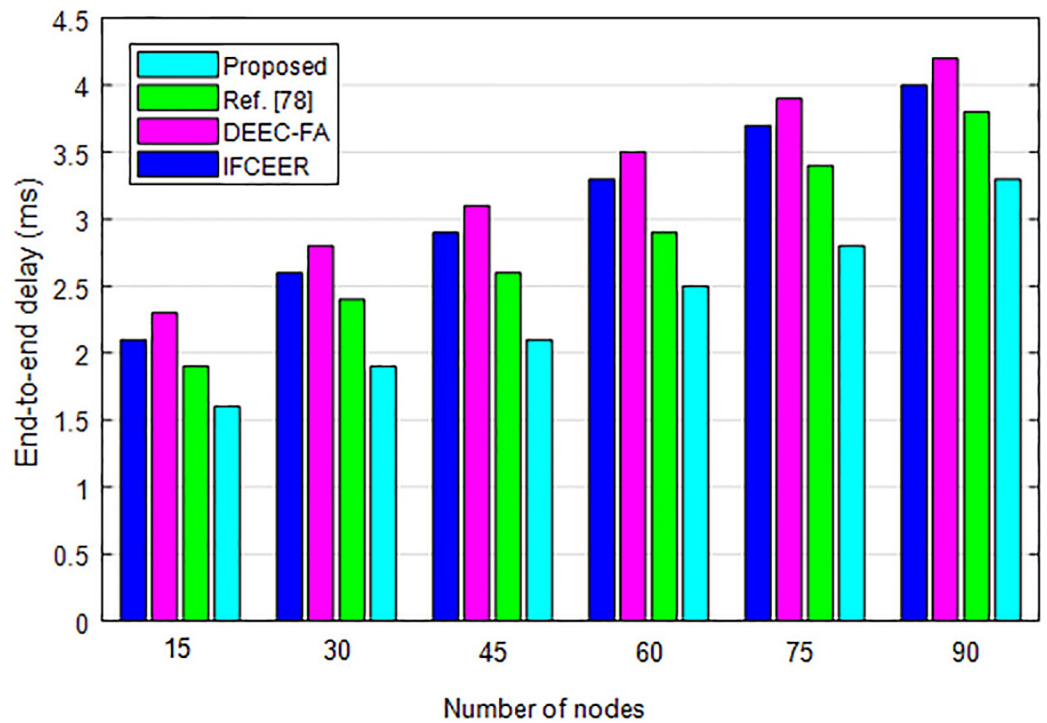
https://doi.org/10.1371/journal.pone.0301078.g010



**Fig 11. End-to-end delay comparison of algorithms under different number of nodes.**
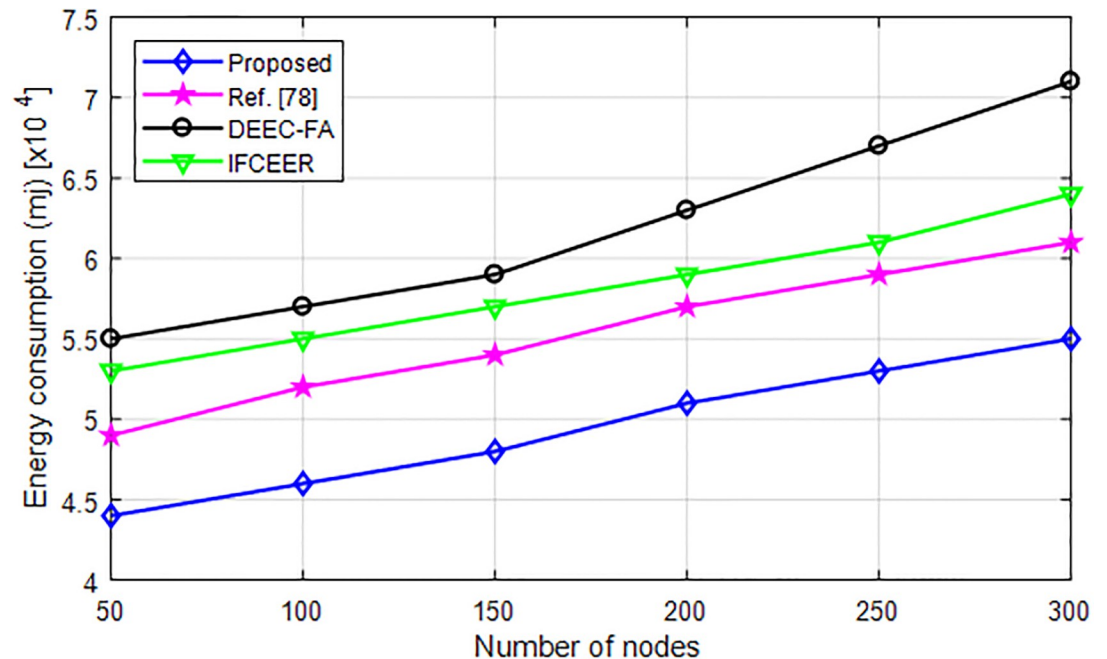
https://doi.org/10.1371/journal.pone.0301078.g011

**Fig 12. Scalability evaluation under large network.**

### 6.5 Energy efficiency analysis under different network scenario

Fig 13 evaluates the energy efficiency of the algorithms under different network scenario. As can be seen from Fig 13, the energy efficiency of each component reduces with increasing data rate. However, its within the required range of operation.

### 6.6 Impact on running time and computational resources

Fig 14 compares the running time of network of algorithms under increasing number of iterations. As can be seen from Fig 14, the network lifetime of the algorithms decreases with increasing number of iterations. However, the running time of the proposed algorithm is longer than existing methods which makes is reliable and require less computational resources.

