IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 23 January 2024, accepted 14 February 2024, date of publication 14 March 2024, date of current version 20 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3375886

RESEARCH ARTICLE

Coverage and Routing Optimization of Wireless Sensor Networks Using Improved Cuckoo Algorithm

JIAN YANG[1] AND YIMIN XIA[2]
[1]School of Automation, Guangdong University of Technology, Guangzhou 510006, China
[2]School of Integrated Circuits, Guangdong University of Technology, Guangzhou 510006, China

Corresponding author: Jian Yang (yangjiangdut@163.com)

ABSTRACT Wireless sensor networks have gradually attracted widespread attention from academia and industry due to their wide application potential and versatility. In response to the coverage and routing issues of wireless sensor networks, this study combines the improved cuckoo algorithm and non-uniform clustering algorithm to design a coverage and routing optimization scheme for wireless sensor networks. Firstly, a wireless sensor network system is studied and built. Subsequently, the traditional cuckoo algorithm is optimized using the Cauchy distribution, and a coverage optimization scheme for wireless sensor networks based on the Cauchy distribution improved cuckoo algorithm is designed. Finally, by optimizing the dynamic range of network area and cluster radius respectively to improve the shortcomings of non-uniform clustering routing algorithms, a wireless sensor networks routing optimization scheme based on dynamic cluster radius optimization non-uniform clustering algorithm is designed. The research results indicated that the designed coverage scheme and routing scheme achieved good application results. Among them, the designed coverage scheme had a minimum running time of 0.89 minutes in node coverage problems and a maximum node coverage rate of 99.52%. The designed routing scheme had a minimum running time of 1.47 minutes and an energy loss rate of 0.84% in the routing optimization problem. In summary, the coverage scheme and routing optimization scheme designed by this research institute have good application effects and can provide certain technical support for the application of wireless sensor networks in other fields.

INDEX TERMS Cuckoo algorithm, wireless network, sensors, router, signal coverage, node.

I. INTRODUCTION

Wireless sensor networks (WSNs) are composed of spatially distributed autonomous sensors, commonly used to monitor physical or environmental conditions such as temperature, sound, pressure, and other data, and transmit the collected data to the host for in-depth analysis through the network [1], [2]. Although WSNs are widely used, their design and optimization still face many challenges. Among them, network coverage and routing selection are the two main issues in WSNs [3]. The network coverage problem focuses on how to use the minimum number of sensors to cover the area of interest, while the routing problem focuses on how to choose the optimal data transmission path to ensure data accuracy and minimize energy consumption. Traditional clustering routing algorithms typically divide the network evenly into multiple clusters, with each cluster selecting a cluster head for data forwarding. This routing algorithm method may be effective in uniformly distributed networks, but in environments with non-uniform distribution or large changes in network density, it may cause cluster heads to consume energy too quickly, thereby reducing the network's life-cycle [4]. The cuckoo bird optimization algorithm is a new heuristic algorithm proposed in recent years, which has performed well in many optimization problems. However, its application in WSNs coverage and routing optimization is still in its early stages [5]. The wide range of applications of WSNs in various industries, such as environmental

The associate editor coordinating the review of this manuscript and approving it for publication was Pietro Savazzi.

monitoring, health care, and industrial automation, highlights their great potential as a versatile technology. This research is dedicated to solving one of the core problems in WSNs, i.e., optimizing network coverage and data routing. To this end, the research proposes a comprehensive scheme combining the improved cuckoo search algorithm(CSA)and the non-uniform cluster routing algorithm. The research focuses on detailing the design and implementation of these two algorithms and exploring how they work together to improve the performance of WSNs, especially in complex environments. In addition, the research will explore the application of these algorithms in different network types and conditions, and their importance in improving the efficiency and reliability of WSNs. This research aims to provide a more efficient and reliable approach to the design and implementation of WSNs, thereby promoting their widespread use in a variety of practical applications.

This research is divided into six parts, the first part is an introduction to the article, the second part is an analysis and summary of the related research of others, the third part is an introduction to the methodology of the article, the fourth part is an analysis of the performance of the algorithmic model, the fifth part is a discussion of the article, and the sixth part is a summary of the full text.

II. RELATED WORKS

CSA is an optimized search algorithm based on the parasitic strategy of the Cuckoo. This algorithm was proposed by Xin-She Yang and Suash Deb in 2009. CSA is popular for its simplicity and efficiency in various optimization problems. S. T. Shishavan et al. improved the cuckoo bird search optimization algorithm through a genetic algorithm and applied the optimized algorithm to community detection in complex networks. CSA has problems such as premature convergence, delayed convergence, and falling into local optima. Genetic algorithms can optimize the shortcomings of the cuckoo bird search algorithm, thereby increasing the search and utilization efficiency of the algorithm. Finally, the modular objective function and standardized mutual information were used as optimization functions to optimize the hybrid algorithm. The research results indicated that the cuckoo bird search algorithm optimized by genetic algorithm had better performance, and compared with other algorithms, the optimization algorithm had an average improvement of 54% in search accuracy in terms of modularity [6]. In medical image processing, fundus images are often affected by non-uniform lighting, low contrast, and noise. Therefore, image pre-processing is necessary to enhance the quality of the original fundus image. Regarding such issues, D Toresa et al. evaluated the effectiveness of various optimization algorithms in selecting the best technology to identify diabetes retinopathy. The image pre-processing techniques compared in the study included various types of contrast stretching and were finally evaluated using standard performance indicators such as mean square error and entropy. The research results indicated that the images processed using the cuckoo bird search algorithm had better performance, and their image quality was significantly higher than that of the comparative images [7]. H. Zavieh et al. combined the cuckoo algorithm and particle swarm optimization algorithm to design a new hybrid algorithm. This hybrid algorithm can determine which virtual machine can be assigned to each host, thereby selecting the best virtual machine. Moreover, when the selected host is overloaded, the hybrid algorithm can determine which virtual machines are generating high loads and migrate them to another host. The research results indicated that compared to individual intelligent algorithms, the proposed hybrid algorithm had lower energy consumption and faster execution speed in the CloudSim simulation environment [8].WSNs have become a hot research direction in modern information technology due to their widespread applications in various fields such as military reconnaissance, environmental monitoring, and smart homes. To extend WSNs' survival time, G Tong et al. proposed a particle swarm optimization routing scheme for energy consumption issues between nodes in WSNs. This scheme considered the factors of residual energy and node distance when selecting cluster heads and found the optimal transmission path through the ant colony algorithm. The final research results indicated that the proposed optimized routing scheme significantly improved the usage time of network survival nodes [9]. To extend the WSNs' life-cycle, S U. Sankar et al. adopted a clustering strategy to optimize the energy consumption of batteries. They developed a secret protocol and determined the distribution of cluster heads and nodes in this protocol. Finally, the information was transmitted to the base station through the determined optimal path to achieve effective data transmission. The research results indicated that the designed secret protocol effectively reduced WSNs' energy consumption, thereby extending their life-cycle [10]. The comparative analysis of relevant studies is shown in Table 1.

In summary, the current cuckoo algorithm has shown good application results in many optimization problems, but its application in WSNs coverage and routing optimization is still in its early stages. Therefore, this study combines the cuckoo bird algorithm with WSNs and optimizes the algorithm to solve node coverage and routing optimization problems in WSNs, aiming to improve WSNs' performance in various application scenarios.

III. WIRELESS SENSOR NETWORKS COVERAGE AND ROUTING OPTIMIZATION BASED ON IMPROVED CUCKOO BIRD AND NON-UNIFORM CLUSTERING ROUTING ALGORITHM

To ensure effective and accurate transmission of sensor data, the coverage and routing strategies of WSNs are particularly crucial. This study optimizes the coverage strategy and routing settings of WSNs using the improved cuckoo algorithm and non-uniform clustering routing algorithm, aiming to further improve the network's coverage range and data transmission capability.

TABLE 1. Comparative analysis of relevant studies.

