Differential Evolution-based 3D Directional Wireless Sensor Network Deployment Optimization

Bin Cao, *Member, IEEE,* Xinyuan Kang, Jianwei Zhao, Po Yang, *Member, IEEE,* Zhihan Lv, *Member, IEEE,* and Xin Liu

Abstract—Wireless sensor networks (WSNs) are applied more and more widely in real life. In actual scenarios, 3D directional wireless sensors (DWSs) are constantly employed, thus, research on the real-time deployment optimization problem of 3D directional wireless sensor networks (DWSNs) based on terrain big data has more practical significance. Based on this, we study the deployment optimization problem of DWSNs in the 3D terrain through comprehensive consideration of coverage, lifetime, connectivity of sensor nodes, connectivity of cluster headers and reliability of DWSNs. We propose a modified differential evolution (DE) algorithm by adopting CR-sort and polynomial-based mutation on the basis of the cooperative coevolutionary (CC) framework, and apply it to address deployment problem of 3D DWSNs. In addition, to reduce computation time, we realize implementation of message passing interface (MPI) parallelism. As is revealed by the experimentation results, the modified algorithm proposed in this paper achieves satisfying performance with respect to either optimization results or operation time.

Index Terms—cooperative coevolution (CC), linear crossover, polynomial-based mutation, differential evolution (DE), 3D directional wireless sensor networks (DWSNs), coverage, lifetime, connectivity, reliability

1 Introduction

With the continuous development of communication technologies and smart sensing devices, the Internet of Things(IoT) provides more and more convenience and efficiency to human living [1]. Wireless sensor networks (WSNs), which are the basic technology of IoT, composed of a certain number of lightweight, low-cost wireless sensor nodes [2], have also experienced great progress. Utilizing the vibration sensors for identification and narrow-band internet of things for communication, Jia et al. [3] proposed an edge computing-based intelligent manhole cover management system. Aguirre et al. [4] applied WSNs to the real-time monitoring of urban traffic environments. Fosalau et al. [5] monitored catastrophic natural phenomena (e.g., landslides) by deploying highly sensitive sensor nodes to perceive the moving direction and displacement of soil.

With the increasingly widespread application of IoT and WSNs, many scholars have studied related issues. Santigo et al. [6] proposed a modified feature selection algorithm which can separate and prioritize the sensor data and applied to industrial IoT. The deployment problem of WSNs can be well resolved by biological heuristic algorithms. How to improve the coverage and prolong the lifetime of the WSNs are two main research directions. By combining adaptive length coding, Alia et al. [7] presented a novel algorithm that could automatically modify and determine the optimum quantity and positions of sensor nodes to achieve coverage maximization with the cost minimization. Manju et al. [8] proposed a method of setting nodes work alternately and giving coverage priority of crucial monitoring areas to prolong the network lifetime. Tuba et al. [9] employed the fireworks algorithm [10] to optimize the coverage rate of WSNs, which realized coverage rate maximization via finding "optimum" sensor positions. However, they only researched the deployment problem on 2D plane, while sensor nodes in the real world exist in 3D space, the sensing range of sensors is 3D and has sensing angles limited. Accordingly, directional sensor nodes with limited sensing angles are more accordant with the actual situation. The concept of directional sensors was proposed by Ma et al. [11]. Based on which, Teng et al. [12] proposed a fuzzy ring based fan-shaped sensing model, and this study was more practical. Therefore, research on the deployment problem of directional wireless sensor networks (DWSNs) on 3D terrain has more realistic significance and practical value.

Sung et al. [13] proposed a distributed greedy algorithm to improve the coverage of directional sensor nodes. Considering the directionality and sensing angle and combining Voronoi diagrams. Nevertheless, only one objective, coverage, was considered. Cao et al. [14] considered the coverage and lifetime of directional sensor nodes in 3D

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 61303001, in part by the Opening Project of Guangdong High Performance Computing Society under Grant No. 2017060101, in part by the Foundation of Key Laboratory of Machine Intelligence and Advanced Computing of the Ministry of Education under Grant No. MSC-201602A, and in part by the Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase) under Grant No. U1501501. (Corresponding authors: Po Yang, Bin Cao)

Bin Cao, Xinyuan Kang, Jianwei Zhao and Xin Liu are in the Hebei University of Technology, Tianjin, 300401, China; Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education; Hebei Provincial Key Laboratory of Big Data Calculation, China. (email: caobin@scse.hebut.edu.cn; 201522102032@stu.hebut.edu.cn; 201422102003@stu.hebut.edu.cn; xinliu10@163.com)

Po Yang is in the Department of Computer Science, Liverpool John Moores University, Liverpool, UK. (email: p.yang@ljmu.ac.uk)

Zhihan Lv is in the School of Data Science and Software Engineering, Qingdao University, Qingdao 266071, China. (email: lvzhihan@gmail.com)

industrial space with obstacles, and distributed parallelism was conducted to reduce the computation time.