### 6.7 Robustness to network dynamics

Fig 15 compares the robustness of the algorithms when the mobile nodes move at different speed. As can be seen from Fig 15a, when the number of nodes increases, the communication delay increases. The communication delay of the proposed algorithm is lower than existing methods. From Fig 15b, the mobile node speed is 5.5 m/s and as the number of nodes increases, the communication delay at this speed is larger than at speed of 1.8 m/s. In both scenarios, the proposed method has lower communication delay which proved its effectiveness. This proves that, the proposed method is robust to dynamic network environment and with varying data traffic.

### 6.8 Convergence of algorithm

Fig 16 illustrates the convergence of the proposed FGB algorithm under increasing number of iterations. As can be seen from Fig 16, the algorithm converges to local minimum that indicates its effectiveness.
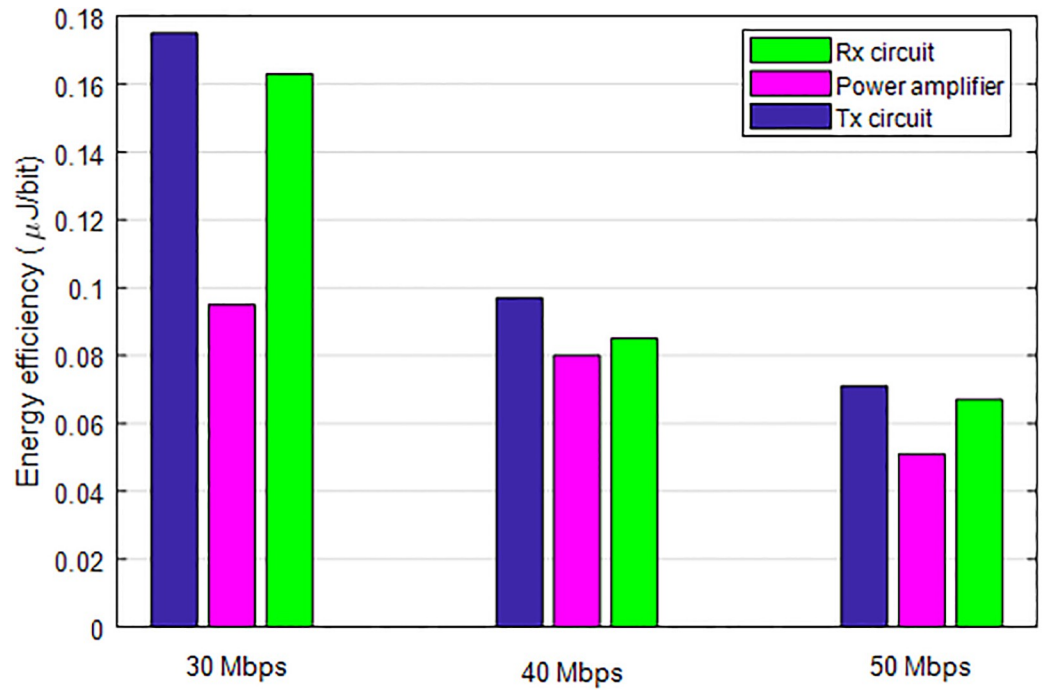
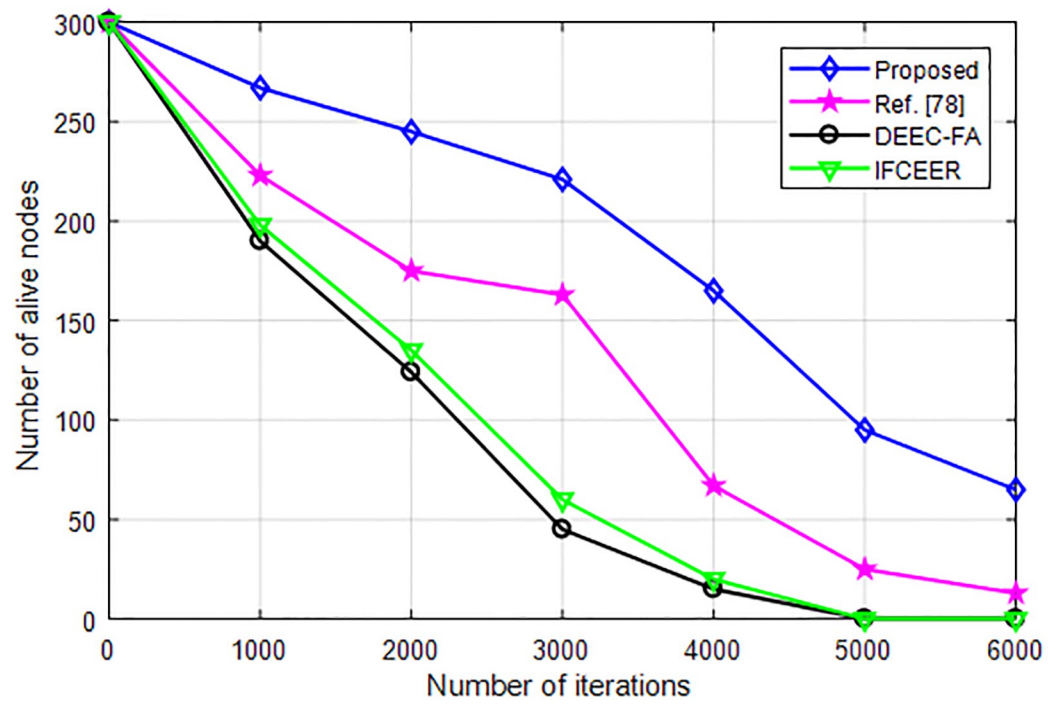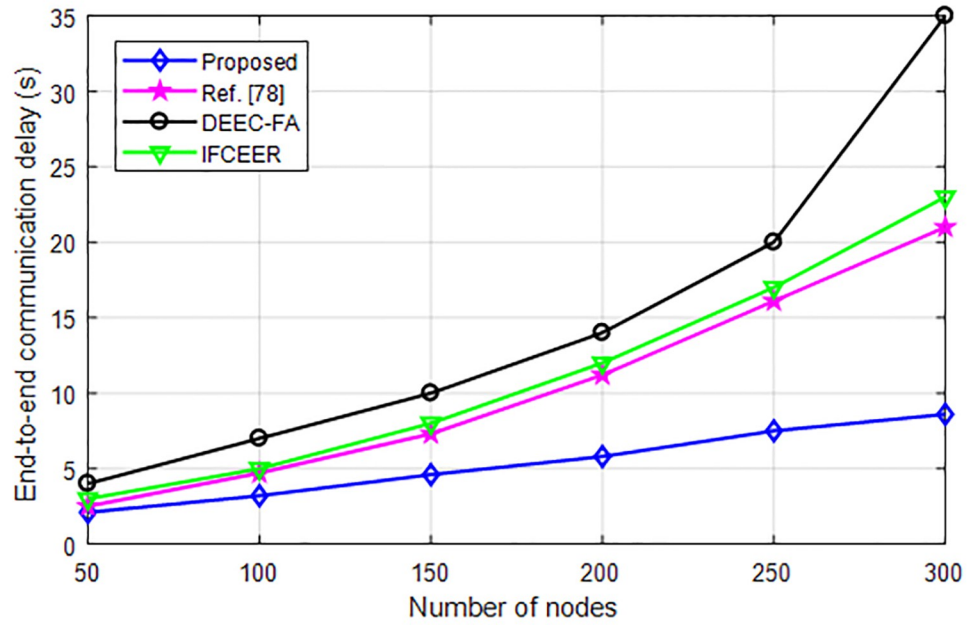**Fig 13. Energy efficiency analysis of each component.**