Author	Article title	Research characteristics	Literature number
S. T. Shishavan, and F. S. Gharehchopogh	An improved cuckoo search optimization algorithm with genetic algorithm for community detection in complex networks	Improvement of cuckoo search optimization algorithms via genetic algorithms applied to community detection in complex networks	[6]
D. Toresa et al.	The Cuckoo Optimization Algorithm Enhanced Visualization of Morphological Features of Diabetic Retinopathy	Evaluating the effectiveness of multiple optimization algorithms in selecting the best technique for identifying diabetic retinopathy	[7]
H. Zavieh et al.	Task processing optimization using cuckoo particle swarm (CPS) algorithm in cloud computing infrastructure	Combining the cuckoo and particle swarm optimization algorithms to design a hybrid algorithm for virtual machine allocation	[8]
G. Tong et al.	A particle swarm optimization routing scheme for WSNs	Proposing a particle swarm optimized routing scheme for the energy consumption problem among nodes in WSNs	[9]
S. U. Sankar et al.	Secure and Energy Concise Route Revamp Technique in WSNs	Optimizing WSNs battery energy consumption using clustering strategy and designing secret protocols for efficient data transmission	[10]

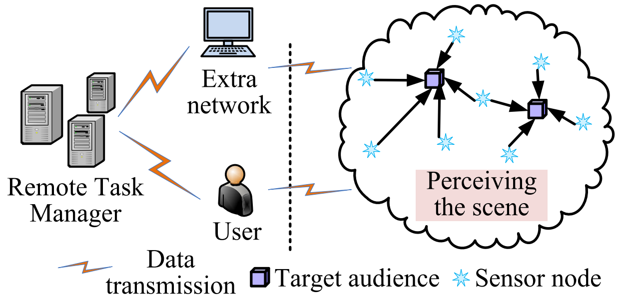

FIGURE 1. WSNs architecture diagram.

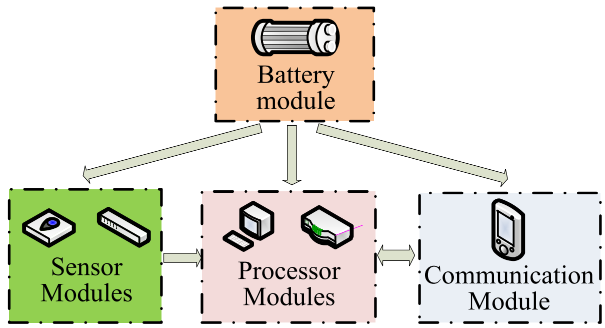

FIGURE 2. Structure of wireless sensor node combination.

A. WIRELESS SENSOR NETWORKS SYSTEM ARCHITECTURE DESIGN

WSNs refer to a network composed of small and inexpensive sensor nodes. These nodes communicate wirelessly to jointly achieve information perception and data collection in the target area [11], [12]. In complete WSNs, there are generally three types of nodes, namely data processing nodes, task scheduling management nodes, and data aggregation and transmission nodes. Each of these three nodes has a unique function and role. First, the data processing node is responsible for receiving the raw data collected by the other sensing nodes and performing the necessary processing, such as data filtering, compression, and preliminary analysis, to ensure the accuracy of the data and to prepare for the subsequent steps. Second, task-scheduling management nodes play a key role in the network and are responsible for the resource management of the overall network and the scheduling of data transmission tasks. Such nodes optimize the performance and efficiency of the entire network by dynamically allocating network resources and adjusting data transmission schedules. Finally, data aggregation and transmission nodes, as an important part of the network, are mainly responsible for collecting data from each data processing node, performing effective data aggregation, and transmitting the aggregated data to the central server or base station. These nodes ensure effective data integration and efficient transmission and play a key role in ensuring network lifetime and data transmission reliability. The common architecture diagram of WSNs is shown in Figure 1.

In Figure 1, the key elements that make up the WSNs system include various sensor nodes, external networks, remote task scheduling management centers, users, perception sites, and data transmission targets. To achieve information transmission and monitoring, WSNs first perceive information through their deployed sensor nodes in the target area. These sensors can monitor various types of information, such as sound, vibration, etc. After collecting data, sensor nodes can perform preliminary data pre-processing to reduce the data that need transmission and improve data accuracy. After data processing, data fusion technology in the WSNs system integrates data from multiple sensors to further reduce data volume and improve data quality. After confirming the target object and data transmission path, data are transmitted to an external network base station or data collection center, and then interact with the remote task scheduling management center to complete the final monitoring task. As a key part of WSNs systems, sensor nodes play a crucial role in collecting and processing data. The structural composition of sensor nodes is shown in Figure 2.

In Figure 2, the main modules that make up the wireless sensor node include the battery module, sensor module, processor module, and communication module. The battery module provides the energy supply required by sensor nodes. The sensor module is mainly responsible for the sensing function of nodes, ensuring that nodes can collect information within the sensing area. The processor module mainly processes the collected data information further to ensure the quality and efficiency of data transmission. The communication module is responsible for data transmission between nodes. In general, sensor nodes have six energy consumption states: sensing, computing, sending, receiving, idle, and sleeping. Among them, the node state that consumes the most energy is sending and receiving [13].

The node coverage problem in WSNs is the core research content, and higher coverage means more accurate and stable monitoring results. The node coverage problem in WSNs essentially explores how to maximize the coverage area under a limited number of nodes or ensure complete monitoring with the minimum number of nodes. When designing coverage algorithms for WSNs, it not only meets the requirements of monitoring coverage but also optimizes node energy consumption and improves the long-term utilization efficiency of the network. The perception range, communication range, and network coverage of nodes can all be used for the performance evaluation of coverage algorithms. The common types of the WSNs coverage are mainly divided into the following three types, as shown in Figure 3.

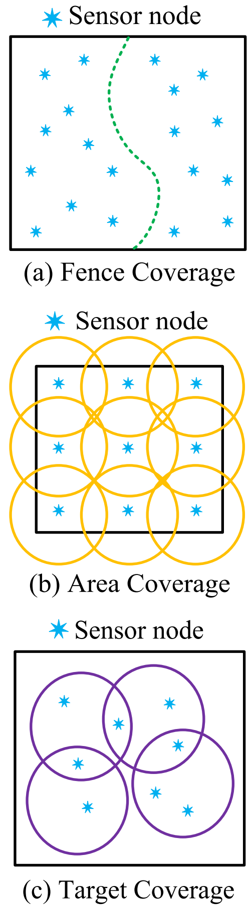

FIGURE 3. Wireless sensor networks coverage types.

In Figure 3, three different types of the WSNs coverage are introduced, namely fence coverage, area coverage, and target coverage [14], [15]. The main focus of fence coverage is how to use the least number of sensors to form a sensor ''fence'' to detect or block any target passing through this fence. The main goal of fence coverage is to ensure that any moving target crossing this fence will be detected by at least one sensor. Regional coverage focuses on how to use sensor networks to cover a specific area, ensuring that any events or changes within that area can be detected. The goal is to maximize coverage within the region of interest while minimizing the number or energy consumption of sensors. Target coverage focuses on how to use a sensor network to cover a specific set of target nodes, rather than the entire area of nodes. For example, there may only be a few key locations within a large area that require special monitoring. The purpose of target coverage is to ensure that all these key target points are covered by enough sensors to ensure data accuracy and reliability.Fence coverage generally places sensors around a specific area (e.g., around a borderline or critical facility) to form a sensor ''fence'' to monitor or prevent any unauthorized crossings. For example, sensors may be placed along national borders to detect illegal crossings. Area coverage involves deploying sensors over a larger area (e.g., a farm or forest) to ensure that events or changes (e.g., temperature changes, fires, or trespassing) throughout the area are detected promptly. Targeted coverage focuses on monitoring a specific set of target points (e.g., critical infrastructure or a specific geographic location) to ensure the security and operational status of these important points.

In addition to studying the node coverage problem in WSNs, it is also necessary to optimize the routing strategies in WSNs. The routing strategy in WSNs determines the transmission path and form of the sensor node. At present, how to complete routing tasks with minimal energy consumption is the main research direction of routing problems. A routing protocol is a set of rules and conventions that define information exchange between routers in a network. Routing protocols enable routers to learn and maintain network topology information, and based on this, decide how to route data packets from the source address to the destination address. The common routing topology is shown in Figure 4.

Figure 4 presents two routing topologies for WSNs, namely hierarchical routing protocols and planar routing protocols. These two classifications represent different methods of routing information and data transmission. The main advantage of hierarchical routing protocols is their ability to reduce energy consumption and extend the life cycle of the network; however, the disadvantage is that the cluster head nodes may face a larger energy burden, and the

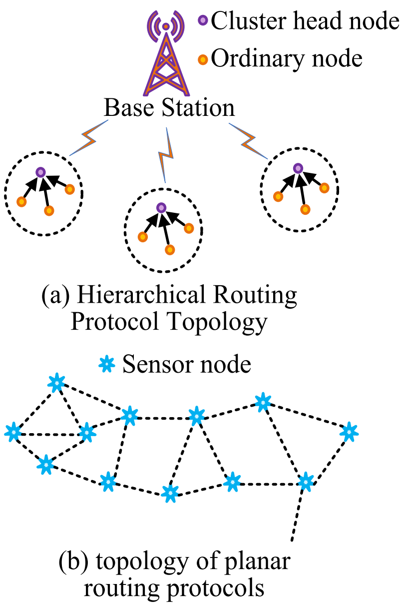

FIGURE 4. Types of routing topology for WSNs.

maintenance of the clusters may incur additional overheads. Planar routing protocols have the advantage of being simpler and easier to maintain, and they avoid the energy overload problem by not requiring cluster head nodes, thus being more cost-effective; the disadvantage is that they may not perform well with long-distance communication and high data transfer volumes, leading to less efficient routing, increased energy consumption, and network congestion. Hierarchical routing architectures are particularly suitable for situations that require long-term operation and frequent data transmission, such as environmental monitoring or agricultural applications. Planar routing protocols are usually suitable for scenarios where data is uniformly distributed and the network size is small or dense with nodes, such as urban transport or home automation. The choice of hierarchical routing protocols is usually motivated by the optimization of network energy consumption and the improvement of network management. The main motivation for choosing planar routing is to simplify the architecture and reduce the cost, especially in resource-limited environments.