Node clustering and routing are two well-known methods to prolong lifetime of WSNs [15]. Usually, these two methods are simultaneously employed to improve the energy utilization rate. Halder et al. [16] discovered that the energy imbalance across the network is mainly owes to the data transmission to relay nodes from different sections, and they put forward a heterogeneous node deployment strategy to extend network lifetime. Chu et al. [17] proposed a distributed cooperative topology control and adaptation algorithm to achieve the extend of network lifetime. Hacioglu et al. [18] presented a clustering-based routing methodology, which minimized the communication costs among clusters and maximized the node quantity in each cluster, and NSGA-II [19] was combined to select excellent solutions. However, all these studies only explored the case of 2D plane.

To achieve the data transmission stability in IoT, a stable and reliable network is essential [20]. Connectivity is basic for the reliable data transmission, and it's basic of the topology control and routing protocol. Besides the basic coverage and lifetime, connection and reliability [21] also should be concerned to ensure the wireless networks performance. Li et al. [22] proposed a deployment strategy for simultaneously considering coverage and connectivity based on the elitist non-dominated sorting genetic algorithm (NSGA-II) [19]. Zakia et al. [23] considered the Quality of Monitoring (QoM) and wireless network connectivity and proposed a 3D underwater deployment scheme. [24] considered coverage, connectivity uniformity and deployment cost, and [24] proposed a distributed parallel cooperative coevolutionary multi-objective large-scale immune algorithm to solve it, while [25] just utilized the existing algorithm, however reliability was both not into consideration. Li et al. [26] improved a three-factor user authentication protocol for WSN to satisfy the security requirement in IoT application. Deif et al. [27] proposed a modified Ant Colony Optimization (ACO) to improve the reliability of WSNs at a minimum deployment cost. Machado et al. [28] presented a diffusion-based approach to satisfy the coverage, connectivity and reliability of WSNs, however it is deployed on the 2D plane. Above all, there are few studies simultaneously considered the coverage, lifetime, connectivity and reliability of DWSNs on 3D terrain.

This paper comprehensively considers the coverage, lifetime, the connectivity of sensor nodes, the connectivity of cluster headers, and the reliability of fuzzy ring-based DWSNs on 3D terrain. To address it, we present a modified DE algorithm: cooperative coevolutionary (CC) [29] differential evolution (DE) algorithm [30] with CR-sort [31] and polynomial-based mutation (CCDEXSPM). In this paper, our main contributions are as follows:

A. In the process of mutation in DE, we simply select parents from the whole population with uniform probability, ensuring a greater search direction of the population and avoiding premature convergence for falling into local optima. For the crossover factor, we adopt a novel dynamic updating scheme of CR-sort [31].
B. To avoid premature convergence of the population, after the mutation and crossover, we append a polynomial-based mutation operator to perform a second-time mutation to each newly generated individual.
C. We combine the above evolutionary strategy with the novel CC strategy proposed in [32] by adopting fixed grouping [29] and allocating variables of the same property to the same group to improve optimization efficiency.
D. To improve the operation speed, message passing interface (MPI) parallelism is adopted.

The rest of the article is arranged as follows: Section 2 presents related concepts. Our work is detailed in Section 3. The experiments and analysis are provided in Section 4. Finally, we conclude this paper in Section 5.

2 Related Concepts

2.1 WSN Sensing Model

According to the shape of the coverage region, the sensing model can be classified as: omni-directional sensing model and directional sensing model. Traditional sensor nodes are generally omni-directional sensor nodes, the sensing range of which is a spherical region.For directional sensor nodes, the sensing range is limited with respect to the horizontal sensing angle,as illustrated in Fig. 1, the deterministic sensing angle is $(\theta_f - \theta_u)$ and the angle range of the fuzzy ring is $2\theta_u$.

If a point p can be detected by a sensor node s, the Line-of-Sight (LOS) is satisfied with respect to s and p, and Fig. 2 is an instance of non-LOS (NLOS). Considering of LOS, the sensing range [12] of directional sensor nodes can be represented as Eq. 1.