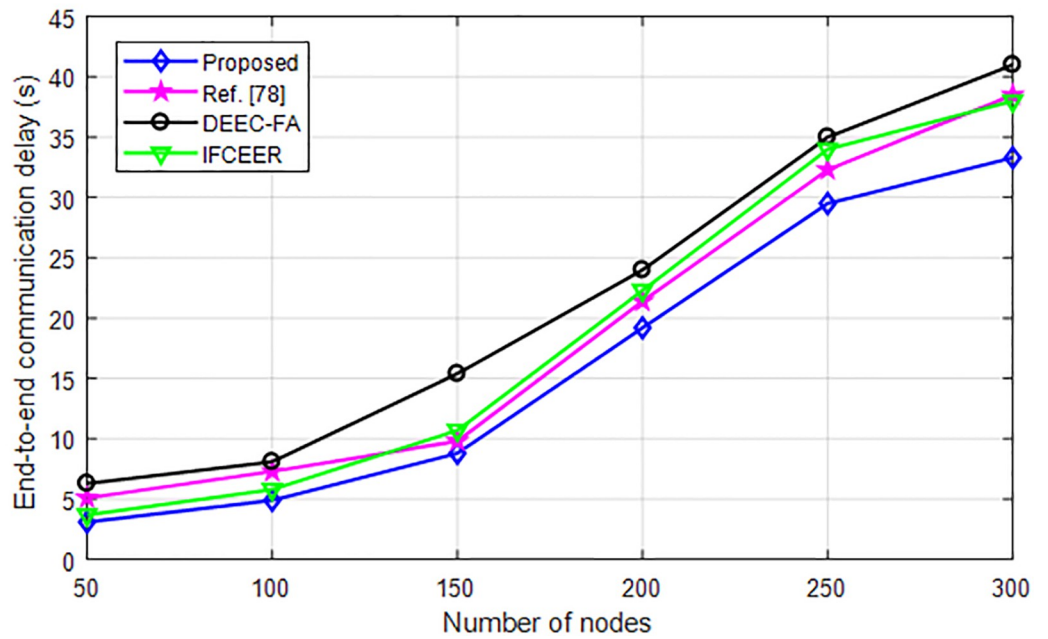**Fig 14. Network running time comparison of algorithms.**

(a)



(b)



**Fig 15. Communication delay evaluation at different node mobility.** (a) 1.8 m/s. (b) 5.5 m/s.
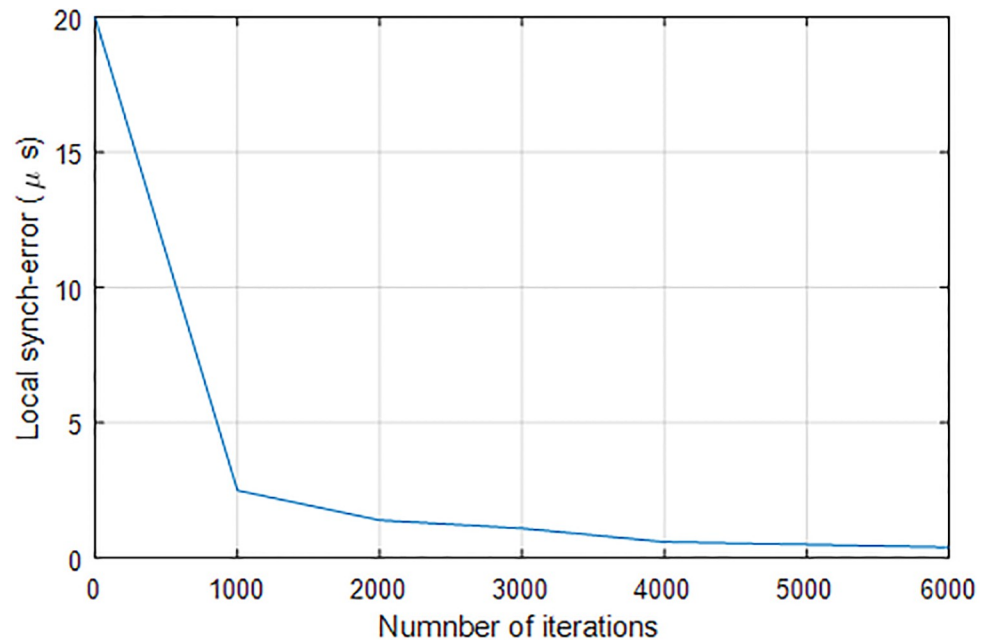
**Fig 16. Convergence of algorithm.**

https://doi.org/10.1371/journal.pone.0301078.g016

## 6.9 Complexity analysis

Fig 17 compares the computational complexity in terms of computing time of the proposed and existing algorithms with increasing number of iterations. As can be seen from Fig 17, the complexity of the proposed algorithm is lower than existing algorithms which makes is computationally efficient and easy to implement.