B. WIRELESS SENSOR NETWORKS COVERAGE OPTIMIZATION BASED ON CAUCHY DISTRIBUTION AND IMPROVED CUCKOO BIRD ALGORITHM

With the WSNs' gradual application in various fields of human life, the sensing and detection capabilities of sensors are inevitably limited by different environments. Therefore, it is necessary to optimize them through joint intelligent algorithms [16]. Traditional CSA still has the drawbacks of slow convergence speed and low convergence accuracy, so it needs to be optimized to overcome a series of shortcomings of traditional CSA. Using optimized CSA to solve the coverage problem of WSNs can not only improve the networks' performance and efficiency, but also extend their lifespan, reduce costs, and ensure that the sensor network meets the needs of specific applications.

To ensure the smooth progress of subsequent simulation experiments, the research regards the network monitoring area as a two-dimensional plane, and first builds a perception model and an energy consumption model. The entire plane length is L, the height is H, the set of sensor nodes is denoted as $S = \{S_1, S_2, \cdots, S_N\}$, and the coordinate of the node i in the two-dimensional plane is denoted as (X_i, Y_i). Assuming that each node can move freely and is in a dormant state in its initial state. In addition, the perception radius and communication radius of each sensor node are R_p and R_c, respectively, $R_c = 2R_p$. The distance expression for node i and j is shown in equation (1) [17], [18].

$$d_{ij} = \sqrt{\left(X_i - X_j\right)^2 + \left(Y_i - Y_j\right)^2} \tag{1}$$

In equation (1), d_{ij} represents the distance between i and j. X_j and Y_j represent the horizontal and vertical coordinates of node j in the two-dimensional plane. The calculation expression for the coverage probability of the sensor node S_i is shown in equation (2).

$$P_{cov}\left[S_i, \left(X_g, Y_g\right)\right] = \begin{cases} 1 & d\left[S_i, \left(X_g, Y_g\right)\right] \leq R \\ 0 & d\left[S_i, \left(X_g, Y_g\right)\right] > R \end{cases} \tag{2}$$

In equation (2), P_{cov} represents the coverage probability. $\left[S_i, \left(X_g, Y_g\right)\right]$ represents the sensor node S_i in the coordinate $\left(X_g, Y_g\right)$ of the grid point g. R represents the radius of the entire detection area. d represents the distance between nodes. The expression for calculating the ratio R_{area} of the area perceived by the sensor node to the area to be measured is shown in equation (3).

$$R_{area} = \frac{\sum_{X=1}^{L}\sum_{Y=1}^{H} P_{cov}\left[S_i, (X, Y)\right]}{L \times H} \tag{3}$$

In equation (3), R_{area} represents the ratio, (X, Y) represents the overall coordinates of the grid. The calculation expression for communication energy consumption E_{Tx} between two nodes is shown in equation (4).

$$E_{Tx}(k, d) = \begin{Bmatrix} kE_{elec} + k\varepsilon_{fs}d^2, & d < d_0 \\ kE_{elec} + k\varepsilon_{mp}d^4, & d \geq d_0 \end{Bmatrix} \tag{4}$$

In equation (4), k and d respectively represent the size and distance of data transmission between two nodes. E_{elec} represents the transmission energy of the loss. d_0 represents the distance boundary point. If $d < d_0$, the free space model is used for E_{Tx} calculation. if not, the multi-path decay model is used for E_{Tx} calculation. ε_{fs} and ε_{mp} represent the power parameters of the two models.

After obtaining the perception model and energy consumption model, the research will use the cuckoo bird search nest idea in CSA to simulate the scenario where the node searches for the optimal perception environment in the lowest energy consumption state [19]. When optimizing, the first step is objective function calculation. The gradient descent method accelerates CSA optimization speed, and its optimization expression is shown in equation (5).

$$\theta_i' = \theta_i - \eta \frac{\partial}{\partial \theta_i} J(\theta) \tag{5}$$

In equation (5), $J(\theta)$ represents the loss function of CSA. η indicates the learning rate. θ_i' indicates the direction of gradient descent. θ_i indicates the opposite direction of gradient descent. ∂ represents a partial differential symbol. The momentum gradient descent method is used to optimize the flight step size in CSA in equation (6).

$$\begin{cases} \Delta l = \beta \Delta l_{t-1} + (1-\beta) l_t \\ l = l_0 - \eta' \Delta l \end{cases} \tag{6}$$

In equation (6), l and Δl respectively represent the flight step size and step size variation. β is a weight parameter. l_{t-1} and l_t represent the step size of $t-1$ and t time, respectively. l_0 represents the initial value step size, with a value of 1. η' represents the learning rate of step size updates. The root mean square calculation method is further used to optimize the learning rate in the horizontal and vertical directions in equation (7).

$$\begin{cases} S_{dxy} = \dfrac{1}{n} \displaystyle\sum_{1}^{n} |(x - x_{best})| \\ V = V_0 - \omega \dfrac{1}{\sqrt{S_{dxy} + \varepsilon}} \end{cases} \tag{7}$$

In equation (7), S_{dxy} represents the average distance from each nest to the optimal location. x and x_{best} represent the position and optimal nest position of each bird's nest, respectively. V and V_0 respectively represent the updated optimization speed and initial speed. n counts bird nests. ω denotes the root mean square calculation method learning rate. ε represents a variable. After optimizing various learning rates through the above methods, the study finally combines the Cauchy mutation algorithm to optimize the local optimal solution of CSA. By incorporating the optimized CSA into the sensor coverage problem, the sensor node distribution under Cauchy mutation is obtained as shown in equation (8).

$$S = \frac{\tau}{\pi \left(x^2 + \tau^2\right)}, \quad -\infty < x < +\infty \tag{8}$$

In equation (8), τ is the scale parameter. Since the law of Cauchy mutation first monotonically increases and then monotonically decreases, applying the Cauchy mutation rule to CSA algorithms can effectively reduce the local search time of CSA, thus enabling it to better spread globally and detect the coverage of all wireless network sensor nodes better. The Cauchy distribution calculation expression is obtained through equation (8) as shown in equation (9).

$$x' = x + \gamma Cauchy(0, 1) \tag{9}$$

In equation (9), x' represents the nest position obtained by optimizing CSA using Cauchy mutation rules. γ represents the influencing factor. $Cauchy(0, 1)$ represent the Cauchy random distribution when $\tau = 1$. γ is calculated in equation (10).

$$\gamma = \frac{\theta \left[f(x) - f(x)_{worst}\right]}{T} \tag{10}$$

In equation (10), θ represents the Cauchy variation intensity coefficient. $f(x)$ represents the probability that the current bird's nest is covered. $f(x)_{worst}$ is the lowest value among all bird nest coverage rates. T is the max iteration.

The optimized cuckoo algorithm after the Cauchy variation algorithm is denoted as the Cauchy Distribution-Cuckoo Search Algorithm (CD-CSA). In this paper, we choose to use CD-CSA to optimize the coverage problem of WSNs because the Cauchy variation can effectively improve the exploration ability of the algorithm and reduce the risk of falling into local optimal solutions. This improvement enables the CD-CSA algorithm to explore the entire solution space more effectively when dealing with the coverage problem of sensor nodes, thus achieving better coverage and higher computational efficiency. Its operation flowchart is shown in Figure 5.

In Figure 5, when running the CD-CSA algorithm, all cuckoo populations and algorithm parameters need to be initialized first. Next, countless nest locations will be randomly generated, and the coverage rate of each nest and the nest location with the highest coverage value will be recorded. Then, the step size will be calculated based on equation (6) above, and the current position of the bird's nest will be determined using the calculated step size value. The Cauchy distribution will be introduced to update the current position of bird nests, and a comparison will be made to determine if their updated coverage is higher than that of the previous generation of bird nests. If the current coverage is not higher than the previous generation, it is determined whether iteration is max. If max, the coverage value is output at this time and the algorithm is ended.