2.2 Coverage Degree

The sensing probability of a certain point is influenced by its distance from the sensor node. This paper adopts multi-point coverage strategy and the Sugeno measure [33],

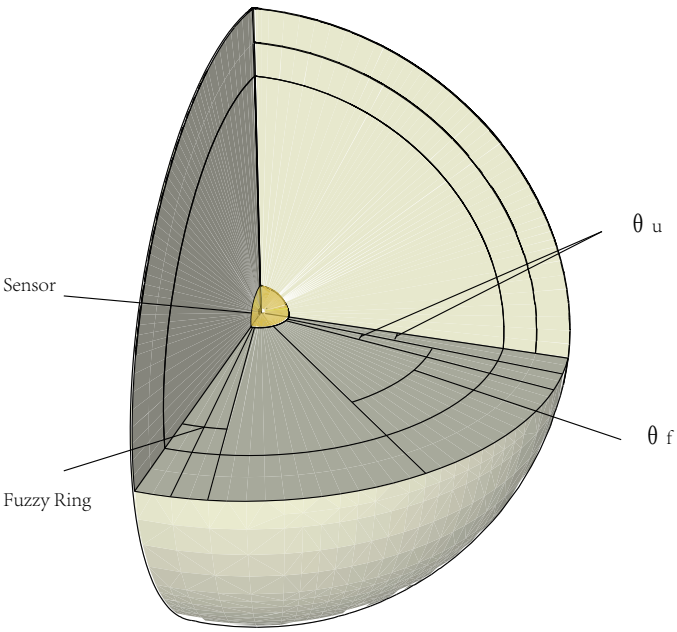


Fig. 1. Directional sensing model.

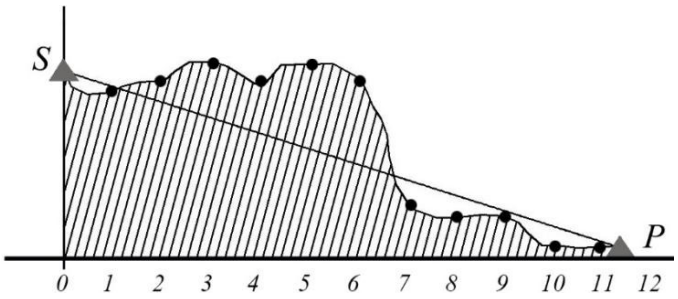


Fig. 2. An instance of NLOS.

$$O_q(s,p)=\begin{cases} 0, & \theta(s,p)>(\theta_f+\theta_u) \\ & \text{or } \Delta(s,p)>(S_r+U_r) \\ & \text{or if } NLOS \\ 1-\int_{-\infty}^{\theta(s,p)-\theta_f}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{t^2}{2\sigma^2}}dt, & (\theta_f-\theta_u)<\theta(s,p)<(\theta_f+\theta_u) \\ & \Delta(s,p)<(S_r-U_r) \\ & \text{and if } LOS \\ e^{-\alpha\times dist^{\beta}}\left(1-\int_{-\infty}^{\theta(s,p)-\theta_f}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{t^2}{2\sigma^2}}dt\right), & (\theta_f-\theta_u)<\theta(s,p)<(\theta_f+\theta_u) \\ & (S_r-U_r)<\Delta(s,p)<(S_r+U_r) \\ & \text{and if } LOS \\ e^{-\alpha\times dist^{\beta}}, & \theta(s,p)<(\theta_f-\theta_u) \\ & (S_r-U_r)<\Delta(s,p)<(S_r+U_r) \\ & \text{and if } LOS \\ 1, & \theta(s,p)<(\theta_f-\theta_u) \\ & \Delta(s,p)<(S_r-U_r) \\ & \text{and if } LOS \end{cases} \tag{1}$$

where $\theta(s,p)$ is the angle between the main sensing direction of sensor node s and the line connecting sensor node s and point p.

---

[34] to describe the uncertain coverage of WSNs, which can be detailed as follows:

$$O_q(p)=\min\left(1,\frac{1}{\lambda}\left\{\prod_{k=1}^{n}\left[1+\lambda\times O_q(s_k,p)\right]-1\right\}\right) \tag{2}$$

where n denotes the number of sensor nodes and λ $(-1\leq\lambda<0)$ is the fusion operator [35]. Let O_{th} denote the coverage degree threshold. To evaluate the coverage degree, define [35]:

$$O_s(p)=\begin{cases} 1, & O_q(p)\geq O_{th} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$QoC=\frac{1}{P}\sum_{j=1}^{P}O_s(p_j) \tag{4}$$

where QoC denotes the quality of coverage of the target region after each deployment and P is the number of points.