## 7 Conclusion and future work

The clustering routing technique for SDWSN is examined in this work. A hybrid optimization algorithm that combines the global optimization capabilities of FA with the local optimization capabilities of BBOGSA is developed with the goal of enhancing the performance of the optimization process for locating an approximate global solution. To hasten the rotation of cluster heads, a distributed high-efficiency entropy energy-saving clustering routing method is developed. The hybrid optimization method is compared with comparable algorithms in terms of convergence accuracy, running time, and stability through experimental simulation and analysis to demonstrate its optimization performance. To demonstrate how the distributed high-efficiency entropy energy-saving clustering routing algorithm may increase network longevity and decrease data transmission energy consumption, it is compared with comparable methods. The limitations of the proposed study are as follows that will be considered as next research. To increase the security of SDWSN, the following phase of development will look at creating a DDoS assault detection and defense system. It will employ an architecture that carries out multi-level DDoS mitigation in the data forwarding layer, first-level DDoS detection in the edge layer, second-level DDoS detection in the control layer, and clustering in the device layer. This approach will be more scalable, energy-efficient and improved.
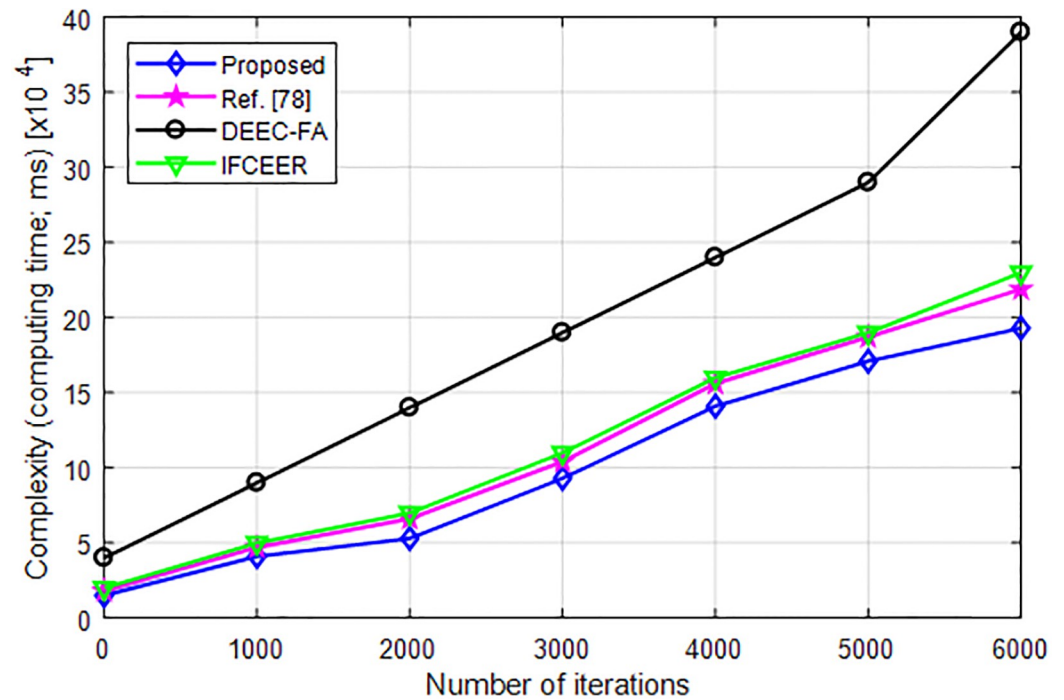
**Fig 17. Computational complexity evaluation.**

https://doi.org/10.1371/journal.pone.0301078.g017

## Author Contributions

**Conceptualization:** Kemal Gökhan Nalbant, Asma Hassan Alshehri, Maha Hatira.

**Data curation:** Kemal Gökhan Nalbant, Suliman A. Alsuhibany, Asma Hassan Alshehri, Maha Hatira.

**Formal analysis:** Asma Hassan Alshehri, Bong Jun Choi.

**Funding acquisition:** Suliman A. Alsuhibany.

**Investigation:** Suliman A. Alsuhibany, Bong Jun Choi.

**Methodology:** Kemal Gökhan Nalbant, Maha Hatira, Bong Jun Choi.

**Project administration:** Suliman A. Alsuhibany.

**Resources:** Asma Hassan Alshehri.

**Software:** Maha Hatira, Bong Jun Choi.

**Validation:** Maha Hatira.

**Visualization:** Kemal Gökhan Nalbant, Bong Jun Choi.

**Writing – original draft:** Kemal Gökhan Nalbant, Asma Hassan Alshehri, Maha Hatira, Bong Jun Choi.

**Writing – review & editing:** Kemal Gökhan Nalbant, Suliman A. Alsuhibany, Asma Hassan Alshehri.

# References

1. Tareq A., Sundarajan E., Mohd M., and Sani N. S. 2020. Online clustering of evolving data streams using a density grid-based method. *IEEE Access* 8: 166472–166490.

2. Guo Y., and Lei X. 2022. Distributed machine learning for multiuser mobile edge computing systems. *IEEE Journal of Selected Topics in Signal Processing* 16(3): 460–473.

3. Zheng S., Shen C., and Chen X. 2020. Design and analysis of uplink and downlink communications for federated learning. *IEEE Journal of Selected Areas in Communications* 39(7): 2150–2167.

4. Xia J., and Fan L. 2019. Secure cache-aided multi-relay networks in the presence of multiple eavesdroppers. *IEEE Transactions on Communications*, 67(11): 7672–7685.

