Received September 30, 2017, accepted December 12, 2017, date of publication December 15, 2017,
date of current version February 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2783969

A Sidelobe and Energy Optimization Array Node Selection Algorithm for Collaborative Beamforming in Wireless Sensor Networks

GENG SUN[1,2], (Student Member, IEEE), YANHENG LIU[1], SHUANG LIANG[1,3], ZHAOYU CHEN[1], AIMIN WANG[1], QIANAO JU[2], (Student Member, IEEE), AND YING ZHANG[2], (Senior Member, IEEE)

[1]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[3]Department of Information Technology, Changchun Vocational Institute of Technology, Changchun 130033, China

Corresponding authors: Aimin Wang (wang_ai_min@126.com) and Ying Zhang (yzhang@gatech.edu)

This work was supported in part by the National Natural Science Foundation of China under Grant 61373123, in part by the Chinese Scholarship Council under Grant [2016] 3100, and in part by the Graduate Innovation Fund of Jilin University under Grant 2017016.

ABSTRACT The beam pattern of the virtual node antenna array (VNAA) in wireless sensor networks has a high maximum sidelobe level (SLL), thereby causing communication interference because of the uncontrollable node positions. A sidelobe and energy optimization array node selection (SEOANS) algorithm is proposed for optimally selecting the sensor nodes to form a VNAA that optimizes the beam pattern of VNAA and reduces the average energy consumption of nodes. SEOANS uses a calculation method to determine the optimal number of array nodes, proposes a node location selection optimization method based on concentric circular ring array and a novel swarm intelligence optimization algorithm called cuckoo search chicken swarm optimization (CSCSO) to optimize the excitation current of each array node. CSCSO uses chaos theory, introduces the inertia weight Lévy flight, and adopts the grade mechanism in chicken swarm optimization to improve the performance of the cuckoo search algorithm. In addition, the scheduling and fault tolerance mechanisms are designed and implemented in SEOANS. Simulation results show that the node position selection optimization method and the excitation current optimization based on CSCSO can effectively reduce the maximum SLL. Furthermore, compared with traditional clustering routing algorithms, SEOANS has advantages in the communication delay and average energy consumption of nodes, thus effectively improving network lifetime.

INDEX TERMS Wireless sensor networks, random antenna arrays, cuckoo algorithm, energy efficient, beamforming.

I. INTRODUCTION

Wireless sensor networks (WSNs) are usually composed of a large number of energy-constrained sensor nodes that are randomly distributed in a wide monitoring area to collect, perceive, and integrate data [1]. Given the restriction on node size, economic cost, calculation ability, and energy supplement infeasibility, the sensing range and communication range of a single sensor node are limited [2]. Therefore, sensor nodes have to communicate with the base station (BS) by self-organization and by a multi-hop wireless communication network. However, WSNs are often deployed in inaccessible outdoor environments with large geographical spaces and network sizes at long distances, thereby causing high communication delay [3]. Multi-hop communication also undoubtedly leads to high routing overhead and improves the communication failure probability. These consequences are unacceptable for scenarios that require high real-time performance [4]. The constrained energy of nodes is also one of the most notable features of WSNs. In some applications, sensor nodes are deployed by unmanned aerial vehicles (UAVs) into extreme environments that cannot be easily reached by humans [5], thus leading to coverage holes in some parts of the monitoring area if some nodes run out of energy. This phenomenon not only causes lack of perceptual data integrity but also disconnects the network in the event of node death, thus forming an information-isolated island [6]. To solve

this problem, the energy efficiency of the nodes must be improved.

Many traditional clustering routing algorithms have been proposed to solve the energy optimization problem. The low-energy adaptive clustering hierarchy (LEACH) algorithm [7] uses circular random clustering methods, and each node has the opportunity to become a cluster head in rotation. However, LEACH assumes that each node has sufficient power to communicate with the BS via a single hop, thereby causing a huge energy consumption with the increase of the distance between the cluster head and the BS. The hybrid energy-efficient distributed (HEED) [8] clustering approach inherits the clustering idea of LEACH and further considers the energy constraint in calculating the probability of selecting a node as the cluster head. HEED, however, cannot guarantee the balance of the energy consumption of the cluster heads and this will lead to the premature death of these cluster head nodes. The LEACH expected residual energy (LEACH-ERE) algorithm [9], which is based on LEACH, uses fuzzy logic theory to predict the residual energy and extend the network lifetime. Energy-efficient LEACH (EE-LEACH) [10] is also an energy-efficient routing algorithm based on LEACH. In this algorithm, the cluster head with the maximum residual energy can provide service routings for multiple clusters. The algorithm can also prolong the network lifetime by balancing the energy consumption of the cluster heads. However, both LEACH-ERE and EE-LEACH assume that the cluster head can communicate with the BS directly, and this assumption is inapplicable to large-scale WSNs because a single sensor node generally does not have a high transmission power. The cross unequal clustering routing algorithm (CUCRA) [11] is a multi-hop communication algorithm based on the idea of competition radius, which is affected by energy factors. The competing radius decreases with the decreasing remaining energy of nodes. Although CUCRA is more practical than LEACH, CUCRA may lead to an excessive number of cluster heads and result in a long transmission delay.

Collaborative beamforming (CB) is an effective technique for improving the energy efficiency of WSNs [12]. Using CB, nodes with limited transmission power can form a virtual array antenna that transmits a high-gain and high-directivity beam to achieve direct communication with the BS without multi-hop. CB can not only reduce communication delay but also obtain the energy gain and prolong the lifetime of a network [13]. While synchronization and information sharing have to be overcome to use CB in WSNs [14], the deterioration of the beam pattern caused by the node position error is usually considered the main problem of CB in WSNs [15]. In WSNs, sensor nodes are usually deployed randomly in the monitoring area. Although the mainlobe of the node array cannot be affected by the node position error if the number of array nodes is sufficiently large, the sidelobe performance is closely related to the position error [16]. A lower sidelobe indicates a lower interference at other directions. In addition, a large number of array nodes will result in an increase of the average energy consumption of nodes [17], which is not desired in WSNs. Consequently, the influence of node position error on both the mainlobe and the sidelobe must be overcome, and the maximum sidelobe level (SLL) of the beam pattern must be reduced. Therefore, how to select the optimal array nodes is a key issue for CB in WSNs.

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

(1) We propose a method for determining the number of array nodes to achieve the optimal energy efficiency and propose an node location selection method for random distributed sensor nodes based on concentric circular antenna array (CCAA).
(2) We propose a new swarm intelligence optimization algorithm called cuckoo search chicken swarm optimization (CSCSO) to further optimize the excitation current of the selected array nodes. CSCSO uses chaos theory to optimize the initial solutions, introduces a inertia weight Lévy flight, and borrows a hierarchical mechanism from chicken swarm optimization (CSO) to increase the utilization rate of the population in cuckoo search (CS) algorithm and to improve the convergence rate of the algorithm, thereby making it suitable for sensor nodes.
(3) We propose an optimization method called sidelobe and energy optimization array node selection (SEOANS) algorithm to select the array sensor nodes for CB in WSNs with joint optimization of beam pattern and energy consumption. The scheduling and fault tolerance mechanisms of SEOANS are designed, and the overhead of SEOANS is analyzed.
(4) The effectiveness and performance in terms of the beam patterns, the convergence rates, the energy consumptions and the communication delay of SEOANS are verified by simulations.

The rest of the paper is organized as follows. Section II shows the related work. Section III describes the system models used in the study. Section IV presents a new method called SEOANS algorithm. Section V demonstrates performance of the proposed algorithm using simulation. Finally, Section VI presents the conclusion of the study and the future research directions.

II. RELATED WORKS

The combination of different node positions has a significant effect on the mainlobe as well as the sidelobe of the beam pattern of the VNAA. To optimize the beam patterns of CB, reference [18] proposes an SLL suppression mechanism based on a node selection algorithm to reduce the maximum SLL of the CB array. The algorithm selects some BS communication test nodes to form a set of candidate nodes on the basis of the stochastic heuristic random method. If the interference-to-noise ratio of a certain set of nodes is lower than a preset threshold, then the array nodes must be re-selected. However, the BS communication test time in the algorithm increases exponentially with the increasing number of BSs in the network. Inspired by the work in [18], Chen *et al.* [19] proposes

a node selection algorithm to suppress the maximum SLL based on cross-entropy optimization (CEO). The algorithm utilizes CEO to determine the candidate node set, thus reducing the complex calculation of the source node and the BS communication test times. However, similar to the work of [18], the CEO-based algorithm can only optimize the SLL in the unintended directions of BSs but cannot effectively suppress the SLL in other directions. Furthermore, the two algorithms do not consider the reasonable number of array nodes and energy efficiency, or the effect of the array shape and array element excitation current on the sidelobe performance. Malik *et al.* [20] propose a node selection method based on a virtual linear antenna array. The algorithm reduces the maximum SLL in a given range. However. the optimal number of array nodes and the node selection scheduling mechanism are not mentioned. Malik *et al.* [21] further propose a node selection method based on virtual circular antenna array with optimized distances between the array nodes. The optimized beam patterns are also compared with the beam patterns without any optimization. However, this algorithm does not consider the optimization of node position and node excitation current simultaneously. Therefore, it cannot achieve the best optimization effect. Moreover, the proposed algorithm does not provide a detailed node scheduling mechanism. In addition, the aforementioned algorithms only consider the optimization of the beam pattern while the optimization of energy is not considered at the same time.

In this paper, we proposed a joint beam pattern and energy consumption optimization method to select CB nodes in WSNs.

III. SYSTEM MODELS

A. NETWORK MODEL

Several reasonable assumptions should be considered to facilitate the discussion. The monitoring area is considered as $A_{monitor}$ and it is covered by randomly distributed isomorphic sensor nodes. The nodes do not move after deployment. Furthermore, the network has the same characteristics as that in [18], [19], and [22], and the main assumptions are shown as follows.

(1) We only consider the uplink beamforming because the downlink beamforming from the BS does not have many challenges.

(2) For simplicity, it is assumed that the BS and the sensor nodes are located on the same plane in an open environment. Thus, the channel characteristics vary very slowly with time so that the channel could be considered to be time-invariant, and hence we can assume the channel conditions are good for DCB in outdoor environment. In addition, It has been demonstrated by [23] that the signal-to-noise ratio (SNR) and the capacity of the receiver will be improved if the maximum SLL of the transmitter is reduced by using DCB. Thus, in this paper, we only consider to reduce the SLL of the transmitter and do not consider specific channel model.