2.3 Lifetime Model

2.3.1 Energy Consumption Model

The radio energy consumption model we adopt is the same as in [15]. By setting the distance threshold d_{th} and adopting different energy consumption patterns, the specific equation for transmission is in the following:

$$E_T(l,d)=\begin{cases} lE_{elec}+l\varepsilon_{fs}d^2, & d<d_{th} \\ lE_{elec}+l\varepsilon_{mp}d^4, & d\geq d_{th} \end{cases} \tag{5}$$

where l is the message quantity with the unit of bit, E_{elec} denotes the energy consumption parameter, and ε_{fs} and ε_{mp} are parameters in the free-space and multi-path channels, respectively. The energy consumption for receiving l bit messages, can be calculated as follows:

$$E_R(l)=lE_{elec} \tag{6}$$

2.3.2 Lifetime Model of CHs

The energy consumption model of sensor nodes E_{sensor} is composed of two parts: sensing energy consumption E_{sense}, and communication energy consumption E_{com} for transmitting data to CHs, which can be formulated as:

$$E_{sensor}=E_{sense}+E_{com} \tag{7}$$

The energy consumption $E_{clu}(g_i)$ within the cluster of CH g_i contains three majors consumptions: the E_{rec}, E_{agg} and E_{send} represent the energy consumption of receiving, aggregating and sending one bit data, respectively :

1) The energy consumption of receiving l_0-bit data from r_i sensor nodes in the current cluster will be $r_i l_0 E_{rec}$.

2) The energy consumption of aggregating the l_0-bit data of all r_i sensor nodes in the cluster reaches $r_i l_0 E_{agg}$.
3) The energy consumption of transmitting l_0-bit data with the distance d_i will be:

$$l_0 E_{send} = \begin{cases} l_0 E_{send} + l_0 \varepsilon_{fs} d_i^2, & d_i < d_{th} \\ l_0 E_{send} + l_0 \varepsilon_{mp} d_i^4, & d_i \geq d_{th} \end{cases} \quad (8)$$

A compression ratio r_{cmp} is set to represent the degree of data aggregation. The energy consumed for transmitting data received from r_i sensor nodes to the next hop will be $r_i r_{cmp} l_0 E_{send}$.

Therefore, the energy consumption $E_{clu}(g_i)$ of CH g_i in its own cluster can be represented as:

$$E_{clu}(g_i) = r_i l_0 \left(E_{rec} + E_{agg} + r_{cmp} E_{send}\right) \quad (9)$$

For a relay node g_i, the energy consumption $E_{gateway}(g_i)$ for relaying data of other CHs mainly includes two aspects:

1) Energy consumption for receiving data of s_i sensor nodes from the previous hop will be $s_i r_{cmp} l_0 E_{rec}$.
2) Energy consumption for relaying data of s_i sensor nodes will be $s_i r_{cmp} l_0 E_{send}$.

Therefore, the energy consumption $E_{gateway}$ of relay node g_i for relaying data for other CHs can be represented as:

$$E_{gateway}(g_i) = s_i r_{cmp} l_0 \left(E_{rec} + E_{send}\right) \quad (10)$$

Finally, the energy consumption $E(g_i)$ of relay node g_i can be denoted as:

$$\begin{aligned} E(g_i) &= E_{clu}(g_i) + E_{gateway}(g_i) \\ &= r_i l_0 \left(E_{rec} + E_{agg} + r_{cmp} E_{send}\right) + \\ &\quad s_i r_{cmp} l_0 \left(E_{rec} + E_{send}\right) \\ &= \left(r_i + s_i r_{cmp}\right) l_0 E_{rec} + \\ &\quad r_i l_0 E_{agg} + \\ &\quad \left(r_i + s_i\right) r_{cmp} l_0 E_{send} \end{aligned} \quad (11)$$

The WSN lifetime adopts the pattern of N-of-N, that is, the lifetime of the whole WSN vanishes when the first CH exhausts its energy.

Assuming that residual energy of CH g_i is $E_{residual}(g_i)$, its lifetime $L(i)$ can be represented as:

$$L(i) = \frac{E_{residual}(g_i)}{E(g_i)} \quad (12)$$

2.4 Network Connectivity

WSNs accomplish the monitoring task mainly through gathering and transferring information and data. All nodes cooperate with each other to guarantee the normal operation of the WSNs. Therefore, the network connectivity is an important guarantee of the network functionality. Each sensor node selects one CH to join its cluster and transmits the gathered information to its CH. Then, the CHs aggregate the data transferred from sensor nodes in the clusters and relay them to the next hops. We considered the connectivity of CHs and sensor nodes, respectively.