5. Tang S., Chen L., He K., and Xia J. 2023. Computational intelligence and deep learning for next generation edge-enabled industrial IoT. *IEEE Transactions on Networks Science and Engineering*, 99: 1–15.

6. Wang J., Han H., Li H., He S., and Sharma P. 2022. Multiple strategies differential privacy on spare tensor factorization for network traffic analysis in 5G. *IEEE Transactions on Industrial Informatics*, 18(3): 1939–1948.

7. Cao D., Zheng B., Ji B., Lei Z., and Feng C. 2020. A robust distance-based relay selection for message dissemination in vehicular network. *Wireless Network*, 26(3): 1755–1771.

8. Zhou W., Fan L., Zhou F., Li F., and Lei X. 2023. Priority-aware resource scheduling for UAV-mounted mobile edge computing networks. *IEEE Transactions on Vehicular Technology* 72(7): 9682–9687.

9. Chen L., and Lei X. 2023. Relay-assisted federated edge learning: Performance analysis and system optimization. *IEEE Transactions on Communications*, 71(6): 3387–3401.

10. Zhou X., and Zhou F. 2023. Profit maximization for cache-enabled vehicular mobile edge computing networks. *IEEE Transactions Vehicular Technology* 72(10): 13793–13798.

11. Zheng X., Zhu F., and Xia J. 2023. Intelligent computing for WPT-MEC-aided multi-source data stream. *EURASIP Journal on Advances in Signal Processing* 23(1): 1–17.

12. Ling J., Xia J., and Zhu F. 2023. DQN-based resource allocation for NOMA-MEC-aided multi-source data stream. *EURASIP Journal on Advances in Signal Processing* 1(44): 1–14.

13. Wu Y., and Gao C. 2023. Intelligent resource allocation scheme for cloud-edge-end framework aided multi-source data stream. *EURASIP Journal on Advances in Signal Processing* 2023(1): 1–16.

14. He L., and Tang X. 2023. Learning-based MIMO detection with dynamic spatial modulation. *IEEE Transactions on Cognitive Communications and Networking* 8(1): 1–13.

15. Xu G., Zhang Q., Song Z., & Ai B. (2023). Relay-Assisted Deep Space Optical Communication System Over Coronal Fading Channels. IEEE Transactions on Aerospace and Electronic Systems, 59(6), 8297–8312. https://doi.org/10.1109/TAES.2023.3301463

16. Yin Y., Guo Y., Su Q., & Wang Z. (2022). Task Allocation of Multiple Unmanned Aerial Vehicles Based on Deep Transfer Reinforcement Learning. Drones, 6(8), 215. https://doi.org/10.3390/drones6080215

17. Fang Z., Wang J., Liang J., Yan Y., Pi D., Zhang H., et al. (2024). Authority Allocation Strategy for Shared Steering Control Considering Human-Machine Mutual Trust Level. IEEE Transactions on Intelligent Vehicles, 9(1), 2002–2015. https://doi.org/10.1109/TIV.2023.3300152

18. Li A., Masouros C., Swindlehurst A. L., & Yu W. (2021). 1-Bit Massive MIMO Transmission: Embracing Interference with Symbol-Level Precoding. IEEE Communications Magazine, 59(5), 121–127. https://doi.org/10.1109/MCOM.001.2000601

19. Qu J., Mao B., Li Z., Xu Y., Zhou K., Cao X., et al. (2023). Recent Progress in Advanced Tactile Sensing Technologies for Soft Grippers. Advanced Functional Materials, 33(41), 2306249. https://doi.org/10.1002/adfm.202306249

20. Yin F., Lin Z., Kong Q., Xu Y., Li D., Theodoridis S., et al. (2020). FedLoc: Federated Learning Framework for Data-Driven Cooperative Localization and Location Data Processing. IEEE Open Journal of Signal Processing, 1, 187–215. https://doi.org/10.1109/OJSP.2020.3036276

21. Ghosh R., Mohanty S., Pattnaik P., "A performance assessment of power-efficient variants of distributed energy-efficient clustering protocols in WSN," *International Journal of Interactive Communication Systems and Technologies*, vol. 10, no. 2, pp. 1–14, 2020.

22. Prasad V., Periyasamy S., "Energy optimization-based clustering protocols in wireless sensor networks and internet of things-survey," *International Journal of Distributed Sensor Networks*, vol. 23, pp. 1–17, 2023.

23. Duan L., Yang S., Zhang D., "The optimization of feature selection based on chaos clustering strategy and niche particle swarm optimization," *Mathematical Problems in Engineering*, vol. 20, ID: 3138659, pp. 1–8, 2020.

**24.** Rayenizadeh M., Rafsanjani M., Saeid A., "Cluster head selection using hesitant fuzzy and firefly algorithm in wireless sensor networks," *Evolving Systems*, vol. 13, no. 1, pp. 65–84, 2022.

**25.** Singh S., Malik A., Kumar R., "Energy efficient heterogeneous DEEC protocol for enhancing lifetime in WSNs," *Engineering Science and Technology: An International Journal*, vol. 20, no. 1, pp. 345–353, 2017.

**26.** Wang H., Wang W., Zhou X., Wang Y., "Artificial bee colony algorithm based on knowledge fusion," *Complex & Intelligent Systems*, vol. 7, pp. 1139–1152, 2021.

**27.** Bhola J., Soni S., Cheema K., "Genetic algorithm based optimized leach protocol for energy efficient wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 3, pp. 1281–1288, 2020.

**28.** Poluru R., Reddy P., Basha S., "Enhanced adaptive distributed energy-efficient clustering (EADEEC) for wireless sensor networks," *Journal of Recent Advances in Computer Science and Communications*, vol. 13, no. 2, pp. 190–194, 2020.

**29.** Lu S., Wu M., Zhao M., "Particle swarm optimization and artificial bee colony algorithm for clustering and mobile based software-defined wireless sensor networks," *Wireless Networks*, vol. 28, no. 4, pp. 1671–1688, 2022.