(3) Each node has the location information of itself and the other nodes in the CB cluster. This can be obtained by many strategies shown in [24].

(4) The sensor nodes are synchronized in terms of the carrier frequency, the time and the initial phase by using some synchronization algorithms presented in [24].

B. RANDOM NODE ANTENNA ARRAY MODEL

Assume the position of the target BS in the spherical coordinate system is (A, φ_0, θ_0). In general, we set $\varphi_0 = 0$. Hence, the array factor (AF) of the virtual node antenna array (VNAA) can be approximated as follows [25]:

$$AF(\phi, \theta, \omega) = \sum_{k=1}^{N} \omega_k e^{j\frac{2\pi}{\lambda} r_k [\sin(\theta_0)\cos(\phi_0 - \psi_k) - \sin(\theta)\cos(\phi - \psi_k)]} \tag{1}$$

where N is the number of the array node, λ is the wavelength, ω_k is the excitation current of the kth node, r_k and ψ_k are the polar coordinates of the kth node. ψ_k is expressed as follows:

$$\psi_k = -\frac{2\pi}{\lambda} d_k (\phi_0, \theta_0) \tag{2}$$

where d_k is the Euclidean distance between the kth node and the BS.

C. ENERGY MODEL

In this paper, the widely used transmission energy consumption model of WSNs is adopted [1]:

$$E_T = bE_{elec} + b\varepsilon_{fs} d^2 \tag{3}$$

where E_T is the transmit energy consumption, E_{elec} is the electronic energy that depends on factors such as digital coding, b is the bit number, ε_{fs} denotes the amplifier energy that depends on the required receiver sensitivity and receiver noise figure, respectively. d is the distance between the two nodes. Moreover, the receiving energy consumption E_R is defined as:

$$E_R = bE_{elec} \tag{4}$$

IV. SEOANS ALGORITHM

Different from the beamforming technology based on the real antenna array, the virtual CB in WSNs faces a series of problems, such as constrained energy, uncertain node position, and high SLL caused by the position errors. Therefore, we propose an effective array node selection algorithm SEOANS to solve the aforementioned problems. SEOANS determines the optimal number of the array nodes first on the basis of the distance between the source node and the BS to minimize the total energy consumption. Thereafter, SEOANS uses a node location selection optimization method and an optimal excitation current optimization method based on swarm intelligence optimization to obtain the optimal performance of the mainlobe and minimize the maximum SLL. Finally, the energy efficient node selection mechanism are proposed in SEOANS.

A. OPTIMAL NUMBER OF THE ARRAY NODES FOR MINIMIZING ENERGY CONSUMPTION

Assume that the distance between a sensor node in the network and BS is d_t. According to Eq. (3), the energy consumption of a node to transmit k bit data to the BS is expressed as follows:

$$E_{total} = kE_{elec} + kE_{transmit} \tag{5}$$

where $E_{transmit}$ is the energy consumption of wireless communication. According to the principle of electromagnetic wave superposition, the transmit power of each node in a virtual array with N nodes is reduced to $E_{transmit}/N^2$ by using CB [26]. However, CB via sensor nodes will undoubtedly increase the energy consumption during the scheduling stage. The total energy consumption of this stage mainly consists of two parts: the source node broadcasts the selection message to be received by N candidate nodes and each candidate node responses by providing the feedback message to the source node. The energy consumption of the first part is expressed as follows:

$$E_{manage\_T} = \left(kE_{elec} + k\varepsilon_{fs}d_{Rc}^2\right) + N \cdot kE_{elec} \tag{6}$$

where d_{Rc} is the communication range of the source node. The energy consumption of the second part is expressed as follows:

$$E_{manage\_R} = kE_{elec} + \left(NkE_{elec} + \sum_{i=1}^{N} k\varepsilon_{fs}d_{cluster_i}^2\right) \tag{7}$$

where $d_{cluster}$ is the distance between the source node and the candidate node. The source node does not participate in the CB transmission in our method, and it is only responsible for selecting and scheduling the candidate node. Therefore, the energy consumption of the wireless communication utilizing CB is:

$$E_{beam} = N\left(kE_{elec} + k\varepsilon_{fs}\frac{d_t^2}{N^2}\right) \tag{8}$$

Thus, the total energy consumption of the scheduling stage and CB stage is as follows:

$$\begin{aligned} E_{beam\_total} &= E_{manage\_T} + E_{manage\_R} + E_{beam} \\ &= \left(kE_{elec} + k\varepsilon_{fs}d_{Rc}^2\right) + N \cdot kE_{elec} \\ &\quad + kE_{elec} + (NkE_{elec} + \sum_{i=1}^{N} k\varepsilon_{fs}d_{cluster_i}^2) \\ &\quad + N\left(kE_{elec} + k\varepsilon_{fs}\frac{d_t^2}{N^2}\right) \end{aligned} \tag{9}$$

In general, $d_{cluster} \leqslant d_t$, Eq. (9) can be simplified to the following form:

$$E_{beam\_total} = (3N+2)kE_{elec} + (N+1)k\varepsilon_{fs}d_{R_c}^2 + k\varepsilon_{fs}\frac{d_t^2}{N} \tag{10}$$

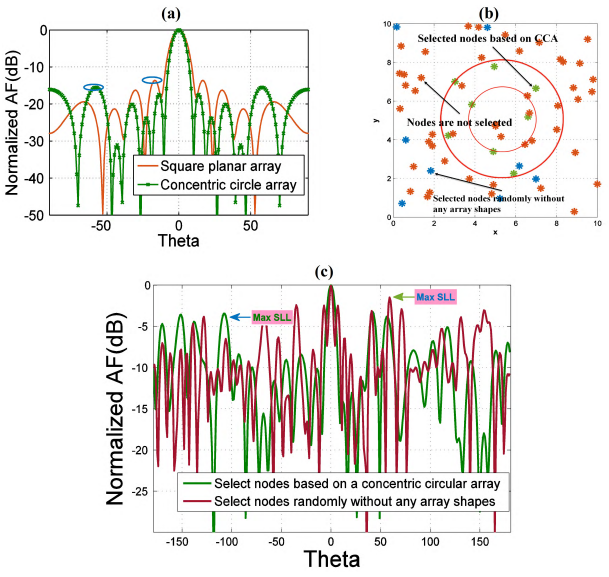


FIGURE 1. (a) Two-dimensional beam patterns of the CCAA and PAA. (b) Location optimization for random array antenna. (c) Beam patterns obtained with and without node location optimization.

The overall energy consumption of a certain d_t with and without CB should be equal, thus $E_{total} = E_{beam\_total}$ is considered for the preceding equation, and N is the unknown quantity. The optimal value for N can be obtained by using differentiation.

$$N_{energy\_best} = \sqrt{\frac{\varepsilon_{fs}}{3E_{elec} + \varepsilon_{fs} \cdot d_{R_c}^2}} \cdot d_t \tag{11}$$

B. ARRAY NODE LOCATION SELECTION OPTIMIZATION

The spacing between the array nodes affects the beam pattern of the antenna array [27]. The coupling between array elements will lead to the increase of SLL if the distance between the array elements is less than 2.5 λ [28]. Given the position errors in the random element antenna array, the maximum SLL of the plane array with desired fixed element positions is lower than that of the antenna array with random elements if the number of array elements is the same. This is due to the coupling between the array elements if all nodes are randomly selected. Selecting array nodes randomly also leads to a range of active nodes, whereas the other ranges are idle. Thus, the distribution of array nodes becomes irregular, and the SLL deteriorates further.

The array nodes must be optimally selected to solve the aforementioned problems. Since the beam pattern performance of the antenna array with a desired fixed shape is better than that of the antenna array with randomly deployed elements when the numbers of the array elements are the same, we can choose a fixed shape as an idealized antenna array model to provide references in selecting the randomly distributed array nodes.

Fig. 1(a) shows the beam patterns of the concentric circular antenna arrays (CCAA) and the planar antenna arrays (PAA). As can be seen, the maximum SLL of CCAA is lower than that of PAA, whereas the mainlobe widths are similar.

Moreover, Fig. 1(b) shows that 64 sensor nodes are randomly distributed in the 10 m × 10 m area, wherein the green nodes are selected according to the CCAA and the blue nodes are selected randomly. Fig. 1(c) shows the beam patterns of the two types of antenna arrays, and the comparison indicates that the maximum SLL of the array selected on the basis of CCAA is significantly lower than that of the array selected randomly while both arrays have similar mainlobe performance. Furthermore, the effective communication range of a sensor node often covers a circular area. Thus, using CCAA can naturally adapt the selection and scheduling mechanism between the source node and the array node. By considering all of these factors, it is reasonable to optimize the array node location selection optimization according to the CCAA. The details of the selection process are described in Section IV D.

C. EXCITATION CURRENT OPTIMIZATION

After selecting the optimal number of nodes and their corresponding locations, the excitation current of each array node should be further optimized. The fitness functions and the proposed optimizer are presented in this section.

1) FITNESS FUNCTIONS

The excitation amplitude ω_n affects the mainlobe and the maximum SLL of the beam pattern. Therefore, we can construct the fitness function based on Eq. (1). The goal is to form a random sensor array antenna with minimum SLL. Thus, the fitness function can be designed as follows [25]:

$$Fitness_{SL} = 20\log_{10}\left|\frac{\max|AF(\phi_{SL})|}{AF(\phi_{ML})}\right| \tag{12}$$

where ϕ_{SL} is the angle of the sidelobe during the optimization process, and ϕ_{ML} is the angle of the mainlobe. The positions of ϕ_{SL} can be determined by finding all the peak points of the AF. Thus, the optimization problem aims to determine the suitable sets of the node excitation current ω_n that minimizing the fitness function $Fitness_{SL}$ in Eq. (12).

Moreover, to jointly optimize the maximum SLL and the nulls, another fitness function which needs to be minimized in this study is consistent with that in [29]. The formula is as follows:

$$Fitness_{SL\_NULL} = \sum_i \frac{1}{\Delta\phi_i}\int_{\phi_{li}}^{\phi_{ui}} |AF(\phi_k)|^2\, d_\phi + \sum_k |AF(\phi_k)|^2 \tag{13}$$

where $[\phi_{li}, \phi_{ui}]$ is the suppression range of SLL, and $\Delta\phi_i = \phi_{ui} - \phi_{li}$ and ϕ_k are the nulls of the beam pattern.