For the CHs, we can guarantee each two CHs perform communication directly or indirectly via other CHs, that is, the number of CHs in the maximal connected set of the whole network is supposed to be the CH number in the network. The uniformity of the numbers of CHs that all CHs can communicate with should be optimized. The standard deviation can be utilized to measure the uniformity [36]:

$$f_{UniOfCH} = \frac{1}{1 + f_{CH}^{std}} \quad (13)$$

$$f_{CH}^{std} = \sqrt{\frac{\sum_{i=1}^{M_{CH}} \left(c_i - A_{CH}\right)^2}{M_{CH}}} \quad (14)$$

$$A_{CH} = \frac{\sum_{i=1}^{M_{CH}} c_i}{M_{CH}} \quad (15)$$

where $f_{UniOfCH}$ is the measure of *Connectivity Uniformity of CH*; f_{CH}^{std} denotes the standard deviation; M_{CH} is the number of CHs in the largest connected subcomponent, c_i is the number of CHs for the CH i can communicate with, and A_{CH} is the average value of the number of CHs that each CH can communicate with in the set. If the size of the largest connected set is less than the total number of CHs N_{CH}, a penalty will be assigned which is expressed as follows:

$$penalty\left(N_{CH}, M_{CH}\right) \quad (16)$$

where p denotes the penalty factor, which is assigned a great value (e.g. $1e6$).

For the sensor nodes, to measure the connectivity uniformity of sensor nodes, we can guarantee the uniformity of the distances of all sensor nodes to their corresponding CHs as far as possible, which can be achieved through utilizing the standard deviation. The specific formula of is as follows:

$$f_{UniOfDS} = \frac{1}{1 + f_{DS}^{std}} \quad (17)$$

$$f_{DS}^{std} = \sqrt{\frac{\sum_{i=1}^{N_{DS}} \left(d_i - A_{DS}\right)^2}{N_{DS}}} \quad (18)$$

$$A_{DS} = \frac{\sum_{i=1}^{N_{DS}} d_i}{N_{DS}} \quad (19)$$

where $f_{UniOfDS}$ is the measure of *Connectivity Uniformity of sensor nodes*; N_{DS} is the number of sensor nodes, d_i is the distance of sensor node i to its corresponding CH, A_{DS} is the average distance of sensor nodes to the CHs.

2.5 Network Reliability

The number of nodes is limited and WSNs are usually applied in sever and complicated environments. When one sensor node corrupts or depletes its energy, coverage holes may occur and a part of the area cannot be sensed; when this happens to a CH, the information gathered by sensor nodes in this cluster and the data transferred from the previous hop would be unsuccessful. More seriously, when this CH is located in the crucial position, the normal operation of the whole network will be influenced. Therefore, the network reliability is an important guarantee of the network functionality. The reliability issue has become a research hot spot of WSNs.

To alleviate this situation, through employing the method of multi-hop communication, each CH and each sensor node are associated to several CHs. So if the selected CH of the current node corrupts or depletes the energy, another CH can be chosen for information forwarding, thus the whole network can still work normally.

To guarantee the network reliability, we prescribe an average number of CHs that all sensor nodes and CHs can communicate with, shown as follows:

$$f_{Rel} = \frac{\sum_{i=1}^{N_{DS}} c_i + \sum_{j=1}^{N_{CH}} c_j}{\left(N_{DS} + N_{CH}\right)^2} \tag{20}$$

where c_i and c_j is the number of CHs of sensor node i and CH j can communicate with, respectively.

Moreover, we set a limit of the minimum number of CHs each node should communicate with, as is shown in Eq. 21:

$$\begin{cases} N_{DS}^{CH} \geq 2 \\ N_{CH}^{CH} \geq 2 \end{cases} \tag{21}$$

where N_{DS}^{CH} and N_{CH}^{CH} denote the number of associated CHs of each sensor node and each CH, respectively.

2.6 Objective Function

We have considered the quality of coverage, lifetime, the connectivity uniformity of sensor nodes, the connectivity uniformity of CHs, and the reliability of the WSN, and these five aspects have different importance degrees, we fuse these five aspects by setting different weight value and obtain the final objective function, as follows:

$$\begin{aligned} cost = \quad & st_1 \times QoC + \\ & st_2 \times L_{min} + \\ & st_3 \times f_{UniOfCH} + \\ & st_4 \times f_{UniOfDS} + \\ & st_5 \times f_{Rel} \end{aligned} \tag{22}$$

$$s.t. \begin{cases} penalty\left(N_{CH}, M_{CH}\right) \\ N_S^{CH} \geq 2 \\ N_{CH}^{CH} \geq 2 \end{cases}$$

where st_1, st_2, st_3, st_4 and st_5 are the weight factors, and L_{min} denotes the lifetime of the first CH that exhausts its energy.