C. WIRELESS SENSOR NETWORKS ROUTING OPTIMIZATION BASED ON DYNAMIC CLUSTER RADIUS OPTIMIZATION NON-UNIFORM CLUSTERING ALGORITHM

In addition to optimizing the coverage of WSNs, the nodes' energy loss is considered during their work process. Therefore, it is necessary to design high-performance routing algorithms to improve nodes utilization efficiency and reduce their energy loss and transmission time. Among the common routing algorithms, the clustered routing algorithm is widely used due to its powerful node collection and processing capabilities. The non-uniform cluster routing algorithm is an improved algorithm of cluster routing algorithm, which generates clusters of varying sizes by setting the node radius competition rules. Then, the node data is received through these cluster head nodes which reduces energy loss. Although the non-uniform cluster routing algorithm can effectively

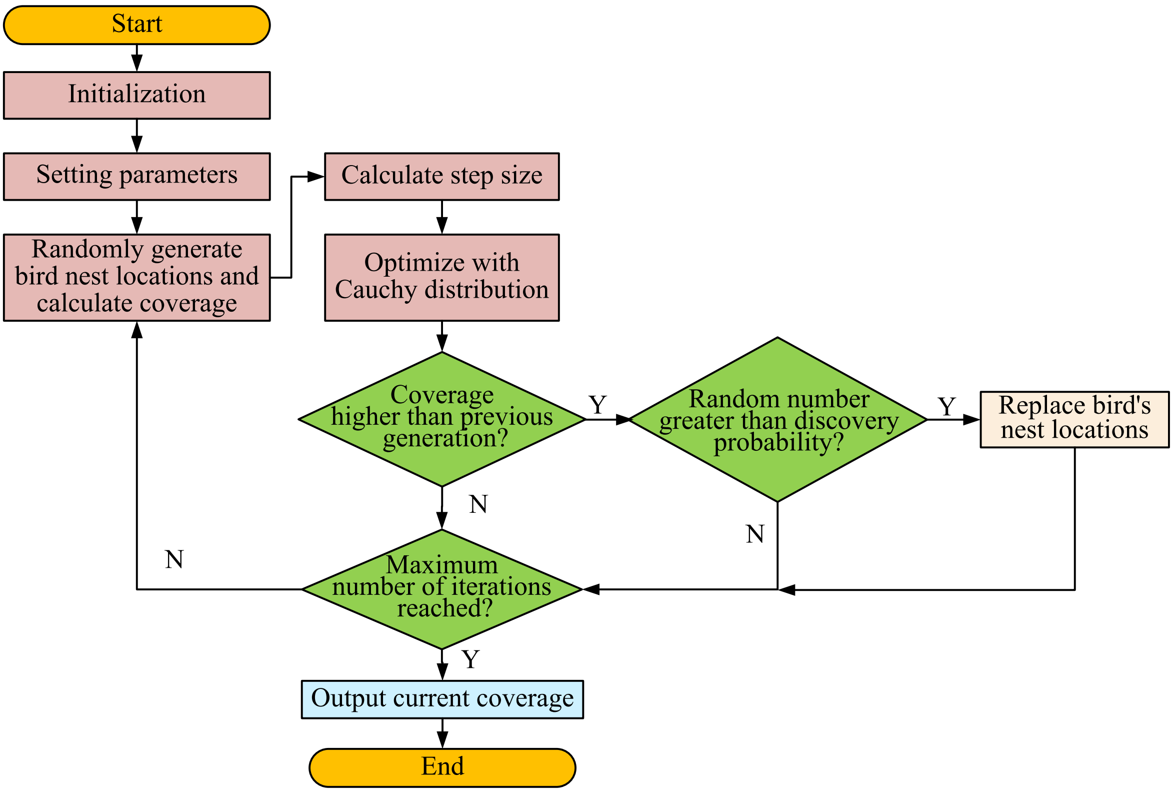

FIGURE 5. Flow chart of CD-CSA operation.

reduce the energy loss, this algorithm is prone to problems such as uneven distribution of cluster heads and node failure because it does not fully consider the positional relationship of each node in space [20]. Based on this background, this study improves the shortcomings of the non-uniform cluster routing algorithm by optimizing the dynamic range of the network area and cluster radius respectively, and finally proposes a Dynamic Cluster Radius Uneven Clustering (DCRUC) algorithm based on dynamic cluster radius optimization, which aims to better optimize the nodes' routing strategies to improve the utilization of wireless network systems. DCRUC can better cope with the challenges of uneven network density and node energy constraints by dynamically adjusting the competitive radius of cluster head nodes. This approach effectively reduces energy consumption and improves the efficiency of data transmission and network stability, especially in environments with uneven node distribution.

Firstly, the WSNs distribution space is optimized. The WSNs area is divided into two types of transmission areas: direct and indirect, and sequentially number each area to enable cluster head nodes to prioritize selection based on node positions and node energy in different areas, thus achieving the goal of distributing cluster head nodes evenly. The optimized WSNs distribution diagram is shown in Figure 6.

In Figure 6, the spatial distribution of the WSNs is divided into four regions. The base station divides each area by transmitting messages to nodes within the sensing area. Region 1 is designated as a direct transmission region, while Region 2, Region 3, and Region 4 are designated as indirect transmission regions. Dividing the space of WSNs into regions can enable cluster head nodes to better select nearby nodes for energy and data collection and make the distribution of cluster heads more uniform [21].

From the WSNs energy consumption calculation expression, it is found that receiving and sending data are the main energy consumptions of cluster head nodes. To reduce them, it is necessary to appropriately reduce the cluster radius of cluster head nodes. Therefore, further research will propose an optimization model for cluster radius. Assuming that each cluster head node is a regular circular shape with a radius of R_1 and a density of ρ, the expression for calculating nodes within the cluster is obtained as shown in equation (11).

$$n_1 = \pi R_1^2 \rho \tag{11}$$

In equation (11), n_1 represents the number of nodes within the cluster. The total energy consumption calculation expression is shown in equation (12).

$$E_{total} = \left(\pi R_1^2 \rho \cdot E_1 + E_2\right) \cdot E'_{elec} + E_3 \cdot \left(E'_{elec} + \varepsilon_{fs} d^2\right) \tag{12}$$

In equation (12), E_{total} represents the total energy consumption. E_1 and E_2 represent a single node receiving data consumption and consumption from a cluster head node. E_3 and E'_{elec} represent the current cluster head node sending data consumption and the energy consumption during communication between the sender and receiver. From equation (12), it can be found that the nodes' radius and the transmission distance between nodes will affect the nodes' consumption. Therefore, by optimizing these two parameters, energy loss can be reduced by reducing the cluster radius. The competition radius is calculated in equation (13).

$$r = \left(1 - c\frac{d_{\max} - d_p}{d_{\max} - d_{\min}}\right) * R_0 \tag{13}$$

In equation (13), c represents the influencing factor. R_0 is the maximum node communication radius. $d_{\max}$ and $d_{\min}$ represent the maximum and minimum distance from a node to a base station, and d_p represents the distance from node p to a base station. Due to the fixed competition radius of cluster head nodes in equation (13), overloaded computing tasks can easily lead to premature failure with an unchanged competition radius when the algorithm runs to a later stage. To avoid this situation, the expression for calculating the competition radius of cluster head nodes in equation (14) is proposed for optimization [22].

$$r' = \left[\lambda_1 * \left(1 - c\frac{d_{\max} - d_p}{d_{\max} - d_{\min}}\right) + \lambda_2\frac{E^p_{res}}{E_{init}}\right] * R_0 \tag{14}$$

In equation (14), two influence factors, λ_1 and λ_2, are introduced, with values ranging from [0,1]. r' represents the optimized cluster head node competition radius value. E_{init} and E^p_{res} represent the initial energy values of all nodes and the p current remaining energy values, respectively [23]. Combining the equations for calculating the competition radius with the area division rules of WSNs, the expression for calculating the competition radius in indirect transmission areas is shown in equation (15).

$$\begin{aligned} r'' = &\left[1 - c\frac{d_{\max} - d_p}{d_{\max} - d_{\min}}\right] * R_0 \\ &+ \left[\beta_1 * Sgn\left(E^p_{res} - E^n_{ave}\right) + \beta_2\frac{n_p}{m}\right] * \Delta R \end{aligned} \tag{15}$$

In equation (15), β_1 and β_2 are the two optimized influencing factors, whose value range is still between [0, 1]. r'' represents the optimized cluster head node competition radius value in the indirect transmission area. n_p indicates the region where the node p is located. m is the number of regions. E^n_{ave} represents the average remaining energy in the region n. ΔR represents the adjustment value of the competition radius [24]. By combining the region division rule with the cluster head node competition radius optimization rule through equations (11) to (15), a non-uniform clustering algorithm based on dynamic cluster radius optimization is obtained. The flowchart of the DCRUC algorithm in the WSNs routing optimization problem is shown in Figure 7.