2) CSCSO

Beam pattern synthesis is a complex non-linear problem, and swarm intelligence optimization methods, such as particle swarm optimization, have been widely used to optimize antenna array beam pattern [30]. The beam pattern optimization for virtual antenna array in WSNs is often executed on low-cost sensor nodes. Although the development of electronic devices and energy storage technology results in significant improvement of the computing ability, information storage capacity, and energy storage capacity of a single node and the energy consumption of calculation can be negligible compared with that of communication, running a complex swarm intelligence optimization algorithm on a sensor node still brings additional delay overhead. Therefore, the convergence speed of the algorithm must be further improved so that the algorithm can find the optimal solution in a short period. Thus, we propose a novel algorithm called CSCSO that combines CS [31] and CSO [32] to improve the performance. In the proposed algorithm, by introducing the hierarchical mechanism of CSO algorithm, the utilization of the populations in CS is improved. Moreover, we introduce an initial solution optimization mechanism based on chaos theory [33] and weight coefficient to improve the quality of the initial population and the convergence speed of CSCSO, respectively.

(a) CS: The CS algorithm is based on three ideal rules [31]. The solution in CS should be updated by using the Lévy flight mechanism. Short-distance search and occasional long-distance walking appear alternately, such that the Lévy flight mechanism can expand the search scope and increase population diversity. The updated formula is expressed as follows:

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{Lévy}(\lambda) \tag{14}$$

where x_i^t represents the location of the ith nest at generation t. α is the step factor. For most applications, its value is 1. The random step value of Lévy flight is taken from the Lévy distribution:

$$Levy \sim u = t^{-\lambda}(1 < \lambda < 3) \tag{15}$$

(b) CSO: The CSO algorithm is proposed for the optimization problems by imitating the hierarchy and crowd behavior of chickens [32] and it is based on four basic rules presented in [32]. The CSO divides the chicken swarm into several groups, and the identity of chickens depends on the fitness values. The chickens with the best fitness value are determined as roosters, whereas the chickens with the worst fitness value are identified as chicks. The rest of the chickens are identified as hens. Hens can randomly select a group for survival, and the motherchild relationship between the hens and the chicks is determined randomly. In the group, each chicken can be considered as a solution, whereas a moved chicken is a new solution. The optimal solution is retained in the end, which is the ultimate goal of this algorithm.

According to the hierarchy of chickens, the updating formula of different grades of individuals can be defined. The updating method of the rooster is expressed as follows:

$$x_{i,j}^{t+1} = x_{i,j}^t * (1 + Randn(0, \sigma^2)) \tag{16}$$

$$\sigma^2 = \begin{cases} 1, & \text{if } f_i \le f_k \\ \exp(\frac{(f_k - f_i)}{|f_i| + \varepsilon}), & \text{otherwise} \end{cases} \quad k \in [1, N],\ k \ne i \tag{17}$$

where $Randn\ (0,\sigma^2)$ is a Gaussian distribution function with a mean of zero and a standard deviation σ^2. f is the fitness value of the corresponding x, and k represents the serial number of the rooster. The rooster has greater priority than the other chicken in foraging food in the same group.

All hens follow the roosters to forage in the same group. However, dominant hens have more advantages than the others. This condition can be formulated as follows:

$$x_{i,j}^{t+1} = x_{i,j}^{t} + S_1 * Rand * (x_{r1,j}^{t} - x_{i,j}^{t}) + S_2 * Rand * (x_{r2,j}^{t} - x_{i,j}^{t}) \quad (18)$$

$$S_1 = \exp\frac{f_i - f_{ri}}{(abs(f_i) + \varepsilon)} \quad (19)$$

$$S_2 = \exp(f_{r2} - f_{r1}) \quad (20)$$

where $Rand$ is a random number between [0, 1]. $r1$ represents the index of the rooster ($r1 \in [1,N]$), and $r2$ represents the index ($r2 \in [1,N]$]) of the hens in the same group, $r1 \neq r2$.

Chicks can only forage around their mother hens, and this situation can be formulated as follows:

$$x_{i,j}^{t+1} = x_{i,j}^{t} + FL * (x_{m,j}^{t} - x_{i,j}^{t}) \quad (21)$$

where $x_{m,j}^{t}$ represents the ith hen. The adjusted parameter FL used in Eq. (21) represents the individual differences of the chicks.

The information of the solutions flows in a single direction, and the entire optimization process leads to the optimal solution.

(c) Population initialization: The initialization of the population is often random in the swarm intelligence optimization algorithms. The optimal population positions have certain blindness, which affects the convergence rate of the algorithm. Thus, we use chaos theory to improve the performance of initial solution and to accelerate the convergence rate by optimizing the initial solutions.

Logistic mapping model has been widely used recently. It has a simple form and has better ergodicity than other chaotic models. Logistic mapping model can be described as follows [33]:

$$Z_{k+1} = 4Z_k(1 - Z_k), \quad Z_k \in (0, 1) \quad (22)$$

The initial solution can be evenly distributed in the population space by chaos theory. Eq. (23) can map the variable range of chaos theory to the variable range of the optimization problem.

$$x_i = L_b + (U_b - L_b)Z_i \quad (23)$$

(d) Weight coefficient: In normal CS algorithm, cuckoos search the path of parasitized nests randomly, thereby reducing the accuracy of the solution. We introduce a weight coefficient factor w to improve the performance of the algorithm. Thus, Eq. (14) can be rewritten as follows:

$$x_i^{t+1} = \omega_{coe} \times x_i^{t} + \alpha \oplus \text{Lévy}(\lambda) \quad (24)$$

The weight coefficient can extend the search space of the algorithm. The global search ability of the algorithm is improved if w_{coe} is large, whereas the local search ability is enhanced if w_{coe} is relatively small. The weight coefficient factor is related to the number of rounds of the same optimal value in the optimization process. If the number of the rounds of the same optimal value is small, the global optimal solution may be located in the vicinity of the range. Thus, the value of w_{coe} can be small. By contrast, if the number of the rounds of the same optimal value gradually increases, the value of w_{coe} should increase correspondingly, so that the algorithm can jump out of the local optimal value. Thus, w_{coe} can be defined as follows:

$$\omega_{ceo} = a + (b - a)\exp(\frac{m}{K})^{0.5} \quad (25)$$

where a and b are the coefficients, m is the record for number of times the same optimal value appears and its value is 1 when the optimal solution appears for the first time, and K is a constant. The values of a, b and K are determined by the parameter tuning tests in Section V.

(e) Population hierarchy mechanism: CS has a strong global search capability, but its utilization rate of populations is low, thus resulting in an increase in the number of iteration and a deceleration of the convergence rate. The convergence rate of CS can be improved by introducing the grade mechanism of CSO. The solutions in the population of CSO are divided into three levels, and different levels of the solutions should use a different location update mechanism. If the solution is the rooster, then the update step of this solution is small. By contrast, if one solution is the chick, then the update step of the solution is large. In this way, the poor solutions in the population are utilized fully, which can effectively improve the utilization rate of population. Therefore, we introduce the grade mechanism of CSO into CS to improve its performance.

(f) Steps of CSCSO: The pseudocode of CSCSO algorithm is shown in Algorithm 1.

D. ARRAY NODE SELECTION AND OPTIMIZATION PROCESS

In this section, the array node selection and optimization process for CB in WSN is presented. Assume all nodes are randomly deployed in $A_{monitor}$ in the initialization phase of the network, and the sensor nodes are constantly aware of information from the environment. The nodes only perceive data but do not communicate with each other. When a node produces abnormal data that must be to transmitted to the BS, the node becomes the source node S_{node}. S_{node} must select a group of collaborating nodes to form a VNAA so that CB can be used for direct communication with the BS. The candidate node set in the communication range R_c of S_{node} can be defined as $C_{candidate}$, and S_{node} should select an optimal group of sensor nodes to form a virtual antenna array. The selected array nodes are defined as C_{beam} and $C_{beam} \in C_{candidate}$.

The steps of the array node selection and optimization process are as follows:

Step 1: S_{node} confirms and selects a BS to communicate. According to the network model, the network has a number

Algorithm 1 Framework of CSCSO

```
Initialize a population of N host nests via chaos theory
  and define the relation parameters;
Evaluate the N nests fitness value, t = 0;
while t <MaxGeneration do
    Update the nest location by taking Lévy flights from
    random nest;
    If the new solution is better than the previous one
    update it;
    if r > p_a then
        The location nest is changed randomly;
        If the new solution is better than the previous
        one update it;
    end
    if t mod G = 1 then
        Rank the nests fitness value and establish a
        hierarchal order in the swarm;
        Divide the swarm into different groups and
        determine the relationship;
        for i = 1 to N do
            if i = nest-> rooster then
                Update its solution by (16);
            end
            if i = nest-> hen then
                Update its solution by (18);
            end
            if i = nest-> chick then
                Update its solution (21);
            end
            Evaluate the new solution;
            If the new solution is better than its previous
            one update it;
        end
    end
end
```

of BSs. Therefore, S_{node} needs to communicate with one of the BSs according to the current network status.

Step 2: S_{node} calculates the optimal number of the array nodes N_{best}. S_{node} first calculates the distance d_t between itself and the BS:

$$d_t = \sqrt{(x_{base} - x_{s\_node})^2 + (y_{base} - y_{s\_node})^2} \tag{26}$$

where x_{base}, y_{base}, x_{s_node}, and y_{s_node} are the coordinates of BS and S_{node}. Then S_{node} should determine the optimal number N_{energy_best} of the array nodes according to Eq. (11). To ensure that the power is sufficient to communicate directly with the BS, the minimum number of the array nodes is:

$$N_{min} = \sqrt{\frac{E_{beam\_total}}{E_{node\_max}}} \tag{27}$$

where E_{node_max} is the energy consumption of a node while transmitting one-bit data to the BS using its maximum transmit power, and E_{beam_total} is the total energy consumption for CB. Thus, $N_{best} = \max(N_{min}, N_{energy\_best})$. In some cases, N_{min} may be larger than N_{energy_best}. Hence, the energy consumption for the scheduling stage has to be improved. However, it can guarantee that the node array has sufficient total power to communicate with the BS directly.

Step 3: Determine azimuth θ and φ. S_{node} determines θ and φ of the VNAA according to the coordinates of the BS and itself. These two angles can determine the transmission direction of the VNAA.