3 Our Works

3.1 Directional Sensing Model

In the 3D directional sensing model [12] mentioned in subsection 2.1, although the horizontal angle constraint is considered, the vertical one should also be taken into account. On the basis of the sensing model presented in [12], we add a vertical angle constraint and put forward a modified 3D directional sensing model.

As to the sensing probability with respect to the distance constraint, we use the same calculation method as in [37], which is detailed in Eq. 23. For the horizontal and vertical angle constraints, we adopt the same probability computation method, as shown in Eq. 25:

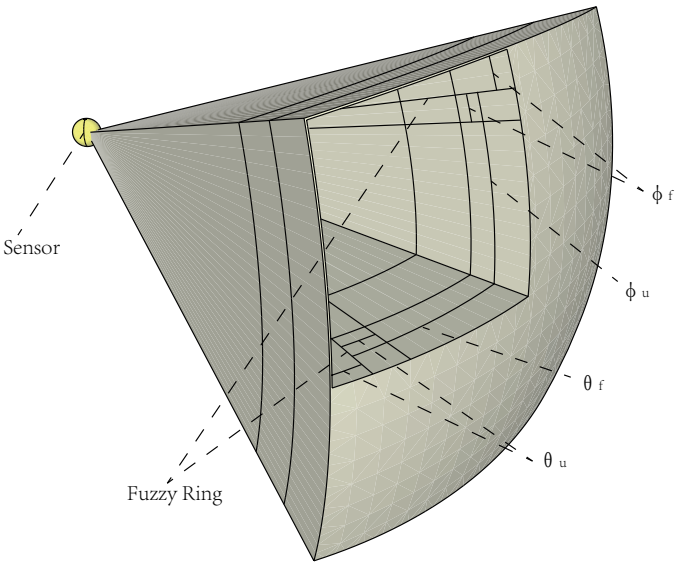


Fig. 3. Modified directional sensing model.

$$O_q\left(s,p\right) = \begin{cases} 1, & \Delta\left(s,p\right) < \left(S_r - U_r\right) \\ & \text{and if } LOS \\ e^{-\alpha \times dist^{\beta}}, & \left(S_r - U_r\right) \leq \Delta\left(s,p\right) < \left(S_r + U_r\right) \\ & \text{and if } LOS \\ 0, & \Delta\left(s,p\right) > \left(S_r + U_r\right) \\ & \text{or if } NLOS \end{cases} \tag{23}$$

$$dist = \Delta\left(s,p\right) - \left(S_r - U_r\right) \tag{24}$$

$$O_q\left(s,p\right) = \begin{cases} 0, & \gamma > \left(\gamma_f + \gamma_u\right) \\ & \text{or NLOS} \\ 1 - \int_{-\infty}^{\gamma - \gamma_f} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-t^2}{2\sigma^2}} dt, & \left(\gamma_f - \gamma_u\right) < \gamma < \left(\gamma_f + \gamma_u\right) \\ & \text{and LOS} \\ 1, & \gamma < \left(\gamma_f - \gamma_u\right) \\ & \text{and LOS} \end{cases} \tag{25}$$

where γ can be the horizontal angle θ or the vertical angle ϕ, γ_f and γ_u are the sensing angles, the radius of the deterministic sensing angle range is $\left(\gamma_f - \gamma_u\right)$, and the radius of the fuzzy ring is $2\gamma_u$, that is to say, the fuzzy ring region is within the angle range of $\left(\gamma_f - \gamma_u\right)$ and $\left(\gamma_f + \gamma_u\right)$. By comprehensively considering the horizontal and vertical angle constraints, we can obtain the modified directional sensing model, as illustrated in Fig. 3.

The specific sensing probability formula is in Eq. 26.

3.2 Routing Algorithm

In the considered WSN, we deploy sensor nodes and relay nodes simultaneously. Each time their positions are determined, we apply the routing algorithm to identify the CH each sensor node belongs to and the routing information of CHs to the BS. After the information collection of all sensor nodes, the data are transmitted to their CHs. After the CHs receive the data, the residual information is discarded through compression, and the aggregated data will be transmitted to the next hop or BS according to the