**30.** Amutha J., Sharma S., Sharma S., "An energy efficient cluster based hybrid optimization algorithm with static sink and mobile sink node for wireless sensor networks," *Expert Systems with Applications*, vol. 203, no. 1, pp. 334–345, 2022.

**31.** Kumar P., Valarmathi K., "Fuzzy based hybrid BAT and firefly algorithm for optimal path selection and security in wireless sensor network," *Journal of Automatika*, vol. 64, no. 2, pp. 199–210, 2022.

**32.** Zhang L., Tang Y., Hua C., "A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques," *Applied Soft Computing*, vol. 15, no. 28, pp. 138–149, 2015.

**33.** Aydilek I., "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems," *Applied Soft Computing*, vol. 18, no. 66, pp. 232–249, 2018.

**34.** Pitchaimanickam B., Murugaboopathi G., "A hybrud firefly algorithm with particle swarm optimization for energy efficient optimal cluster head selection in wireless sensor networks," *Neural Computing and Applications*, vol. 32, no. 12, pp. 7709–7723, 2020.

**35.** Basith K., Shankar T., "Hybrid state analysis with improved firefly optimized linear congestion models of WSN for DDoS & CRA attacks," *PeerJ Computer Science*, vol. 22, no. 8, pp. 845–858, 2022.

**36.** Kumari C., Yadav H., Chikara R., "Hybrid model using firelfy and BBO for feature selection in software product line," *Journal of Recent Advances in Computer Science and Communications*, vol. 14, no. 9, pp. 2754–2760, 2021.

**37.** Shrivastava A., Sharma A., Pandit M., "Hybrid protection scheme based optimal overcurrent relay coordination strategy for RE integrated power distribution grid," *Energies*, vol. 14, no. 21, pp. 1–18, 2021.

**38.** Aydin M., Karabekir B., Zaim A., "Energy efficient clustering-based mobile routing algorithm on WSNs," *IEEE Access*, vol. 9, pp. 89593–89601, 2021.

**39.** Yin F., Fritsche C., Jin D., Gustafsson F., & Zoubir A. M. (2015). Cooperative Localization in WSNs Using Gaussian Mixture Modeling: Distributed ECM Algorithms. IEEE Transactions on Signal Processing, 63(6), 1448–1463. https://doi.org/10.1109/TSP.2015.2394300

**40.** Ma B., Liu Z., Dang Q., Zhao W., Wang J., Cheng Y., et al (2023). Deep Reinforcement Learning of UAV Tracking Control Under Wind Disturbances Environments. IEEE Transactions on Instrumentation and Measurement, 72. https://doi.org/10.1109/TIM.2023.3265741

**41.** Zhang J., Ren J., Cui Y., Fu D., & Cong J. (2024). Multi-USV Task Planning Method Based on Improved Deep Reinforcement Learning. IEEE Internet of Things Journal. https://doi.org/10.1109/JIOT.2024.3363044

**42.** Dai M., Luo L., Ren J., Yu H., & Sun G. (2022). PSACCF: Prioritized Online Slice Admission Control Considering Fairness in 5G/B5G Networks. IEEE Transactions on Network Science and Engineering, 9(6), 4101–4114. https://doi.org/10.1109/TNSE.2022.3195862

**43.** Sun G., Xu Z., Yu H., & Chang V. (2021). Dynamic Network Function Provisioning to Enable Network in Box for Industrial Applications. IEEE Transactions on Industrial Informatics, 17(10), 7155–7164. https://doi.org/10.1109/TII.2020.3042872

**44.** Wang Q., Li P., Rocca P., Li R., Tan G., Hu N., et al. (2023). Interval-Based Tolerance Analysis Method for Petal Reflector Antenna with Random Surface and Deployment Errors. IEEE Transactions on Antennas and Propagation, 71(11), 8556–8569. https://doi.org/10.1109/TAP.2023.3314097

**45.** Sun G., Sheng L., Luo L., & Yu H. (2022). Game Theoretic Approach for Multipriority Data Transmission in 5G Vehicular Networks. IEEE Transactions on Intelligent Transportation Systems, 23(12), 24672–24685. https://doi.org/10.1109/TITS.2022.3198046

46. Zhao S., Liang W., Wang K., Ren L., Qian Z., Chen G., et al. (2023). A Multiaxial Bionic Ankle Based on Series Elastic Actuation with a Parallel Spring. IEEE Transactions on Industrial Electronics. https://doi.org/10.1109/TIE.2023.3310041

47. Cai L., Yan S., Ouyang C., Zhang T., Zhu J., Chen L., et al. (2023). Muscle synergies in joystick manipulation. Frontiers in Physiology, 14. https://doi.org/10.3389/fphys.2023.1282295 PMID: 37900948

48. Mao Y., Sun R., Wang J., Cheng Q., Kiong L. C., Ochieng W. Y. (2022). New time-differenced carrier phase approach to GNSS/INS integration. GPS Solutions, 26(4), 122. https://doi.org/10.1007/s10291-022-01314-3

49. Mao Y, Zhu Y, Tang Z, Chen Z. A Novel Airspace Planning Algorithm for Cooperative Target Localization. Electronics. 2022; 11(18):2950. https://doi.org/10.3390/electronics11182950

50. Sun R., Dai Y., & Cheng Q. (2023). An Adaptive Weighting Strategy for Multisensor Integrated Navigation in Urban Areas. IEEE Internet of Things Journal, 10(14), 12777–12786. https://doi.org/10.1109/JIOT.2023.3256008

51. Wang Y., Sun R., Cheng Q., & Ochieng W. Y. (2024). Measurement Quality Control Aided Multisensor System for Improved Vehicle Navigation in Urban Areas. IEEE Transactions on Industrial Electronics, 71(6), 6407–6417. https://doi.org/10.1109/TIE.2023.3288188

52. Min H., Lei X., Wu X., Fang Y., Chen S., Wang W., et al. (2024). Toward interpretable anomaly detection for autonomous vehicles with denoising variational transformer. Engineering Applications of Artificial Intelligence, 129, 107601. https://doi.org/10.1016/j.engappai.2023.107601