Step 4: Select the array nodes with optimal locations. Since the SLL generated by the mutual coupling increases if the space between the array element spacing is less than 2.5 λ, we will set the radius of the innermost circle of the CCAA to be 2.5 λ and the radius of each layer of the ring is increased by 2.5 λ. According to these constraints, the ideal number of array nodes on each layer of the CCAA is expressed as follows:

$$N_{circle\_i} = \frac{C_{circle\_i}}{S_i} = \frac{2\pi R_i}{2R_i \arcsin\frac{L}{2R_i}} = \frac{\pi}{\arcsin\frac{L}{2R_i}} \tag{28}$$

where C_{circle_i} is the circumference of the ith ring, S_i is the arc length between the two nodes on the i_{th} ring, R_i is the radius of the ith ring and L is the distance between the two nodes on the ith ring. Thus, the ideal number of array nodes in the communication radius of S_{node} is:

$$N_{circle\_most} = \sum_{i=1}^{M} N_{circle\_i} = \sum_{i=1}^{M} \frac{\pi}{\arcsin\frac{L}{2R_i}} \tag{29}$$

where M is the number of rings of the CCAA within the communication radius of S_{node}. Moreover, N_{circle_most} should large or equal to N_{best}, and if it cannot find out the whole integral number of the nodes on a ring, the mechanism should select parts of the nodes on the innermost ring as much as possible. Based on the above analysis, the specific steps of selecting the array nodes with optimal locations are presented as follows:

Step 4.1: S_{node} calculates the coordinates of the optimal locations of the array nodes based on the CCAA and stores these coordinates in the optimal node location set $L_{opt}(Node_i)$. This set can be recorded as the location of the optimal node list $List_{opt}$. The nodes in L_{opt} should satisfy the limiting conditions described by Eqs. (30) and (31):

$$\sqrt{(x_{R\_i} - x_{s\_node})^2 + (y_{R\_i} - y_{s\_node})^2} = R_i \tag{30}$$

$$\sqrt{(x_{R\_i} - x_{R\_i+1})^2 + (y_{R\_i} - y_{R\_i+1})^2} = 2.5\lambda \tag{31}$$

where x_{R_i} and y_{R_i} are the coordinates of the ith ideal node in the concentric circular array with radius of R_i. Given that the calculation process is relatively simple, the details are not shown.

Step 4.2: S_{node} broadcasts the selection message M_{select} of the array node to $C_{candidate}$. The message will notify the candidate nodes to reply to S_{node} with their ID and coordinates.

Step 4.3: The nodes in $C_{candidate}$ receives M_{select} and send their ID and coordinates to S_{node}.

Step 4.4: S_{node} receives the coordinates of the nodes from $C_{candidate}$ and creates a candidate node list $List_{candidate}$. S_{node} needs to compare $List_{candidate}$ and $List_{opt}$ and calculate the distance between each node in $List_{opt}$ and each node in $List_{candidate}$. The nodes in $List_{candidate}$ that are closest to the optimal locations in $List_{opt}$ will be selected as the array nodes. The selected array node set is C_{beam}, and the number of the nodes in C_{beam} is N_{best}. The order of node selection process should start from the inner layer to the outer layer centered at S_{node}, which will ensure that the S_{node} does not jump over a layer to select nodes. Selecting nodes that jump over layers will increase the communication interference and energy consumption.

If the number of the nodes in $List_{opt}$ is M_{opt} and the number of the nodes in $List_{candidate}$ is $N_{candidate}$, the time to calculate the distance between each node in the two sets is $M_{opt}N_{candidate}$ using the exhaustive method. This step not only causes the S_{node} to spend more energy for local computing, but also increases communication delay. To solve this problem, we propose an improved method: First, S_{node} calculates the distance between each node in $List_{candidate}$ and itself, which requires $N_{candidate}$ times calculation. Second, assuming that we use a CCAA with four rings and divide the nodes in $List_{opt}$ into four groups based on the radius. Assume the number of nodes in each ring equals $N_{candidate}/4$ to facilitate calculation and without loss of generality. In the same way, the nodes in $List_{candidate}$ are divided into four groups according to the distance from the S_{node}, and the same assumption that the number of the nodes in each group is equal is held, which is $M_{opt}/4$. Finally, S_{node} can calculate the distance between the two corresponding groups. With the above method, the number of calculation required for this step is $(N_{candidate}+M_{opt}N_{candidate}/4)$, which is significantly less than $M_{opt}N_{candidate}$, and computation energy consumption and communication delay are reduced. Moreover, this method can also be used to ensure that the S_{node} could select array nodes from the inner layer to the outer layer more accurately.

Step 5: S_{node} uses the CSCSO algorithm to calculate optimal excitation current ω_n of each array node. S_{node} takes the coordinates of the array nodes into AF and calculates the fitness function to obtain N_{best} optimal excitation currents of the array nodes.

Step 6: S_{node} broadcasts the CB control message $M_{beam\_control}$. The content of $M_{beam\_control}$ includes the following:

(1) CB management message M_{manage}, which mainly contains the *ID* of the array nodes. This message is used to inform the node in $C_{candidate}$, which is selected as the array node and added to C_{beam}.
(2) The residual energy message M_{manage} indicating the remaining energy of S_{node}.
(3) Optimal excitation currents message M_{excit}, which mainly includes the excitation currents of the array nodes.
(4) Data message M_{data} that needs to be transmitted to the BS.
(5) Time synchronization control message T_{sync}, which is used to synchronize the transmission time of the array nodes.
(6) Random time delay control message T_{rand_delay}.

Step 7: The nodes in C_{beam} should respond to S_{node} after receiving $M_{beam\_control}$ in T_{rand_delay}. The response message includes the residual energy information of each array node and other status information. If S_{node} receives the response messages from all of the array nodes successfully, the node in C_{beam} will send M_{data} to the BS through CB according to the appointed time of T_{sync} and the optimal excitation current M_{excit}. If S_{node} does not receive the response message from C_{beam} during T_{rand_delay}, $M_{beam\_control}$ should be broadcasted again until the S_{node} receives response messages from all the nodes. To ensure that S_{node} has sufficient time to broadcast the control information again when the failure occurs, T_{sync} needs to be longer than T_{rand_delay}.

At this point, the array node selection and optimization process is completed. According to these steps, the entire algorithm mainly consists of two parts, namely, the array node selection stage and CB stage. After these two stages, the nodes in C_{beam} will be retained in a period. If $Node_i$ in $C_{candidate}$ needs to transmit messages to the BS in the following rounds, it can first send the data to S_{node} for data fusion. S_{node} can then transmit the data generated by $Node_i$ to the BS by utilizing CB. The array nodes do not need to be calculated and selected each time, thus saving energy. By using the array node maintaining strategy, only steps 6 and 7 of the process need to be used when a node needs to transmit data to the BS. This strategy can further reduce energy consumption and the communication delay of the nodes.

Furthermore, the conditions for re-running all steps of SEOANS need to be proposed. With the operation of the network, the sensor nodes will inevitably exhaust its energy and die and the array node selection algorithm needs to be run again. According to the description of the algorithm, S_{node} exchanges the residual energy information with the nodes in C_{beam}. Thus, S_{node} and C_{beam} know the residual energy of each other. If a node in C_{beam} will run out of energy (we set it to be less than 15% of the maximum energy), SEOANS needs to be re-run and form a new VNAA.

E. OVERHEAD ANALYSIS OF SEOANS

In this section, the performances of the proposed SEOANS, including the overhead of energy consumption, the overhead of communication delay, and the fault tolerance, will be analyzed.

1) OVERHEAD OF ENERGY CONSUMPTION

The energy consumption of the SEOANS algorithm mainly consists of two major categories: the energy consumption of the node selection stage and the energy consumption of CB. The following stages 1 to 4 show the energy consumption of the node selection stage, and stage 5 indicates the energy consumption of CB.

Stage 1: S_{node} broadcasts the node selection message M_{select}, and its energy consumption is expressed as follows:

$$E_{overh1} = E_{T\_source} + N_{candi} \cdot E_{R\_candi} \tag{32}$$

Stage 2: The nodes in $C_{candidate}$ receives M_{select} and respond M_{resp} to S_{node}. The energy consumption of this part is expressed as follows:

$$E_{overh2} = N_{candi} \cdot E_{T\_candi} + E_{R\_source} \tag{33}$$

Stage 3: S_{node} broadcasts CB control message $M_{beam\_control}$ to C_{beam}. The energy consumption of this part is:

$$E_{overh3} = E_{T\_source} + N_{beam} \cdot E_{R\_beam} \tag{34}$$

Stage 4: C_{beam} receives $M_{beam\_control}$ and sends response message M_{beam_resp} to S_{node}. The energy consumption of this part is expressed as follows:

$$E_{overh4} = N_{beam} \cdot E_{T\_beam} + E_{R\_source} \tag{35}$$

Stage 5: C_{beam} utilized CB according to T_{sync}. The consumption of this part is:

$$E_{overh5} = N_{beam} \cdot E_{beamform} \tag{36}$$

By defining ρ as the ratio of energy consumption of the selection stage and the total energy consumption, we can assume the distance between the array nodes and the source is the communication range R_c to simplify the calculation and without losing generality. Thus, ρ can be described in Eq. (37), as shown at the bottom of the next page.

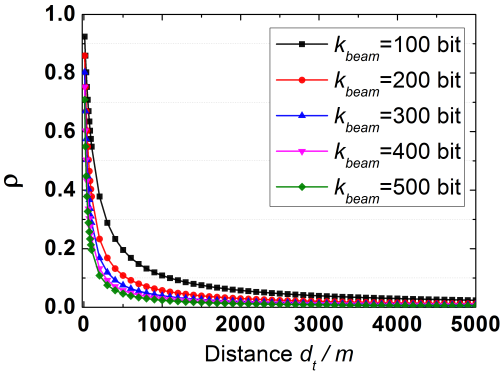


FIGURE 2. Energy overhead in the node selection stage.

where N_{candi} is the number of the candidate nodes in $C_{candidate}$, N_{beam} is the number of nodes in C_{beam}. k_s is the length of the control message, and k_{beam} is the length of the data transmitted by CB. Fig. 2 shows the change of ρ with d_t. Here, $k_s = 20$ bit, k_{beam} is from 100 to 500 bit, $R_c = 10$ m. The figure indicates that when the BS is far away from a certain monitoring node array, the control overhead of the total energy consumption is small. With increasing d_t, the control overhead can even be disregarded. Conversely, when the distance between the monitoring node array and the BS is close, the control overhead is a large proportion of the total energy consumption. Hence, beamforming is no longer applicable.