In Figure 7, the base station broadcasts a message to the nodes in the sensing area in advance to inquire about their

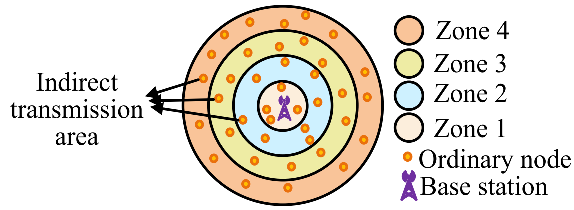

FIGURE 6. Schematic diagram of WSNs distribution.

information. Then, the WSNs need to be divided into several regions and the competition radius needs to be calculated. Next, based on the above expression, the energy consumption and competition radius optimization value are calculated, and the algorithm is ended through a series of judgment rules. After integrating all the feedback data received, the base station can automatically modify the nodes' radius based on the energy consumption expression until the minimum node competition radius is obtained in the allowed state.

IV. SIMULATION EXPERIMENT SETTINGS AND RESULT ANALYSIS

To better demonstrate CD-CSA and DCRUC performance, the study first established coverage simulation environments and routing simulation environments for algorithm performance testing. In addition, the study applied two algorithms to the WSNs system, and optimized network coverage and routing strategies in the system in practical applications, further proving that the two algorithms also perform better in practical applications.

A. COVERAGE OPTIMIZATION ALGORITHMS FOR WSNS PERFORMANCE ANALYSIS

To test the proposed CD-CSA performance in wireless sensor node coverage, the algorithm was studied to calculate the coverage of sensor nodes, and CD-CSA was compared with traditional CSA and Principal Component Analysis Optimized Cuckoo Search Algorithm (PCA-CSA). The parameter settings for covering the simulation experimental environment are shown in TABLE 2.

Table 2 provides the specific parameter indicators and values for this coverage simulation experiment. The selected parameter values in Table 2 were intended to reflect a typical small-scale WSNs environment. An experimental area of $100m^2$ and 40 nodes was chosen to simulate a densely deployed sensor network, and a sensing radius of 10m and a communication radius of 20m were effective in demonstrating the interactions between nodes.500 iterations and a learning rate of 0.5 were set to ensure that the algorithm converged to a stable solution in a limited amount of time. These parameter values brought the simulation closer to real-world applications, thus increasing the realism and relevance of the study. Under the parameter conditions in Table 2, sensor nodes were randomly arranged and the optimization

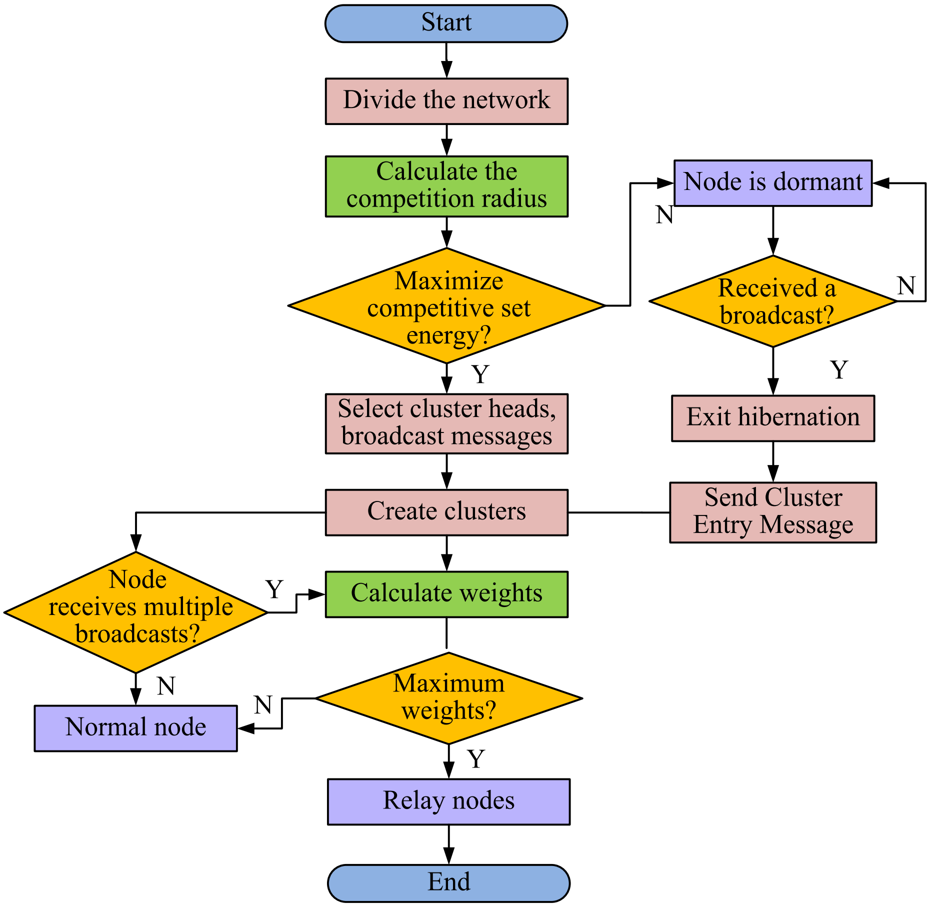

FIGURE 7. Flowchart of DCRUC algorithm operation.

results of different algorithms for sensor node area coverage were obtained, as shown in Figure 8.

Figure 8 shows the random coverage of nodes and the coverage optimization of nodes under three coverage algorithms. In Figure 8 (a), there are 36 nodes initially, which are randomly distributed in the spatial coordinate system. From Figures 8 (b), (c), and (d), the CD-CSA algorithm can better optimize the coverage positions of 36 nodes in the spatial coordinate system, making the nodes arranged more orderly. Compared to the CD-CSA algorithm, the CSA and PCA-CSA algorithms can optimize the position of some nodes in the spatial coordinate system, and the optimization effect is not as good as the CD-CSA algorithm.

Figure 9 shows the coverage changes under three coverage algorithms: CD-CSA, CSA, and PCA-CSA. Figure 9 (a) shows the coverage of nodes in three different algorithms as the number of iterations changes in a single experiment. Among them, the coverage of CD-CSA ranges from 0.92 to 0.97, the coverage of PCA-CSA ranges from 0.83 to 0.88, and the coverage of CSA ranges from 0.76 to 0.85. Figure 9 (b) shows the average coverage of nodes in three different algorithms as the number of iterations varies under multiple experiments. Among them, the average coverage of CD-CSA is around 0.96, CSA is around 0.83, and PCA-CSA is around 0.89.

TABLE 2. Parameters of coverage simulation experiment.

Parameter	Number
Experimental area	100(m^2)
Population size	40
Sensing radius	10(m)
Communication radius	20(m)
Iteration number	500
Learning rate	0.5
Step weights	0.1
Individual to population distance weight	0.9
Elimination probability	0.3
Cauchy's coefficient of variation strength	0.6

B. ROUTING OPTIMIZATION ALGORITHMS FOR WSNS PERFORMANCE ANALYSIS

To test the proposed DCRUC performance in wireless sensor routing optimization, the study compared DCRUC with the traditional Energy efficient Unequal Clustering Algorithm (EEUC) and immune-based Uneven Clustering (IBUC) proposed by others in a routing simulation environment. Table 3 shows the routing simulation experimental environment.

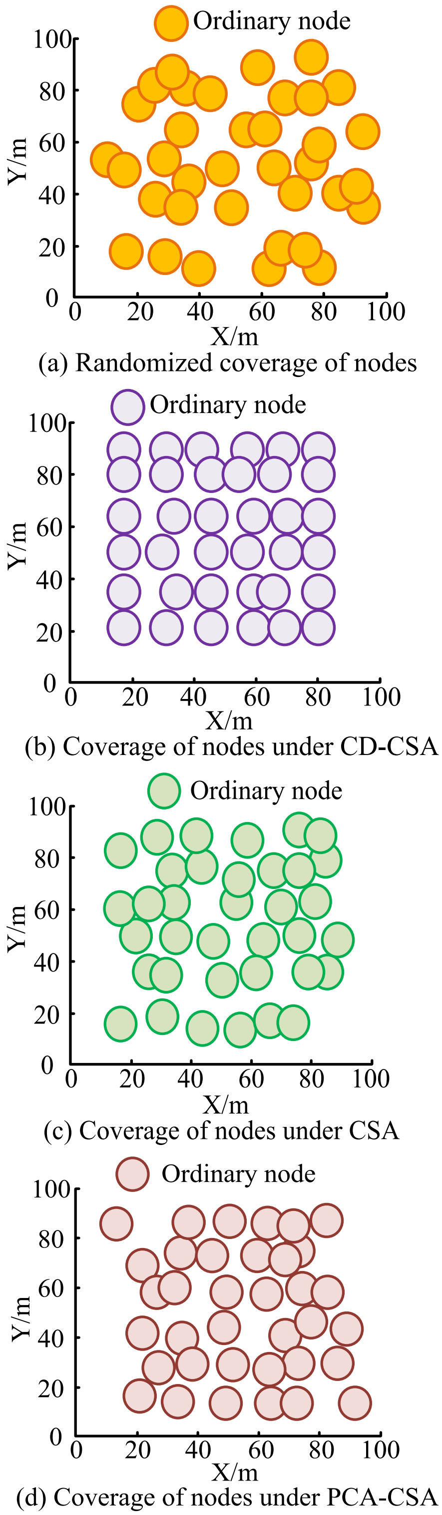

FIGURE 8. Random coverage of nodes with optimization.