$$O_q(s,p)=\begin{cases}1, & \Delta(s,p)<(S_r-U_r)\\ & \theta(s,p)<(\theta_f-\theta_u)\\ & \phi(s,p)<(\phi_f-\phi_u)\\ & \text{and LOS}\\ e^{-\alpha\times dist^{\beta}}, & (S_r-U_r)<\Delta(s,p)<(S_r+U_r)\\ & \theta(s,p)<(\theta_f-\theta_u)\\ & \phi(s,p)<(\phi_f-\phi_u)\\ & \text{and LOS}\\ 1-\int_{-\infty}^{\theta(s,p)-\theta_f}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{t^2}{2\sigma^2}}dt, & \Delta(s,p)<(S_r-U_r)\\ & (\theta_f-\theta_u)<\theta(s,p)<(\theta_f+\theta_u)\\ & \phi(s,p)<(\phi_f-\phi_u)\\ & \text{and LOS}\\ 1-\int_{-\infty}^{\phi(s,p)-\phi_f}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{t^2}{2\sigma^2}}dt, & \Delta(s,p)<(S_r-U_r)\\ & \theta(s,p)<(\theta_f-\theta_u)\\ & (\phi_f-\phi_u)<\phi(s,p)<(\phi_f+\phi_u)\\ & \text{and LOS}\\ e^{-\alpha\times dist^{\beta}}\left(1-\int_{-\infty}^{\theta(s,p)-\theta_f}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{t^2}{2\sigma^2}}dt\right), & (S_r-U_r)<\Delta(s,p)<(S_r+U_r)\\ & (\theta_f-\theta_u)<\theta(s,p)<(\theta_f+\theta_u)\\ & \phi(s,p)<(\phi_f-\phi_u)\\ & \text{and LOS}\\ e^{-\alpha\times dist^{\beta}}\left(1-\int_{-\infty}^{\phi(s,p)-\phi_f}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{t^2}{2\sigma^2}}dt\right), & (S_r-U_r)<\Delta(s,p)<(S_r+U_r)\\ & \theta(s,p)<(\theta_f-\theta_u)\\ & (\phi_f-\phi_u)<\phi(s,p)<(\phi_f+\phi_u)\\ & \text{and LOS}\\ \left(1-\int_{-\infty}^{\theta(s,p)-\theta_f}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{t^2}{2\sigma^2}}dt\right)\left(1-\int_{-\infty}^{\phi(s,p)-\phi_f}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{t^2}{2\sigma^2}}dt\right), & \Delta(s,p)<(S_r-U_r)\\ & (\theta_f-\theta_u)<\theta(s,p)<(\theta_f+\theta_u)\\ & (\phi_f-\phi_u)<\phi(s,p)<(\phi_f+\phi_u)\\ & \text{and LOS}\\ e^{-\alpha\times dist^{\beta}}\left(1-\int_{-\infty}^{\theta(s,p)-\theta_f}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{t^2}{2\sigma^2}}dt\right)\left(1-\int_{-\infty}^{\phi(s,p)-\phi_f}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{t^2}{2\sigma^2}}dt\right), & (S_r-U_r)<\Delta(s,p)<(S_r+U_r)\\ & (\theta_f-\theta_u)<\theta(s,p)<(\theta_f+\theta_u)\\ & (\phi_f-\phi_u)<\phi(s,p)<(\phi_f+\phi_u)\\ & \text{and LOS}\\ 0, & \text{otherwise}\end{cases} \tag{26}$$

where S_r and U_r denote two distance ranges of sensor nodes, respectively, here, the radius of the deterministic sensing distance is (S_r-U_r), and the radius of the fuzzy ring is $2U_r$, that is, it locates in the ring between (S_r-U_r) and (S_r+U_r); θ_f and θ_u are two horizontal angle ranges, here, the radius of the horizontal deterministic sensing angle is $(\theta_f-\theta_u)$, and the radius of the fuzzy ring is $2\theta_u$, that is, the horizontal fuzzy ring region is between the angle ranges of $(\theta_f-\theta_u)$ and $(\theta_f+\theta_u)$; ϕ_f and ϕ_u are two vertical angle ranges, here, the radius of the deterministic sensing angle is $(\phi_f-\phi_u)$, and the radius of the fuzzy ring is $2\phi_u$, that is, the angle range between $(\phi_f-\phi_u)$ and $(\phi_f+\phi_u)$ is the vertical fuzzy ring region.

routing algorithm. The specific routing algorithm is detailed in Algorithm 1.

First, we cluster all sensor nodes by allocating each sensor node to its nearest CH, which can reduce the energy consumption of data transmission. Then the routing path will be determined. Specifically, for each CH, its distances to the BS and other CHs are calculated, and the CHs are sorted in descending order according to their distances to the BS. This sorting is important, because we check each CH from far to near, which is convenient for calculating the relayed data for each relay node. Each CH chooses the closest CH as its next hop from those that are nearer to the BS, thus preventing choosing the nearest one which is located farther from the BS and increasing the length of the relaying path.