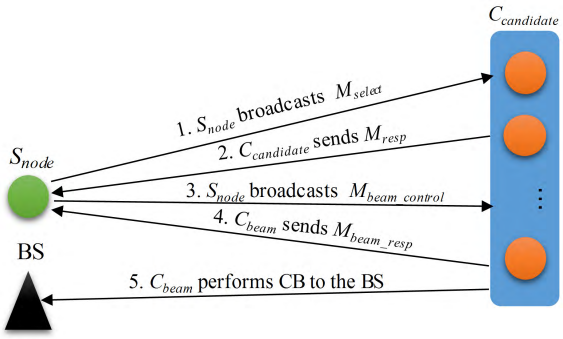


FIGURE 3. Communication delay of the node selection stage.

2) OVERHEAD OF COMMUNICATION DELAY

The communication delay of SEOANS also includes five parts and it is shown in Fig. 3. Corresponding to the energy consumption stages, in the first four stages, S_{node} needs to communicate with the nodes in $C_{candidate}$ and C_{beam} twice that generates the node selection communication delay, and in the fifth stage, the nodes in C_{beam} can communicate with the BS by one hop, which can reduce the communication delay effectively. This indicates that if S_{node} is close to the BS, utilizing CB will increase the communication delay due to the delay at the node selection stage. Therefore, SEOANS is not suitable for small-scale networks. Instead, SEOANS is substantially suitable for long-distance communication. This conclusion will be further validated using simulation in Section V.

F. FAULTY TOLERANT MECHANISM

1) NODE FAULTY

The two main failures of the array nodes are node faulty and asynchronous error. In Step 7 of SEOANS, if S_{node} does not receive the response message from an array node after broadcasting $M_{beam\_control}$ three times, the array node is considered to be in the node faulty state. Node faulty will reduce the number of array nodes, which causes the VNAA not to have sufficient power to communicate with the BS directly and result in communication failure. In addition, the node faulty will also cause some sensing blind areas of the network. SEOANS has a fault tolerance mechanism under the node faulty conditions. In the CB stage, S_{node} broadcasts the parameters of the array nodes and requests these nodes to send back the response message. If S_{node} cannot receive the response message from $Node_k$ in the array more than three times, this node will be removed from C_{beam}. S_{node} will select another node that is closest to $Node_k$ and add it to C_{beam}. $M_{beam\_control}$ will be broadcasted again. Although the alternate node selected by the fault tolerance mechanism is not optimal, the mechanism can ensure the effective use of CB under the node faulty condition and only slightly reduce the SLL performance. The repeated running of Steps 1-5 of SEOANS is avoided, thus improving communication effectiveness.

2) ASYNCHRONOUS ERROR

Furthermore, T_{sync} is introduced in the algorithm, which can ensure synchronization of the array nodes in the CB stage. However, asynchronous error of the array nodes is possible in a practical situation because of internal timer error and other reasons. Since the number of array nodes is not reduced, the VNAA with the synchronization error should have sufficient power to communicate with the BS and the reliability of the information transmission should not be affected. However, this error will affect data rate and may cause higher SLL and wider mainlobe of the beam pattern. To solve the problem of synchronization error, the array nodes should send the actual CB time of the last CB stage back to S_{node} in Step 7 of SEOANS. If S_{node} determines that the actual transmit time of one array node is different from T_{sync} three times, this node can be considered the synchronization error node with inner timer faulty and it will be removed from C_{beam}. The follow-up step is consistent with the fault tolerance mechanism under the node faulty condition.

V. SIMULATION RESULTS AND ANALYSIS

In this section, we will simulate and verify the performance of the proposed SEOANS algorithm based on the Matlab platform. The simulation scenarios include single BS and multiple BSs. The SLL suppression of the beam pattern and the nulls of the beam pattern are shown in our simulations. Moreover, the communication delay and energy consumption of SEOANS are compared with that of other clustering and routing algorithms. The basic simulation parameters are shown in Table 1.

TABLE 1. Basic parameters of the simulations.

Parameter	Definition	Values
M_{Length}	Side length of the square area of the monitoring area	1000 m
N_{total_lagre}	Number of all nodes in the monitoring area	100000
S_{node_close}	Coordinate of source node in the near monitoring area	(77.66, 55.21)
S_{node_remote}	Coordinate of source node in the far monitoring area	(967.27,899.04)
L_{base}	Coordinate of the intended BS	(0, 0)
$E_{initial}$	Initial energy of the node	2 J
E_{elec}	Energy consumption of the transmitting circuit	50 nJ/bit
ε_{fs}	Energy consumption coefficient	10 pJ/bit/m2
d_{t_close}	Distance between the source node and BS in near area	95.28 m
d_{t_remote}	Distance between the source node and BS in far area	1320.34 m
R_{c_close}	Communication radius of the node in the near area	10 m
R_{c_remote}	Communication radius of the node in the far area	20 m

To facilitate the presentation and description of the node selection results of the SEOANS algorithm, we first select a sensor node S_{node_close}, which is close to the BS as the source node. According to the parameters in Table 2, the distance between S_{node_close} and the BS is 95.28 m, the optimal number of the array nodes is 8.57, as calculated using Eq. (11), and the minimum number of array nods is 9.30, as calculated using Eq. (27). Based on the mechanism of the algorithm, S_{node_close} will select 10 nodes to form a VNAA to ensure that the node array has sufficient power to directly communicate with the BS. Fig. 4 shows the node selection schematic diagram after S_{node_close} runs the SEOANS algorithm. The figure indicates that BS-0 is the intended BS, whereas BS-1 and BS-2 are the unintended BSs, which are the interfere BSs. The red dots are the candidate nodes in communication radius R_c of S_{node_close}, and these nodes correspond to $C_{candidate}$ in SEOANS. The bigger blue solid dots are the ideal locations of CCAA, the red arrows point to actual sensor nodes C_{beam} according to the ideal locations of the array nodes.

TABLE 2. Statistical results of different algorithms for 10-node case.

Algorithm	CS	BBO	CSCSO
Worst	-7.08	-6.48	-7.23
Best	-7.73	-7.39	-7.80
Median	-7.18	-6.64	-7.23
Mean	-7.26	-6.69	-7.39
SD	0.18	0.20	0.15

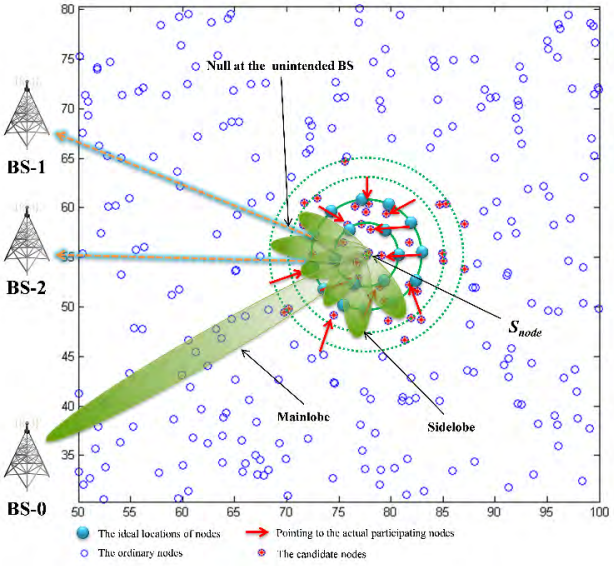


FIGURE 4. Node selection optimization in WSNs with the random distribution of nodes.

A. RADIATION BEAM PATTERN

To verify the SLL suppression performance and the nulls performance, the simulation scenes will be divided into

$$\begin{aligned}\rho &= \frac{E_{select\_stage}}{E_{select\_stage}+E_{beam\_stage}} \\ &= \frac{E_{overh1}+E_{overh2}+E_{overh3}+E_{overh4}}{E_{overh1}+E_{overh2}+E_{overh3}+E_{overh4}+E_{overh5}} \\ &= \frac{k_s E_{elec}(2N_{candi}+2N_{beam}+4)+k_s\varepsilon_{fs}R_c^2(N_{candi}+N_{beam}+2)}{[k_s E_{elec}(2N_{candi}+2N_{beam}+4)+k_s\varepsilon_{fs}R_c^2(N_{candi}+N_{beam}+2)]+k_{beam}N_{beam}E_{elec}+N_{beam}(\frac{k_{beam}\varepsilon_{fs}d_t^2}{N_{beam}})}\end{aligned} \tag{37}$$

single BS and multi-BS environments for discussion, respectively.

1) PARAMETER TUNING

Different parameters of a heuristic algorithm will result in different optimization performance for the same problem. Moreover, for different optimization problems, the optimal parameter selections of a heuristic algorithm will also be different. Thus, it is necessary to tune the parameters of the algorithm to achieve the best performance. In this paper, we refer to the parameter tuning method in reference [25] to tune the key parameters a, b and G in CSCSO. The tuning tests are conducted for 50 times repeatedly and the average results are presented.

The restrictions of a and b are [0, 2] and the tuning results are shown in Fig. 5(a). It can be seen from the figure that the best values are $a = 0.4$ and $b = 0.9$. Moreover, we use the these values of a and b to tune G (varied from 1 to 10) and Fig. 5(b) shows that the optimal value of G is 5 for our optimization problem.

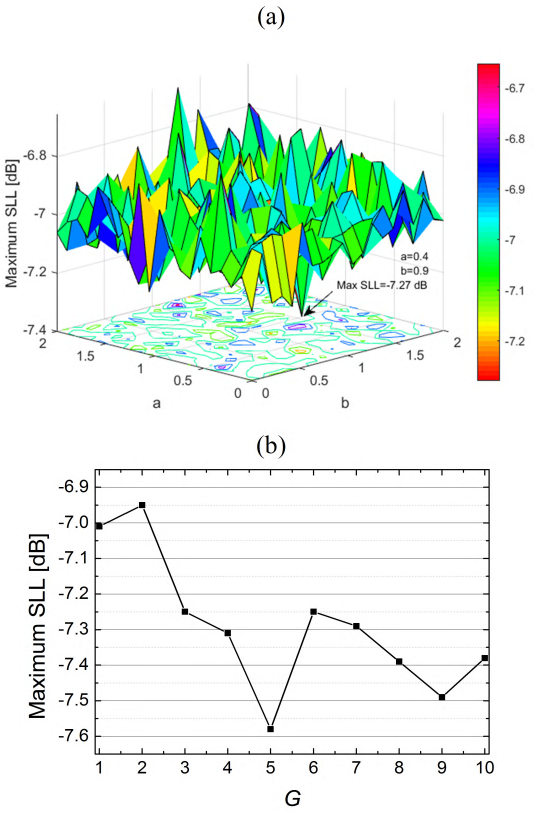


FIGURE 5. Key parameter tuning of CSCSO.