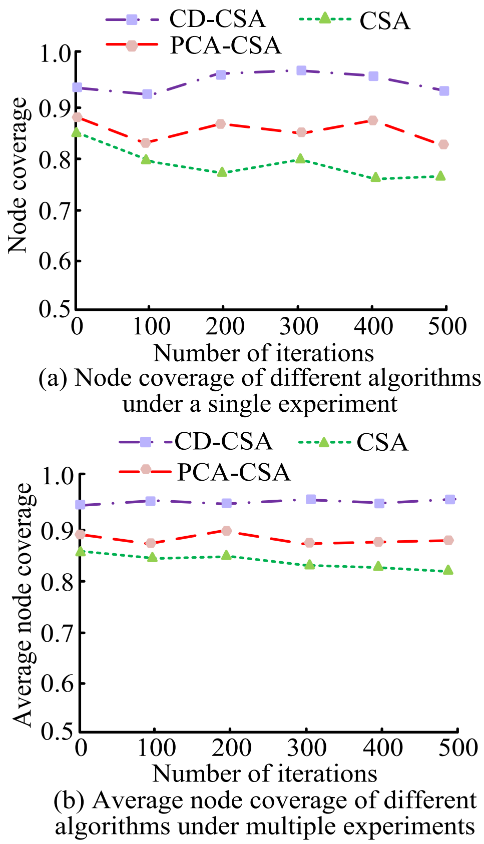

FIGURE 9. Coverage iterations for different coverage algorithms.

Table 3 provides the specific parameters of the routing simulation experimental environment. The parameter settings in Table 3 are designed to simulate large-scale WSNs. 200m^2 of experimental area and 400 nodes reflect a wider monitoring area and a larger number of nodes, while the packet size of 3000bit and the control message size of 80bit reflect the need for transmission of data and control information in a real network. Node initial energy of 0.5J was then used to model the energy limit of the nodes. These parameter settings help in evaluating the performance of the routing algorithm in real applications and ensure the realism and relevance of the simulation results. Under the parameter conditions in Table 3, the iterative performance of three routing algorithms, EEUC, DCRUC, and IBUC, was first compared. Then, the nodes' energy changes and the number of failed nodes of the three routing algorithms were compared in the same simulation environment. The iteration of the three routing algorithms is shown in Figure 10.

In Figure 10, the optimal fitness values of three routing algorithms, EEUC, DCRUC, and IBUC, are shown as a function of iteration. Among the three routing algorithms, EEUC needs to iterate 141 times, IBUC needs to iterate 109 times, and DCRUC can iterate as quickly as possible, only requiring 68 iterations for optimal fitness value. In summary, DCRUC has better stability during the iteration process.

In Figure 11, the energy changes and the number of failed nodes under the three routing algorithms of EEUC, DCRUC, and IBUC are shown. The "Energy consumption of cluster head nodes" in Figure 11(a) is in Joules (J) and it measures the energy consumption of cluster head nodes over a certain

TABLE 3. Routing simulation experiment parameters.

Parameter	Number
Experimental area	200(m^2)
Number of nodes	400
Packet size	3000(bit)
Control message size	80(bit)
Node initial energy	0.5(J)

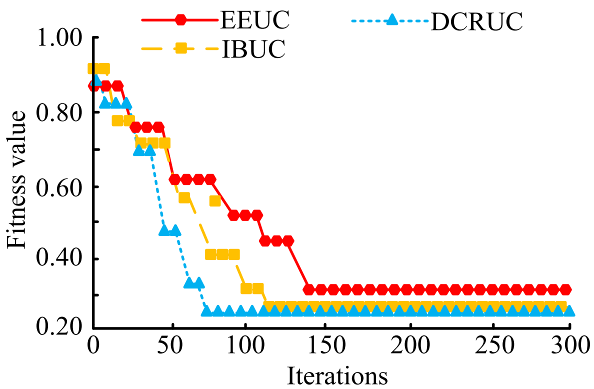

FIGURE 10. Iterations of different routing algorithms.

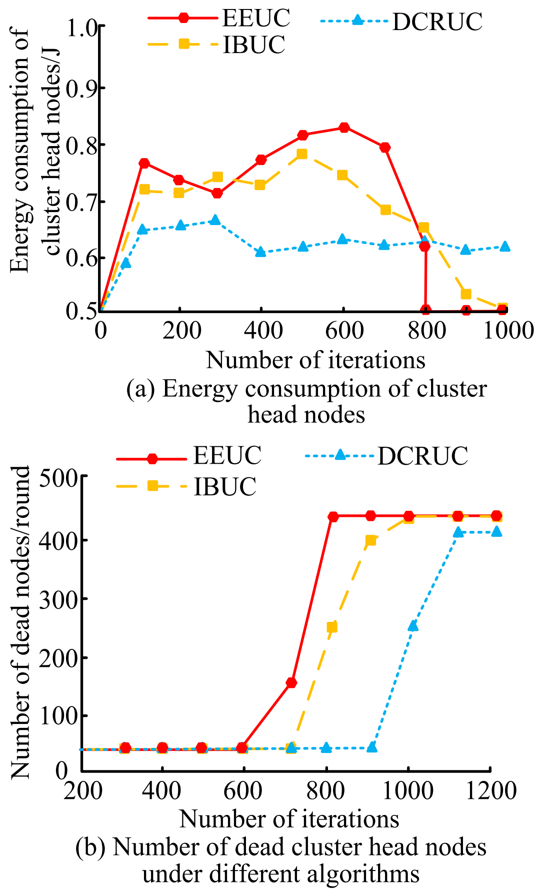

FIGURE 11. Iterations of different routing algorithms.

period. A low value of energy consumption of cluster head nodes indicated that the algorithm was more efficient in conserving energy, which was essential for extending the life-cycle of the WSNs. For example, in Figure 11(a), the minimum magnitude of cluster head node energy consumption under the DCRUC algorithm implies that the algorithm outperforms the EEUC and IBUC algorithms in terms of energy utilization efficiency. The "Number of dead nodes" in Figure 11(b) shows the number of nodes that fail in each iteration in terms of rounds. Early failure of nodes reduced the overall performance and coverage of the network, so this metric reflected the algorithm's ability to maintain network stability and functionality. For example, the DCRUC algorithm only started to increase the number of failed nodes at iteration 900, which indicated that it was more effective in maintaining the stability and continuous operation time of the network compared to other algorithms. Specifically, in Figure 11(a), the cluster head node energies under all three routing algorithms show an increasing and then decreasing trend as the number of iterations increases, but the cluster head node energies under the DCRUC optimization show the smallest change. When the number of iterations was 400, the cluster head node energy value under DCRUC stabilized around 0.63. In Figure 11(b), the number of failed nodes under all three routing algorithms increases as the number of iterations increases. Compared to EEUC and IBUC, DCRUC needed to iterate up to 900 generations before it started increasing the number of failed nodes.

In Figure 12, the energy calculation accuracy values of cluster head nodes under three routing algorithms are shown. As nodes increase, the accuracy values of the energy calculation for all three algorithms have changed. Compared to EEUC and IBUC, DCRUC has a calculation accuracy

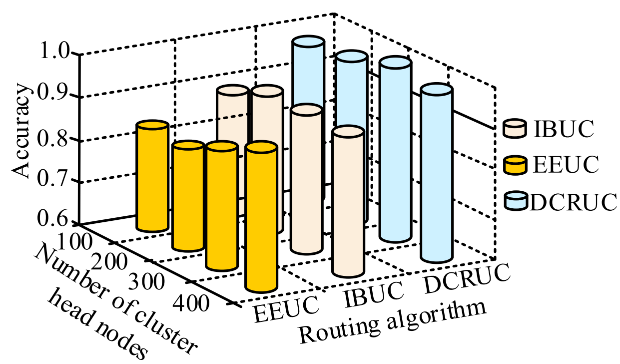

FIGURE 12. Accuracy of energy calculation for different routing algorithms.

of over 0.95 for the nodes' energy consumption value, thus being able to calculate the energy loss of nodes more accurately.