53. Min H., Li Y., Wu X., Wang W., Chen L., Zhao X. (2023). A Measurement Scheduling Method for Multi-vehicle Cooperative Localization Considering State Correlation. Vehicular Communications. https://doi.org/10.1016/j.vehcom.2023.100682

54. Hou X., Zhang L., Su Y., Gao G., Liu Y., Na Z., et al. (2023). A space crawling robotic bio-paw (SCRBP) enabled by triboelectric sensors for surface identification. Nano Energy, 105, 108013. https://doi.org/10.1016/j.nanoen.2022.108013

55. Hou X., Xin L., Fu Y., Na Z., Gao G., Liu Y., et al. (2023). A self-powered biomimetic mouse whisker sensor (BMWS) aiming at terrestrial and space objects perception. Nano Energy, 118, 109034. https://doi.org/10.1016/j.nanoen.2023.109034

56. Sun L., Liang J., Zhang C., Wu D., & Zhang Y. (2023). Meta-Transfer Metric Learning for Time Series Classification in 6G-Supported Intelligent Transportation Systems. IEEE Transactions on Intelligent Transportation Systems. https://doi.org/10.1109/TITS.2023.3250962

57. Liu D., Cao Z., Jiang H., Zhou S., Xiao Z., Zeng F. (2022). Concurrent Low-Power Listening: A New Design Paradigm for Duty-Cycling Communication. ACM Trans. Sen. Netw., 19(1). https://doi.org/10.1145/3517013

58. Xiao Z., Shu J., Jiang H., Lui J. C. S., Min G., Liu J., et al. (2022). Multi-Objective Parallel Task Offloading and Content Caching in D2D-aided MEC Networks. IEEE Transactions on Mobile Computing. https://doi.org/10.1109/TMC.2022.3199876

59. Jiang H., Chen S., Xiao Z., Hu J., Liu J., Dustdar S. (2023). Pa-Count: Passenger Counting in Vehicles Using Wi-Fi Signals. IEEE Transactions on Mobile Computing. https://doi.org/10.1109/TMC.2023.3263229

60. Hu J., Wu Y., Li T., & Ghosh B. K. (2019). Consensus Control of General Linear Multiagent Systems with Antagonistic Interactions and Communication Noises. IEEE Transactions on Automatic Control, 64(5), 2122–2127. https://doi.org/10.1109/TAC.2018.2872197

61. Chen B., Hu J., Zhao Y., & Ghosh B. K. (2022). Finite-Time Velocity-Free Rendezvous Control of Multiple AUV Systems with Intermittent Communication. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52(10), 6618–6629. https://doi.org/10.1109/TSMC.2022.3148295

62. Bo C., Jiangping H., & Bijoy G. (2023). Finite-Time Observer Based Tracking Control of Heterogeneous Multi-AUV Systems with Partial Measurements and Intermittent Communication. SCIENCE CHINA Information Sciences. https://doi.org/10.1007/s11432-023-3903-6

63. Jiang Y., & Li X. (2022). Broadband cancellation method in an adaptive co-site interference cancellation system. International journal of electronics, 109(5), 854–874. https://doi.org/10.1080/00207217.2021.1941295

64. Zhang X., Deng H., Xiong Z., Liu Y., Rao Y., Lyu Y., et al. (2024). Secure Routing Strategy Based on Attribute-Based Trust Access Control in Social-Aware Networks. Journal of Signal Processing Systems. https://doi.org/10.1007/s11265-023-01908-1

65. Lyu T., Xu H., Zhang L., & Han Z. (2024). Source Selection and Resource Allocation in Wireless-Powered Relay Networks: An Adaptive Dynamic Programming-Based Approach. IEEE Internet of Things Journal, 11(5), 8973–8988. https://doi.org/10.1109/JIOT.2023.3321673

66. Liu G., "Data Collection in MI-Assisted Wireless Powered Underground Sensor Networks: Directions, Recent Advances, and Challenges," in IEEE Communications Magazine, vol. 59, no. 4, pp. 132–138, April 2021, https://doi.org/10.1109/MCOM.001.2000921

67. Shi J., Li Z., Jia J., Li Z., Shen C., Zhang J., et al. (2023). Waveform-to-Waveform End-to-End Learning Framework in a Seamless Fiber-Terahertz Integrated Communication System. Journal of Lightwave Technology, 41(8), 2381–2392. https://doi.org/10.1109/JLT.2023.3236400

68. Jiang Z., & Xu C. (2023). Disrupting the Technology Innovation Efficiency of Manufacturing Enterprises Through Digital Technology Promotion: An Evidence of 5G Technology Construction in China. IEEE Transactions on Engineering Management. https://doi.org/10.1109/TEM.2023.3261940

69. Cao K., Ding H., Li W., Lv L., Gao M., Gong F., et al. (2022). On the Ergodic Secrecy Capacity of Intelligent Reflecting Surface Aided Wireless Powered Communication Systems. IEEE Wireless Communications Letters, PP, 1. https://doi.org/10.1109/LWC.2022.3199593

70. Cheng B., Wang M., Zhao S., Zhai Z., Zhu D., Chen J. (2017). Situation-Aware Dynamic Service Coordination in an IoT Environment. IEEE/ACM Transactions on Networking, 25(4), 2082–2095. https://doi.org/10.1109/TNET.2017.2705239

71. Zheng W., Lu S., Yang Y., Yin Z., Yin L., Ali H. (2024). Lightweight transformer image feature extraction network. PeerJ Computer Science, 10, e1755. https://doi.org/10.7717/peerj-cs.1755

72. Zheng W., Gong G., Tian J., Lu S., Wang R., Yin Z., et al. (2023). Design of a Modified Transformer Architecture Based on Relative Position Coding. International Journal of Computational Intelligence Systems, 16(1), 168. https://doi.org/10.1007/s44196-023-00345-z

73. Cao B., Zhao J., Gu Y., Fan S., & Yang P. (2020). Security-Aware Industrial Wireless Sensor Network Deployment Optimization. IEEE Transactions on Industrial Informatics, 16(8), 5309–5316. https://doi.org/10.1109/TII.2019.2961340

74. Cao B., Zhao J., Yang P., Gu Y., Muhammad K., Rodrigues J. J. P. C., et al. (2020). Multiobjective 3-D Topology Optimization of Next-Generation Wireless Data Center Network. IEEE Transactions on Industrial Informatics, 16(5), 3597–3605. https://doi.org/10.1109/TII.2019.2952565

75. Wu Y., Hao Y., Yong L., Rong X., "A clustering routing algorithm based on wolf pack algorithm for heterogeneous wireless sensor networks," *Computer Networks*, vol. 167, no. 3, pp. 1069–1078, 2020.