2) SINGLE BS SCENARIO

In this scenario, only one BS exists in the network, and the location of the BS is shown in Table 1. Moreover, the beam pattern obtained by SEOANS is compared with that of the randomly selected array nodes and the array nodes with optimal locations but without excitation current optimization. To verify the performance of the proposed CSCSO algorithm, we optimize the excitation currents using CS, biogeography-based optimization (BBO) [34], and CSCSO based on the location optimized array and compare the results obtained by these different swarm intelligence algorithms. These simulations are repeated for 50 times independently and we select one result randomly from these tests to show intuitively the beam patterns.

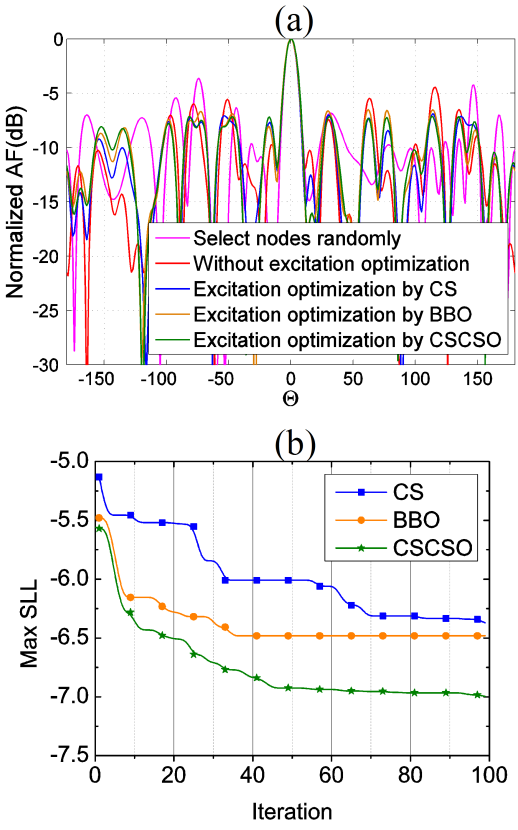


FIGURE 6. Radiation beampattern optimization for the 10 array nodes obtained by CSCSO-based SEOANS. (a) Beampatterns. (b) Convergence rates.

Fig. 6(a) shows the beam patterns of the selected 10-element VNAA, and these beam patterns are relative to the node selection schematic diagram in Fig. 4. The maximum SLL of the beam pattern that was obtained by randomly selected nodes is -3.62 dB, and the maximum SLL obtained by location optimized method is -4.44 dB. After excitation current optimization using CS, BBO, and CSCSO based on array nodes with optimized locations, the maximum SLL is -7.08, -6.48, and -7.22 dB, respectively. The figure indicates that the maximum SLL of the location-optimized node array is significantly lower than the randomly selected node array. The maximum SLL was further reduced after excitation current optimization by employing swam intelligence algorithms. Note that there is a trade-off between the mainlobe and the sidelobe of the beam pattern of an antenna array, i.e. if the sidelobe is low, the interference will be reduced and the energy of the signal will focus on the mainlobe. Therefore,

this method can be easily adapted to other channel conditions to improve the received SNR by effectively reducing the sidelobe of the transmitter [23].

Fig. 6(b) shows the convergence rates of the excitation current optimization process using CS, BBO, and the proposed CSCSO algorithm. The figure indicates that the solution of CSCSO exhibits higher accuracy and faster convergence rate because CSCSO includes chaos-based initial solution mechanism and the initial solutions of the algorithm are distributed in more favorable positions. Moreover, CSCSO can determine the steps of the Lévy flight adaptively based on the time when the same optimal solution appears and prevent the algorithm from getting stuck in the local optimum by utilizing the weight coefficient in the Lévy flight mechanism. In addition, the CSCSO algorithm adopts the hierarchical mechanism and allows the poor solutions to transform into better ones, which improve the utilization ratio of the populations, and further improve the convergence rate. Improving the accuracy of the solution and convergence rate is very important for sensor nodes with limited energy and computation ability. Thus, CSCSO is more suitable for WSNs compared with other algorithms.

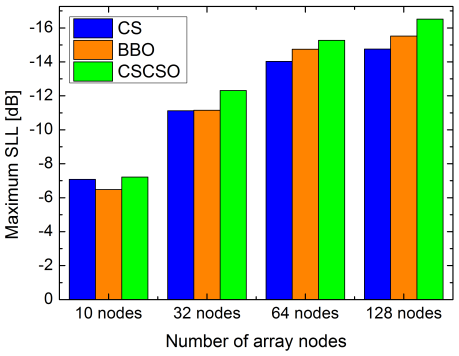


FIGURE 7. Maximum SLLs versus different numbers of nodes obtained by different algorithms.

Furthermore, to evaluate the performance of the node selection method for different network sizes, and to verify the performance of the proposed CSCSO algorithm for solving the excitation current optimization problems with different solution dimensions, the 32, 64, and 128 node VNAA are simulated and the maximum SLLs of these conditions are shown in Fig. 7. It can be seen from the figure that the proposed CSCSO algorithm achieves the lowest maximum SLL in all cases. Moreover, the numerical statistical results of these cases are shown in Tables 2, 3, 4, and 5, respectively. As can be seen, the proposed CSCSO algorithm has the best performance in terms of the accuracy and stability.

Moreover, the CCDF of a beam pattern is the probability that the average power at a direction exceeds threshold power P_0 [25], such that the CCDF is expressed as follows:

$$Pr[P(\theta) > P_0] = Pr[20\log_{10}(\frac{AF(\theta)}{AF(\theta_{ML})}) > P_0] \quad (38)$$

where $\theta \in [-\pi, \pi]$. Figs. 8(a), 8(b), 8(c), and 8(d) indicate the CCDF of 10 elements, 32 elements, 64 elements, and

TABLE 3. Statistical results of different algorithms for 32-node case.

Algorithm	CS	BBO	CSCSO
Worst	-11.12	-11.15	-12.31
Best	-11.96	-12.01	-12.86
Median	-11.32	-11.34	-12.45
Mean	-11.37	-11.42	-12.47
SD	0.23	0.23	0.12

TABLE 4. Statistical results of different algorithms for 64-node case.

Algorithm	CS	BBO	CSCSO
Worst	-14.04	-14.76	-15.27
Best	-14.79	-15.50	-15.83
Median	-14.32	-15.03	-15.35
Mean	-14.33	-15.03	-15.42
SD	0.22	0.22	0.15

TABLE 5. Statistical results of different algorithms for 128-node case.

Algorithm	CS	BBO	CSCSO
Worst	-14.76	-15.52	-16.52
Best	-15.66	-16.29	-17.04
Median	-14.81	-15.76	-16.62
Mean	-14.98	-15.77	-16.67
SD	0.18	0.21	0.14

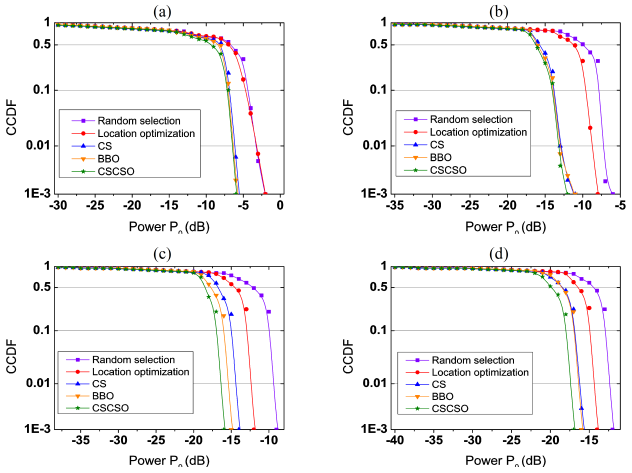


FIGURE 8. CCDF of the beampatterns for different algorithms. (a) 10 nodes. (b) 32 nodes. (c) 64 nodes. (d) 128 nodes.

128 elements. The figure shows that the CCDF of the beampattern obtained by CSCSO has significant advantages than other algorithms.

3) MULTI-BS SCENARIO

In this scenario, the multiple BSs in the network are assumed to be consistent with [18] and [19]. An intended BS is located at $\phi_0 = 0°$, and four unintended BSs are located at $\phi_1 = 140°$, $\phi_2 = -70°$, $\phi_3 = 70°$, and $\phi_4 = 140°$. To analyze the performance of the SEOANS algorithm, we use SEOANS to optimize the beam pattern of the 10-node random antenna array and 64-node random antenna array, and compare this beam pattern with the beam pattern obtained by CEO-based node selection algorithm. The results are shown in Figs. 9(a) and 9(b). The figures indicate that the width of the mainlobe is basically the same. Thus, the differences of the direction of the beam pattern obtained by each method are

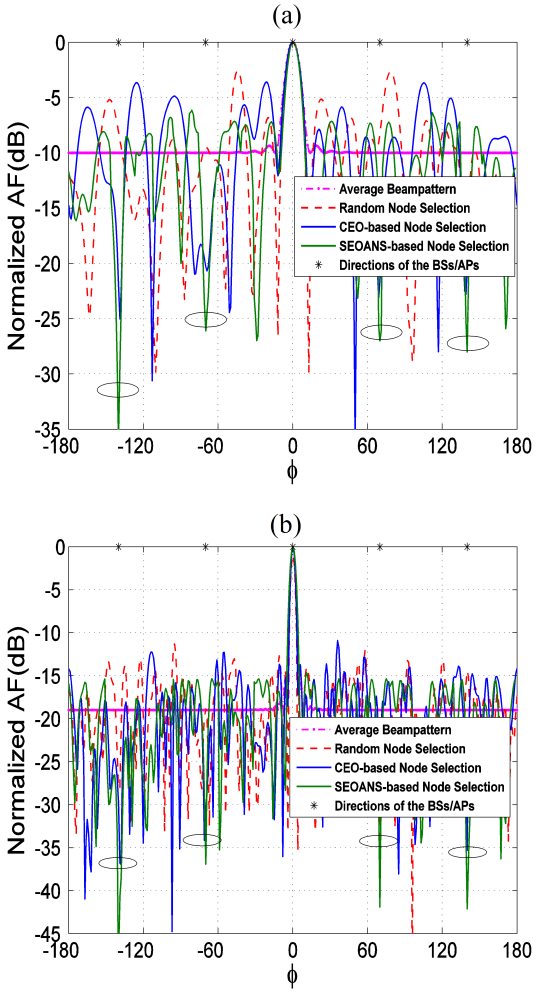


FIGURE 9. Radiation beampattern optimization for the 10 and 64 array nodes obtained by SEOANS and other algorithms in multi-BSs environment. (a) 10 nodes. (b) 64 nodes.

very minimal. However, the nulls of the beampattern obtained by using SEOANS are lower than that of CEO-based node selection algorithm. Moreover, SEOANS has the minimum max SLL at -6.14 and -14.63 dB because the fitness function is a multi-objective optimization function, which can optimize the maximum SLL and the nulls simultaneously. However, the maximum SLL of the 10-node array and 64-node array obtained by SEOANS in this scenario is higher than the corresponding SLL in the single BS scenario. This result occurred because SEOANS not only needs to optimize the SLL but also the nulls. Thus, the two optimization objectives need to be balanced and cannot guarantee the best result for each individual goal. Therefore, Fig. 9 indicates that the beam pattern obtained using SEOANS compromises the direction performance of the mainlobe, but this result does not affect the application scenarios of WSNs.