To demonstrate the good performance of the coverage algorithm and routing algorithm proposed in this study in practical applications, three types of WSNs were randomly selected for testing. The application effects of the algorithm proposed in the article and the optimization algorithm proposed by others in practical network problems are shown in Table 4.

TABLE 4. Effect of different algorithms in practice.

Network Problems	Algorithm type	Evaluation metrics	Network type 1	Network type 2	Network type 3
Node Coverage	**CD-CSA**	System runtime/min	1.25	**0.89**	1.57
Node Coverage	PCA-CSA	System runtime/min	3.10	1.85	3.26
Node Coverage	CSA	System runtime/min	5.12	2.02	5.54
Node Coverage	**CD-CSA**	Node coverage/%	98.65	96.78	**99.52**
Node Coverage	PCA-CSA	Node coverage/%	93.21	92.62	95.14
Node Coverage	CSA	Node coverage/%	87.32	85.65	86.23
Routing Optimization	**DCRUC**	System runtime/min	1.65	1.95	**1.47**
Routing Optimization	IBUC	System runtime/min	3.65	3.81	4.02
Routing Optimization	EEUC	System runtime/min	5.48	4.99	8.65
Routing Optimization	**DCRUC**	Energy loss rate/%	1.25	0.97	**0.84**
Routing Optimization	IBUC	Energy loss rate/%	5.62	7.26	3.21
Routing Optimization	EEUC	Energy loss rate/%	9.06	13.63	6.37

In Table 4, the actual application effects of different algorithms in three network problems under two scenarios of node coverage and routing optimization are presented. The accuracy of the energy calculation was measured by comparing the difference between the energy consumption values calculated by the algorithm and the actual energy consumption values. This is represented by a value between 0 and 1. The closer the value is to 1 the higher the accuracy of the calculation. For example, the DCRUC algorithm achieved an accuracy of more than 0.95 in calculating the energy consumption value of the cluster head node, which indicated that its predicted value was very close to the actual energy consumption, thus verifying the efficiency of the algorithm. The high accuracy of the energy calculation had a significant impact on the effectiveness of the algorithm as it ensured a more accurate prediction of the energy consumption, which helped to optimize the energy allocation and prolong the operation time of the network. Therefore, this metric directly affected the overall performance and stability of the WSNs.

In the node coverage problem, the running time of CD-CSA was 1.25 minutes, 0.89 minutes, and 1.57 minutes in Type 1, 2, and 3, respectively, with node coverage rates of 98.65, 96.78, and 99.52. The running time of PCA-CSA was 3.10 minutes, 1.85 minutes, and 3.26 minutes, respectively, with node coverage rates of 93.21, 92.62, and 95.14, respectively. The runtime of CSA was 5.12 minutes, 2.02 minutes, and 5.54 minutes, respectively, with node coverage rates of 87.32, 85.65, and 86.23.

In the routing optimization problem, the running time of DCRUC was 1.65 minutes, 1.95 minutes, and 1.47 minutes, respectively, with energy loss rates of 1.25, 0.97, and 0.84, respectively. The operating time of IBUC was 3.65 minutes, 3.81 minutes, and 4.02 minutes, respectively, with energy loss rates of 5.62, 7.26, and 3.21. The operating time of EEUC was 5.48 minutes, 4.99 minutes, and 8.65 minutes, respectively, with energy loss rates of 9.06 minutes, 13.63 minutes, and 6.37 minutes.

V. DISCUSSION

To prove that CD-CSA and DCRUC have better performance, the study introduces the metrics such as coverage, iteration case of fitness value, and the cluster head node energy calculation accuracy value, respectively, which are tested, and the corresponding data results are obtained. Figure 8 shows the random coverage of nodes and the coverage optimization under three coverage algorithms CD-CSA, CSA, and PCA-CSA. In the initial layout, 36 nodes are randomly distributed in the spatial coordinate system. The CD-CSA algorithm performs best in optimizing the node locations, resulting in a more orderly arrangement of nodes, which improves the coverage efficiency. In comparison, the CSA and PCA-CSA can only optimize some of the node locations, and their optimization is not as effective as the CD-CSA. Figure 9 shows the coverage changes under CD-CSA, CSA, and PCA-CSA. The results of a single experiment showed that the coverage ratio of CD-CSA ranged from 0.92 to 0.97, which was significantly better than that of PCA-CSA, which ranged from 0.83 to 0.88, and that of CSA, which ranged from 0.76 to 0.85. The average coverage ratios under multiple experiments showed a similar trend. These data indicated that the CD-CSA had a significant advantage in improving node coverage, which was crucial to ensuring effective monitoring of WSNs. In Figure 10, compared to EEUC and IBUC, DCRUC iterated to a steady state with only 68 iterations, which indicated that the algorithm had better stability. Figure 11(a) and Figure 11(b) show the change in the energy of cluster head nodes and the change in the number of failed nodes under different routing algorithms, respectively. The DCRUC algorithm performed the best in terms of cluster head node energy consumption and the number of failed nodes, which indicated that it had the advantage of maintaining network stability and conserving energy. In particular, the DCRUC algorithm showed a lower energy loss rate and later node failure trend with an increasing number of iterations, which reflected its effectiveness in network lifetime extension. In Figure 12, the cluster head node energy calculation accuracy values are given for the three routing algorithms. With the increase in the number of cluster head nodes, the accuracy of DCRUC for calculating the cluster head node energy consumption value was above 0.95, which indicated that the algorithm cancalculate the energy loss situation more accurately. Finally, to prove that both the coverage algorithm and the routing algorithm proposed in this study have better performance in practical applications, the study randomly selected three types of WSNs types for testing. As further shown by the results in Table 4, the CD-CSA algorithm performed well in terms of node coverage, especially in network type 3, where the coverage was as high as 99.52%, which was significantly higher than the other algorithms. This indicatedthat the CD-CSA algorithm washighly applicable and efficient in dense and high-demand network environments. In the node coverage problem, the CD-CSA algorithm significantly improved the coverage of the network by optimizing the layout of sensor nodes, which wascrucial for practical applications such as environmental monitoring or military reconnaissance scenarios. Second, the DCRUC algorithm showed higher efficiency in routing optimization. In the experiments, the algorithm outperformed other compared algorithms in terms of system runtime and node energy consumption. Especially in terms of energy efficiency, the DCRUC algorithm effectively reduced the energy loss and prolonged the lifetime of the network by optimizing the selection and adjustment of cluster head nodes.

In summary, this study verifies the effectiveness of CD-CSA and DCRUC algorithms in coverage and routing optimization for WSNs through detailed experiments and analysis. These findings not only enhance the theoretical understanding but also provide valuable references for the design and optimization of WSNs in practical applications.

VI. CONCLUSION

To improve the WSNs' node utilization and reduce system energy loss, this study combined the cuckoo algorithm and non-uniform clustering algorithm to design coverage optimization and routing optimization schemes for WSNs, respectively. The research results indicated that in algorithm performance testing, the average coverage of CD-CSA was around 0.96, much higher than the 0.83 of CSA and the 0.89 of PCA-CSA. DCRUC only needed to iterate 68 times to achieve stability. In addition, when iteration was 400, the cluster head node energy value under DCRUC remained stable at around 0.63. The calculation accuracy of DCRUC for the nodes' energy consumption value was also above 0.95, far higher than EEUC and IBUC. In practical applications, taking the node coverage problem as an example, compared to the PCA-CSA, CD-CSA had shorter system runtime and higher node coverage in various network types. Especially in Network Type 3, the node coverage of the CD-CSA was as high as 99.52%, far exceeding 95.14% of the PCA-CSA. In routing optimization problems, the DCRUC algorithm outperformed the IBUC algorithm in system runtime and energy loss rate. The minimum running time of the DCRUC algorithm in routing optimization problems was 1.47 minutes, and the minimum energy loss rate was 0.84%. In summary, the node coverage algorithm and routing optimization algorithm designed by this research institute have good performance and application performance. However, due to the complex structure of WSNs, there are still certain limitations in studying only their coverage and routing issues. Therefore, the scope of research can be further expanded in the future.

REFERENCES

[1] R. M. A. Ikram, A. A. Dehrashid, B. Zhang, Z. Chen, B. N. Le, and H. Moayedi, "A novel swarm intelligence: Cuckoo optimization algorithm (COA) and SailFish optimizer (SFO) in landslide susceptibility assessment," *Stochastic Environ. Res. Risk Assessment*, vol. 37, no. 5, pp. 1717–1743, Jan. 2023, doi: 10.1007/s00477-022-02361-5.

[2] H. Su, D. Zhao, F. Yu, A. A. Heidari, Z. Xu, F. S. Alotaibi, M. Mafarja, and H. Chen, "A horizontal and vertical crossover cuckoo search: Optimizing performance for the engineering problems," *J. Comput. Design Eng.*, vol. 10, no. 1, pp. 36–64, Jan. 2023, doi: 10.1093/jcde/qwac112.