76. Abdulzahra A.M.K., Al-Qurabat A.K.M. and Abdulzahra S.A., 2023. Optimizing energy consumption in WSN-based IoT using unequal clustering and sleep scheduling methods. Internet of Things, 22, p.100765.

77. Kingston Roberts M. and Thangavel J., 2023. An improved optimal energy aware data availability approach for secure clustering and routing in wireless sensor networks. Transactions on Emerging Telecommunications Technologies, 34(3), p.e4711.

78. Gharghan S.K., Al-Kafaji R.D., Mahdi S.Q., Zubaidi S.L. and Ridha H.M., 2023. Indoor Localization for the Blind Based on the Fusion of a Metaheuristic Algorithm with a Neural Network Using Energy-Efficient WSN. Arabian Journal for Science and Engineering, 48(5), pp.6025–6052.

79. Roberts M.K. and Thangavel J., 2022. An optimized ticket manager based energy-aware multipath routing protocol design for IoT based wireless sensor networks. Concurrency and Computation: Practice and Experience, 34(28), p.e7398.

80. Gururaj H.L., Natarajan R., Almujally N.A., Flammini F., Krishna S. and Gupta S.K., 2023. Collaborative energy-efficient routing protocol for sustainable communication in 5G/6G wireless sensor networks. IEEE Open Journal of the Communications Society.

81. Al-hajjar A. and Al-qurabat A. An overview of machine learning methods in enabling IoMT-based epileptic seizure detection. The Journal of Supercomputing, 1–48, 2023. https://doi.org/10.1007/s11227-023-05299-9 PMID: 37359338

82. M. Jabar and A. A-qurabat Human activity diagnosis system based on the internet of things. In Journal of Physics: Conference Series, vol. 1879, no. 2, p. 022079. IOP Publishing, 2021.

83. Al-hajjar A. and Al-qurabat A. Epileptic Seizure Detection Using Feature Importance and ML Classifiers. Journal of Education for Pure Science-University of Thi-Qar, 13(2), 2023.

84. Raheem R. and Al-qurabat A. Developing a Predictive Health Care System for Diabetes Diagnosis as a Machine Learning-Based Web Service. Journal of University of Babylon for Pure and Applied Sciences, 1–32, 2022.

85. Abdulzahra A. and Al-qurabat A. A Clustering Approach Based on Fuzzy C-Means in Wireless Sensor Networks for IoT Applications. Karbala International Journal of Modern Science: Vol. 8: Iss. 4, Article 2, 2022. https://doi.org/10.33640/2405-609X.3259

86. Abdulzahra A., Al-qurabat A., and Abdulzahra S. Optimizing energy consumption in WSN-based IoT using unequal clustering and sleep scheduling methods. Internet of Things, 2023. https://doi.org/10.1016/j.iot.2023.100765

87. Al-qurabat A., Mohammed Z., and Hussein Z. Data Traffic Management Based on Compression and MDL Techniques for Smart Agriculture in IoT. Wireless Personal Communications, 120(3), pp.2227–2258, 2021.

88. Al-qurabat A. A Lightweight Huffman-based Differential Encoding Lossless Compression Technique in IoT for Smart Agriculture. International Journal of Computing and Digital System, 11(1), 117–127, 2022.

89. Saeedi I. and Al-qurabat A. Perceptually Important Points-Based Data Aggregation Method for Wireless Sensor Networks. Baghdad Science Journal, 19(4), pp. 875–886, 2022.

90. A. Al-qurabat and S. Abdulzahra. An Overview of Periodic Wireless Sensor Networks to the Internet of Things. IOP Conference Series: Materials Science and Engineering, 928(3), 2020.

91. Finjan A., Salman H., and Al-qurabat A. Important extrema points extraction-based data aggregation approach for elongating the WSN lifetime. International Journal of Computer Applications in Technology, 68(4), 357–368, 2022.

92. I. Saeedi and A. Al-qurabat. An energy-saving data aggregation method for wireless sensor networks based on the extraction of extrema points. In AIP Conference Proceedings (Vol. 2398, No. 1, p. 050004), 2022. AIP Publishing LLC.

93. W. Nedham and A. Al-qurabat. An Improved Energy Efficient Clustering Protocol for Wireless Sensor Networks. 2022 International Conference for Natural and Applied Sciences (ICNAS), 2022, pp. 23–28.

94. Abdulzahra A. and Al-qurabat A. An Energy-Efficient Clustering Protocol for the Lifetime Elongation of Wireless Sensors in IoT Networks. In: Abu Bakar M.H., Abdul Razak T.A., Öchsner A. (eds) IT Applications for Sustainable Living. SpringerBriefs in Applied Sciences and Technology. Springer, Cham.

95. Bassam W. and Al-qurabat A. A comprehensive review of clustering approaches for energy efficiency in wireless sensor networks. International Journal of Computer Applications in Technology, 72(2), 139–160, 2023.

96. Bassam W. and Al-qurabat A. A review of current prediction techniques for extending the lifetime of wireless sensor networks. International Journal of Computer Applications in Technology, 71(4), 352–362, 2023.