B. COMMUNICATION DELAY ANALYSIS

According to the analysis in Section IV E, CB has the advantages for long-distance communication in WSNs. Therefore,

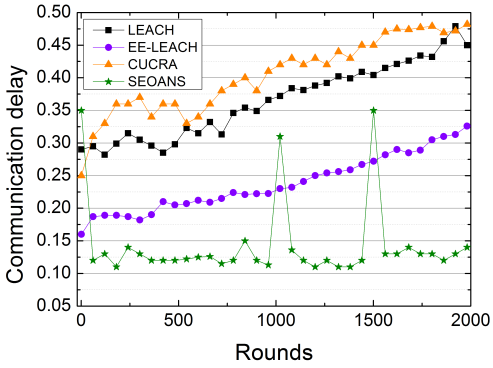


FIGURE 10. Communication delay in large monitor area.

we can select node S_{node_remote}, which is far away from the BS, to run SEOANS to verify the performance of the algorithm. The distance between the BS and S_{node_remote} is 1320.34 m. Fig. 10 shows the communication delays of LEACH, EE-LEACH, CUCRA, and SEOANS. SEOANS has significant advantage in terms of communication delay because the nodes in the algorithm can directly communicate with the BS using beamforming without multi-hop in the data transmission stage. The selected node array can remain stable for a long time. Thus, selecting the array nodes in every round is unnecessary. However, SEOANS has the maximum communication delay at the initial stage of the network because S_{node_remote} needs to select a set of array nodes, causing the increase of communication delay. LEACH, EE-LEACH, and CUCRA are all clustering algorithms. However, LEACH and EE-LEACH assumed that the cluster head can directly communicate with the BS and CUCRA assumes that the cluster head undergoes multi-hop communication with the BS. The communication delay of LEACH is less than that of CUCRA. However, the clusters need to be re-divided during the rounds. The energy consumption of the cluster head will increase dramatically, and the number of the dead nodes will increase constantly in the case of long-range network, causing the overhead of communication delay to increase unceasingly with the operation of the network. Moreover, to verify the faulty tolerant mechanism of SEOANS, we assumed that some of the array nodes malfunction in round 1000 and 1500, respectively. Thus, the algorithm must be re-run. The figure shows that the communication delay of SEOANS increased significantly at round 1000 and 1500, and the communication delay is restored to a lower level after re-running the node selection algorithm.

C. ENERGY CONSUMPTION ANALYSIS

1) AVERAGE ENERGY CONSUMPTION OF THE NODE

Fig.11(a) shows the average energy consumptions of SEOANS and other algorithms. The figure indicates that SEOANS has advantage in average energy consumption compared with LEACH, HEED, LEACH-ERE, EE-LEACH, and CUCRA because the BS can achieve an increase of N^2 gain if N sensor nodes transmit data through CB, and the power

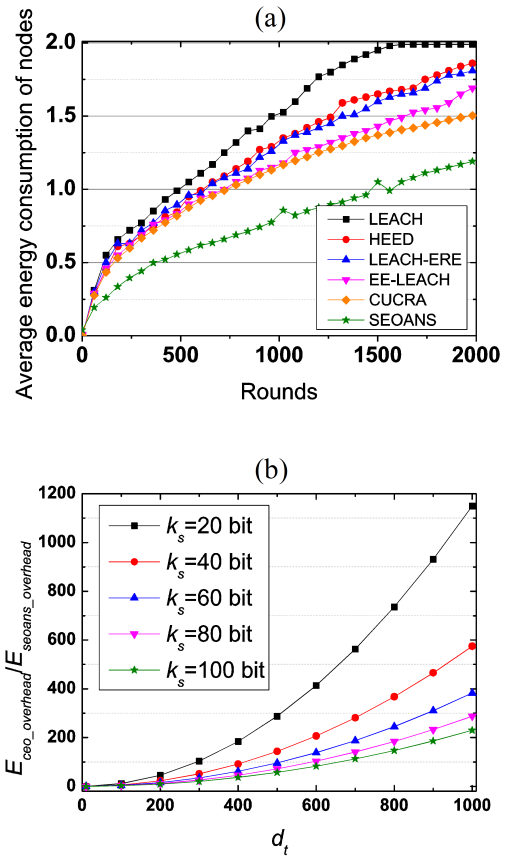


FIGURE 11. (a) Average energy consumption in the large monitor area. (b) Energy overhead ratio of the selection stage with CEO-based node selection algorithm and SEOANS.

of each sensor node in the network will be reduced to $1/N^2$. Thus, the SEOANS algorithm based on CB can save energy.

Furthermore, the average energy consumption of SEOANS is higher than that of other algorithms in the initial stage of the network. This result occurs because it needs to spend more energy for array node selection and inner array communications. The source node requires a large number of nodes in long-distance communication to evidently increase the energy consumption overhead of selecting array nodes. However, based on the analysis in Section IV E, the energy consumption overhead of node selection can be basically disregarded compared with the energy consumption overhead of CB stage. Consistent with the communication delay, the node faulty of SEOANS appeared in 1000 and 1500 rounds were produced and the energy consumption increased abruptly because the array node selection algorithm was re-run. After utilizing the faulty tolerant mechanism, the energy consumption tends to normalize, thus demonstrating the effectiveness of this mechanism of SEOANS.

2) ENERGY CONSUMPTION OF ARRAY NODE SELECTION

The CEO-based node selection algorithm mainly includes four steps, namely, the source node broadcasts node selection message, sending back the response message by the array node, the source node confirmation, and BS trial communication. The overhead of these steps can be described as follows:

$$\begin{aligned}
&E_{ceo-overhead} \\
&\quad = (E_{T_sk} + N_L E_{R_candi}) + (N_L E_{T_candi} \\
&\qquad + E_{R_sk}) + (E_{T_sk} + N_L E_{R_candi}) \\
&\qquad + (N_{trial} \cdot E_{trial}) + E_{R_sk} \\
&\quad = 2E_{T_sk} + 2N_L E_{R_candi} + N_L E_{T_candi} \\
&\qquad + E_{R_sk} + N_{trial} \cdot E_{trial} + E_{R_sk} \\
&\quad = (k_{ceo}E_{elec} + k_{ceo}\varepsilon_{fs}d_R^2 + N_L k_{ceo}E_{elec}) \\
&\qquad + (N_L k_{ceo}E_{elec} + N_L k_{ceo}\varepsilon_{fs}d_R^2 \\
&\qquad + k_{ceo}E_{elec}) + (k_{ceo}E_{elec} + k_{ceo}\varepsilon_{fs}d_R^2 \\
&\qquad + N_L k_{ceo}E_{elec}) + N_{trial}.\frac{k_{ceo}\varepsilon_{fs}d_t^2}{L} + k_{ceo}E_{elec} \\
&\quad = (4 + 3N_L)k_{ceo}E_{elec} + (2 + N_L)k_{ceo}\varepsilon_{fs}d_R^2 \\
&\qquad + N_{trial} \cdot \frac{k_{ceo}\varepsilon_{fs}d_t^2}{L}
\end{aligned} \tag{39}$$

where E_{T_sk} and E_{R_sk} are the transmit and receive energy consumption of the source node, respectively. E_{T_candi} and E_{R_candi} are the transmit and receive energy consumption of the candidate node, respectively. d_R is the communication radius of the source node in CEO-based node selection algorithm. N_L is the number of candidate nodes, N_{trial} is the time of the BS trial communications, k_{ceo} is the bit length of the selection information in CEO-based node selection algorithm, and L is the number of the nodes in the trial set. Eq. (39) shows that the main energy consumption of the CEO-based node selection algorithm is the BS trial communication. N_{trial} is 769 based on the simulation results. We set the length of the control message to 1 bit in the node selection stage of the CEO-based node selection algorithm. Thus, the ratio of the energy consumption of the scheduling stage of the CEO-based node selection algorithm and SEOANS is in Eq. (40), as shown at the top of the next page.

Fig. 11(b) shows the energy consumption ratio of SEOANS and CEO-based node selection algorithm with different node selection message lengths k_s. The figure indicates that the energy consumption of CEO-based node selection algorithm is significantly higher than SEOANS especially when k_s is short. Moreover, with the increasing of d_t, the energy consumption of CEO-based node selection algorithm will show an exponential growth trend. This outcome is unacceptable for the energy-constrained node in WSNs. Thus, the CEO-based node selection algorithm is not suitable in a practical environment in terms of energy consumption.

VI. CONCLUSION

In this paper, a novel SEOANS approach is proposed to effectively select array nodes for CB in WSNs. First, SEOANS determines the energy efficient number of array nodes and chooses a set of location optimized array nodes. Second, a new CSCSO algorithm is proposed to optimize the excitation current of each selected array node for further

$$\eta_E = \frac{E_{ceo\_overhead}}{E_{seoans\_overhead}} = \frac{(4+3N_L)k_{ceo}E_{elec} + (2+N_L)k_{ceo}\varepsilon_{fs}d_R^2 + N_{trial}\cdot\frac{k_{ceo}\varepsilon_{fs}d_t^2}{L}}{k_sE_{elec}(2N_{candi}+2N_{beam}+4) + k_s\varepsilon_{fs}R_c^2(N_{candi}+N_{beam}+2)} \quad (40)$$

suppressing the SLL. Finally, the scheduling and the fault tolerance mechanisms are designed in SEOANS. Simulation results show that the location optimization method and excitation current optimization based on CSCSO can suppress the maximum SLL effectively. CSCSO exhibits faster convergence speed and lower maximum SLL than that of CS and BBO. Moreover, CSCSO has advantages in CCDF. In multi-BS scenario, the nulls of the beampattern obtained using SEOANS are lower than that of the CEO-based node selection method. In addition, the communication delay of SEOANS is reduced by 60.77%, 39.86%, and 64.52% compared with LEACH, EE-LEACH, and CUCRA, respectively. The average energy consumption is decreased by 46.37%, 38.42%, 37.28%, 31.68% and 28.52% compared with LEACH, HEED, LEACH-ERE, EE-LEACH and CUCRA, respectively, which improves the network lifetime. However, the energy efficiency of SEOANS can only be reflected in the large-scale networks. If the network scale is small, SEOANS will spend significant energy in the node selection stage, and it is not suitable to use the approach.