[3] M. Braik, A. Sheta, H. Al-Hiary, and S. Aljahdali, "Enhanced cuckoo search algorithm for industrial winding process modeling," *J. Intell. Manuf.*, vol. 34, no. 4, pp. 1911–1940, Apr. 2023, doi: 10.1007/s10845-021-01900-1.

[4] Y.-D. Yao, X. Li, Y.-P. Cui, L. Deng, and C. Wang, "Game theory and coverage optimization based multihop routing protocol for network lifetime in wireless sensor networks," *IEEE Sensors J.*, vol. 22, no. 13, pp. 13739–13752, Jul. 2022, doi: 10.1109/JSEN.2022.3178441.

[5] S. Juneja, K. Kaur, and H. Singh, "An intelligent coverage optimization and link-stability routing for energy efficient wireless sensor network," *Wireless Netw.*, vol. 28, no. 2, pp. 705–719, Jan. 2022, doi: 10.1007/s11276-021-02818-5.

[6] S. T. Shishavan and F. S. Gharehchopogh, "An improved cuckoo search optimization algorithm with genetic algorithm for community detection in complex networks," *Multimedia Tools Appl.*, vol. 81, no. 18, pp. 25205–25231, Mar. 2022, doi: 10.1007/s11042-022-12409-x.

[7] D. Toresa, F. Wiza, A. A. Irwanda, W. S. Abiyus, E. Edriyansyah, and T. Taslim, "The cuckoo optimization algorithm enhanced visualization of morphological features of diabetic retinopathy," *J. Appl. Eng. Technol. Sci.*, vol. 4, no. 2, pp. 929–939, Jun. 2023, doi: 10.37385/jaets.v4i2.1978.

[8] H. Zavieh, A. Javadpour, Y. Li, F. Ja'fari, S. H. Nasseri, and A. S. Rostami, "Task processing optimization using cuckoo particle swarm (CPS) algorithm in cloud computing infrastructure," *Cluster Comput.*, vol. 26, no. 1, pp. 745–769, Feb. 2023, doi: 10.1007/s10586-022-03796-9.

[9] G. Tong, S. Zhang, W. Wang, and G. Yang, "A particle swarm optimization routing scheme for wireless sensor networks," *CCF Trans. Pervasive Comput. Interact.*, vol. 5, no. 2, pp. 125–138, Jun. 2023, doi: 10.1007/s42486-022-00118-1.

[10] S. M. U. Sankar, M. S. Christo, and P. S. U. Priyadarsini, "Secure and energy concise route revamp technique in wireless sensor networks," *Intell. Autom. Soft Comput.*, vol. 35, no. 2, pp. 2337–2351, Jun. 2023, doi: 10.32604/iasc.2023.030278.

[11] G. A. Senthil, A. Raaza, and N. Kumar, "Internet of Things energy efficient cluster-based routing using hybrid particle swarm optimization for wireless sensor network," *Wireless Pers. Commun.*, vol. 122, no. 3, pp. 2603–2619, Feb. 2022, doi: 10.1007/s11277-021-09015-9.

[12] B. Joshi and M. K. Thakur, "Genetic algorithm- and cuckoo search algorithm-based routing optimizations in network-on-chip," *Arabian J. Sci. Eng.*, vol. 48, no. 8, pp. 9635–9644, Aug. 2023, doi: 10.1007/s13369-022-07272-9.

[13] K. Thirugnanasambandam, U. Prabu, D. Saravanan, D. K. Anguraj, and R. S. Raghav, "Fortified cuckoo search algorithm on training multi-layer perceptron for solving classification problems," *Pers. Ubiquitous Comput.*, vol. 27, no. 3, pp. 1039–1049, Mar. 2023, doi: 10.1007/s00779-023-01716-1.

[14] X. Yang, X. Hao, T. Yang, Y. Li, Y. Zhang, and J. Wang, "Elite-guided multi-objective cuckoo search algorithm based on crossover operation and information enhancement," *Soft Comput.*, vol. 27, no. 8, pp. 4761–4778, Apr. 2023, doi: 10.1007/s00500-022-07605-8.

[15] C. Liu, J. Wang, L. Zhou, and A. Rezaeipanah, "Solving the multi-objective problem of IoT service placement in fog computing using cuckoo search algorithm," *Neural Process. Lett.*, vol. 54, no. 3, pp. 1823–1854, Jan. 2022, doi: 10.1007/s11063-021-10708-2.

[16] V. B. Christopher, R. I. Sajan, T. S. Akhila, and M. J. Kavitha, "A QoS aware three way point rule based fusion of Earth worm and deer hunt optimization routing in wireless sensor network," *Wireless Pers. Commun.*, vol. 128, no. 2, pp. 1193–1215, Jan. 2023, doi: 10.1007/s11277-022-09995-2.

[17] K. Saxena, N. Gupta, J. Gupta, D. K. Sharma, and K. Dev, "Trajectory optimization for the UAV assisted data collection in wireless sensor networks," *Wireless Netw.*, vol. 28, no. 4, pp. 1785–1796, Mar. 2022, doi: 10.1007/s11276-022-02934-w.

[18] S. K. Barnwal, A. Prakash, and D. K. Yadav, "Improved African buffalo optimization-based energy efficient clustering wireless sensor networks using metaheuristic routing technique," *Wireless Pers. Commun.*, vol. 130, no. 3, pp. 1575–1596, Apr. 2023, doi: 10.1007/s11277-023-10345-z.

[19] S. Gudla and R. Kuda, "A reliable routing mechanism with energy-efficient node selection for data transmission using a genetic algorithm in wireless sensor network," *Facta Universitatis Ser., Electron. Energetics*, vol. 36, no. 2, pp. 209–226, Jan. 2023, doi: 10.2298/fuee2302209g.

[20] Y. Guo, Z. Mustafaoglu, and D. Koundal, "Spam detection using bidirectional transformers and machine learning classifier algorithms," *J. Comput. Cogn. Eng.*, vol. 2, no. 1, pp. 5–9, Feb. 2023, doi: 10.47852/bonviewjcce2202192.

[21] R. Yarinezhad and S. N. Hashemi, "A sensor deployment approach for target coverage problem in wireless sensor networks," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 5, pp. 5941–5956, May 2023, doi: 10.1007/s12652-020-02195-5.

[22] S. Nematzadeh, M. Torkamanian-Afshar, A. Seyyedabbasi, and F. Kiani, "Maximizing coverage and maintaining connectivity in WSN and decentralized IoT: An efficient metaheuristic-based method for environment-aware node deployment," *Neural Comput. Appl.*, vol. 35, no. 1, pp. 611–641, Jan. 2023, doi: 10.1007/s00521-022-07786-1.

[23] L. Chen, Y. Xu, F. Xu, Q. Hu, and Z. Tang, "Balancing the trade-off between cost and reliability for wireless sensor networks: A multi-objective optimized deployment method," *Int. J. Speech Technol.*, vol. 53, no. 8, pp. 9148–9173, Apr. 2023, doi: 10.1007/s10489-022-03875-9.

[24] D. K. Sah, S. Srivastava, R. Kumar, and T. Amgoth, "An energy efficient coverage aware algorithm in energy harvesting wireless sensor networks," *Wireless Netw.*, vol. 29, no. 3, pp. 1175–1195, Apr. 2023, doi: 10.1007/s11276-022-03125-3.

JIAN YANG was born in Anhui, China, in 1982. He received the B.S. and M.S. degrees in biomedical engineering from Northeastern University, China, in 2003 and 2006, respectively, and the Ph.D. degree in communication and information systems from Sun Yat-sen University, China, in 2009.

From 2009 to 2019, he was a Lecturer with the School of Automation, Guangdong University of Technology, Guangdong, China, where he has been an Associate Professor, since 2020. He is the author of one book and more than 20 articles. His current research interests include intelligent information processing, RFID anti-collision, and wireless sensor networking.

YIMIN XIA was born in Hubei, China, in 1980. She received the B.S. degree in electronic information engineering and the M.S. degree in communication and information systems from China University of Geosciences, Hubei, in 2001 and 2004, respectively, and the Ph.D. degree in control theory and control engineering from Guangdong University of Technology, Guangdong, China, in 2011.

From 2004 to 2006, she was a Teaching Assistant with the School of Automation, Guangdong University of Technology, where she was a Lecturer, from 2007 to 2021. Since 2022, she has been an Associate Professor with the School of Integrated Circuits, Guangdong University of Technology. She is the author of one book and more than 20 articles. Her research interests include wireless sensor networking, mobile robot navigation, AI algorithms, and integrated circuit EDA design.