ACKNOWLEDGMENTS

We would like to thank the anonymous referees for their many valuable suggestions and comments.

REFERENCES

[1] J. Zhang, Y. Liu, D. Y. Sun, and B. Li, "Prolonging the lifetime of wireless sensor networks by utilizing feedback control," *Wireless Netw.*, vol. 20, no. 7, pp. 2095–2107, Oct. 2014.

[2] D. S. Ghataoura, J. E. Mitchell, and G. E. Matich, "Networking and application interface technology for wireless sensor network surveillance and monitoring," *IEEE Commun. Mag.*, vol. 49, no. 10, pp. 90–97, Oct. 2011.

[3] Q.-Q. Li, H. Gong, M. Liu, M. Yang, and J. Zheng, "On prolonging network lifetime through load-similar node deployment in wireless sensor networks," *Sensors*, vol. 11, no. 4, pp. 3527–3544, 2011.

[4] M. U. Ilyas and H. Radha, "A dynamic programming approach to maximizing a statistical measure of the lifetime of sensor networks," *ACM Trans. Sensor Netw.*, vol. 8, no. 2, 2012, Art. no. 18.

[5] Q. Ju and Y. Zhang, "Predictive power management for Internet of battery-less things," *IEEE Trans. Power Electron.*, vol. 33, no. 1, pp. 299–312, Jan. 2017.

[6] X. Wu, G. Chen, and S. K. Das, "Avoiding energy holes in wireless sensor networks with nonuniform node distribution," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 5, pp. 710–720, May 2008.

[7] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[8] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, Oct./Dec. 2004.

[9] J. S. Lee and W. L. Cheng, "Fuzzy-logic-based clustering approach for wireless sensor networks using energy predication," *IEEE Sensors J.*, vol. 12, no. 9, pp. 2891–2897, Sep. 2012.

[10] G. S. Arumugam and T. Ponnuchamy, "EE-LEACH: Development of energy-efficient LEACH protocol for data gathering in WSN," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, pp. 1–9, 2015.

[11] W. Tong, W. Jiyi, X. He, Z. Jinghua, and C. Munyabugingo, "A cross unequal clustering routing algorithm for sensor network," *Meas. Sci. Rev.*, vol. 13, no. 4, pp. 200–205, 2013.

[12] K. Zarifi, S. Affes, and A. Ghrayeb, "Collaborative null-steering beamforming for uniformly distributed wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1889–1903, Mar. 2010.

[13] B. Bejar Haro, S. Zazo, and D. P. Palomar, "Energy efficient collaborative beamforming in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 496–510, Jan. 2014.

[14] M. F. A. Ahmed and S. A. Vorobyov, "Collaborative beamforming for wireless sensor networks with Gaussian distributed sensor nodes," *IEEE Trans. Wireless Commun.*, vol. 8, no. 2, pp. 638–643, Feb. 2009.

[15] L. Dong, A. P. Petropulu, and H. V. Poor, "A cross-layer approach to collaborative beamforming for wireless ad hoc networks," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2981–2993, Jul. 2008.

[16] W. Zhang, H. Yang, Z. Yang, L. Huang, and Z. Guo, "Collaborative beamforming for wireless sensor networks with sector-based node selection," in *Proc. Int. Conf. Commun., Circuits Syst. (ICCCAS)*, Nov. 2013, pp. 141–144.

[17] H. Ochiai, P. Mitran, H. V. Poor, and V. Tarokh, "Collaborative beamforming for distributed wireless ad hoc sensor networks," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4110–4124, Nov. 2005.

[18] M. F. A. Ahmed and S. A. Vorobyov, "Sidelobe control in collaborative beamforming via node selection," *IEEE Trans. Signal Process.*, vol. 58, no. 12, pp. 6168–6180, Dec. 2010.

[19] J.-C. Chen, C.-K. Wen, and K.-K. Wong, "An efficient sensor-node selection algorithm for sidelobe control in collaborative beamforming," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 5984–5994, Aug. 2016.

[20] N. N. N. A. Malik, M. Esa, S. K. S. Yusof, and S. A. Hamzah, "Optimization of linear sensor node array for wireless sensor networks using particle swarm optimization," in *Proc. Asia–Pacific Microw. Conf.*, Dec. 2010, pp. 1316–1319.

[21] N. N. N. A. Malik, M. Esa, S. K. S. Yusof, S. A. Hamzah, and M. K. H. Ismail, "Circular collaborative beamforming for improved radiation beampattern in WSN," *Int. J. Distrib. Sensor Netw.*, vol. 2013, pp. 914–917, Jul. 2013.

[22] G. Sun, Y. H. Liu, J. Zhang, A. M. Wang, and X. Zhou, "Node selection optimization for collaborative beamforming in wireless sensor networks," *Ad Hoc Netw.*, vol. 37, pp. 389–403, Feb. 2016.

[23] S. Jayaprakasam, S. K. A. Rahim, C. Y. Leow, and T. O. Ting, "Sidelobe reduction and capacity improvement of open-loop collaborative beamforming in wireless sensor networks," *PLoS ONE*, vol. 12, no. 5, p. e0175510, 2017.

[24] S. Jayaprakasam, S. K. A. Rahim, and C. Y. Leow, "Distributed and collaborative beamforming in wireless sensor networks: Classifications, trends, and research directions," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2092–2116, 4th Quart., 2017.

[25] S. Jayaprakasam, S. K. A. Rahim, and C. Y. Leow, "PSOGSA-explore: A new hybrid metaheuristic approach for beampattern optimization in collaborative beamforming," *Appl. Soft Comput.*, vol. 30, pp. 229–237, May 2015.

[26] J. Feng *et al.*, "Energy-efficient transmission for beamforming in wireless sensor networks," in *Proc. 7th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw. (SECON)*, Jun. 2010, pp. 1–9.

[27] A. Sharaqa and N. Dib, "Circular antenna array synthesis using firefly algorithm," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 24, no. 2, pp. 139–146, Mar. 2014.

[28] V. Agrawal and Y. Lo, "Mutual coupling in phased arrays of randomly spaced antennas," *IEEE Trans. Antennas Propag.*, vol. AP-20, no. 3, pp. 288–295, May 1972.

[29] M. M. Khodier and C. G. Christodoulou, "Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization," *IEEE Trans. Antennas Propag.*, vol. 53, no. 8, pp. 2674–2679, Aug. 2005.

[30] S. Jayaprakasam, S. K. A. Rahim, C. Y. Leow, T. O. Ting, and A. A. Eteng, "Multiobjective beampattern optimization in collaborative beamforming via NSGA-II with selective distance," *IEEE Trans. Antennas Propag.*, vol. 65, no. 5, pp. 2348–2357, May 2017.
[31] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biol. Inspired Comput. (NaBIC)*, Dec. 2009, pp. 210–214.
[32] X. Meng, Y. Liu, X. Gao, and H. Zhang, "A new bio-inspired algorithm: Chicken swarm optimization," in *Proc. Int. Conf. Swarm Intell.*, 2014, pp. 86–94.
[33] M. A. Nowak and R. M. May, "Evolutionary games and spatial chaos," *Nature*, vol. 359, no. 6398, pp. 826–829, Oct. 1992.
[34] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008.

GENG SUN received the B.S. degree in communication engineering from Dalian Polytechnic University, China, in 2011.

He is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Jilin University. He is currently a Visiting Ph.D. student with the School of Electrical and Computer Engineering, Georgia Institute of Technology. His research interests include the beamforming and beam pattern synthesis of array antennas.

YANHENG LIU received the M.Sc. and Ph.D. degrees in computer science from Jilin University, China.

He was a Visiting Scholar with the University of Hull, England, The University of British Columbia, Canada, and the University of Alberta, Canada. He is currently a Professor with Jilin University. He has co-authored over 120 research publications in peer-reviewed journals and international conference proceedings of which one has won best paper awards. His primary research interests are in network security, network management, mobile computing network theory and applications, and so on.

SHUANG LIANG received the B.S. degree in communication engineering from Dalian Polytechnic University, China, in 2011.

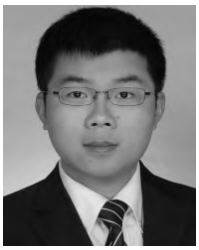
She is currently pursuing the M.S. degree in software engineering with Jilin University. Her research interests focus on wireless communication and design of array antennas.

ZHAOYU CHEN received the B.S. degree in computer science from Jilin University, China, in 2015.

He is currently pursuing the M.S. degree in computer science with Jilin University. His research interests focus on wireless communications, networks, and optimizations.

AIMIN WANG received the B.S. degree in computer software, the M.S. degree in computer application technology, and the Ph.D. degree in communication and information system from Jilin University.

He is currently an Associate Professor with Jilin University. His research interests are wireless sensor networks and QoS for multimedia transmission.



QIANAO JU received the B.S. and M.E. degrees in electrical engineering from Shanghai Jiao Tong University, Shanghai, in 2010 and 2013, respectively, and the M.S. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, in 2013, where he is currently pursuing the Ph.D. degree. His research interests include Internet of Things, energy harvesting systems, adaptive power management, and embedded systems.

YING ZHANG received the M.S. degree in materials engineering from the University of Illinois, Chicago, in 2001, the M.S. degree in electrical engineering from the University of Massachusetts Lowell in 2002, and the Ph.D. degree in systems engineering from the University of California at Berkeley, in 2006.

She is currently an Associate Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology. Her research interests are in the areas of sensors and smart wireless sensing systems, power management for energy-harvesting wireless sensor networks, intelligent monitoring and diagnostic systems, artificial intelligence, information retrieval and data mining, and computer-aided optimal design.