3.3 Modified Differential Evolution Algorithm

We propose a modified DE algorithm [30] with CR-sort [31] and polynomial-based mutation by combining a novel CC strategy, denoted as CCDEXSPM. The work of [32] utilized the dynamic grouping method [38] into separate variables to several groups constantly and randomly. They used the optimizer of particle swarm optimization (PSO) [39], and a context vector (i.e., the global best solution) was maintained. Variables in the allocated group updated their values according to the velocity updating formula of PSO; while the remaining variables came from the personal bests or the global best according to several predefined thresholds, but the velocity updating formula was not used. Meanwhile, they utilized another PSO operator to update the context vector.

We combine this CC scheme with our DE strategy. According to the characteristics of the deployment problem, we adopt fixed grouping [29] to divide variables into several groups of unequal dimensions by separating variables of the same property into the same group, and each group is optimized in turn. Variables in the current optimization group are mutated, while other variables are the crossover result of the personal best and the global best.

The selection mechanism we utilized is to uniformly choose individuals from the current population for parents, ensuring a wider search direction, and avoiding being trapped into local optima and resulting in premature con-

Algorithm 1: Routing

Input: Coordinates of all sensor nodes and CHs; number of sensor nodes N_{DS}, number of CHs N_{CH}.
Output: Routing scheme.
Initialization;
for $i = 0$ *to* $N_{DS} - 1$ **do**
  for $j = 0$ *to* $N_{CH} - 1$ **do**
    Calculate the distance of i-th sensor node and j-th CH;
  end
  Choose the closest CH as its own CH;
end
for $j = 1$ *to* $N_{CH} - 1$ **do**
  Calculate the distance of j-th CH and BS;
  Calculate the distances of j-th CH and other CHs;
end
Sort the distances of all CHs with BS in descending order;
for $j = 1$ *to* $N_{CH} - 1$ **do**
  For each CH, from those CHs that are closer to the BS, choose the closest CH as its next hop, while for the CH closest to the BS, choose the BS;
end

vergence. This mutation is detailed in Eq. 27:

$$v_{i,j}^{g} = x_{r1,j}^{g} + F_i \times \left(x_{r2,j}^{g} - x_{r3,j}^{g}\right) \quad s.t.\ j \in S_{var}^{m} \tag{27}$$

where x_{r1}^{g}, x_{r2}^{g} and x_{r3}^{g} are randomly and uniformly selected from the g-th generation of population, here $1 \leq g \leq g_{max}$ and g_{max} is the maximum generation quantity, j is the variable. S_{var}^{m} is the set of variables to be optimized in subpopulation m, here $1 \leq m \leq M$ and M is the number of subpopulations. The scale factor F_i of each individual follows the updating scheme in JADE [40], which satisfies Cauchy distribution within the range of $[0, 1]$ with the position value of μ_F and the scale parameter of 0.1, as follows in Eq. 28.

$$F_i = CauchyRandom\left(\mu_F, 0.1\right) \tag{28}$$

where μ_F has an initial value of 0.5. Let S_F denote the set of all F values of individuals that are successfully mutated, then we can update μ_F as in Eq. 29:

$$\mu_F = (1 - c) \times \mu_F + c \times mean_L\left(S_F\right) \tag{29}$$

where $mean_L\left(\bullet\right)$ represents the Lehmer mean, as in Eq. 30:

$$mean_L\left(S_F\right) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F} \tag{30}$$

After the mutation of variables in the current group, we integrate the remaining variables. The remaining variables do not come directly from the global best, instead, each individual stores its personal best. Therefore, we randomly select variables from the global best and the personal best, that is, we conduct crossover between the global best and the personal best, as in Eq. 31:

$$v_{i,j}^{g} = \begin{cases} x_{i,j}^{g}, & r > CR_i \\ Best_j, & r \leq CR_i \end{cases} \quad s.t.\ j \notin S_{var}^{m} \tag{31}$$

$$r = rand() \tag{32}$$

The crossover rate is determined by CR. For the crossover factor CR, we adopt the updating strategy of CR-sort in [31]. The generation of CR satisfies Gaussian distribution, and after the initial population randomly generated, the fitness of individuals are calculated and ranked in order of best to worst, and the values of CR are sorted in increasing order. Then the better individual would be assigned the smaller value of CR, which helps maintain the better personal best.

To avoid the population from premature convergence, after one mutation of variables in the current group and the generation of the remaining variables from crossover, we add a polynomial-based mutation operator, executing a second-time mutation to the generated individual to improve the diversity of the population. Henceforth, one individual accomplishes its evolution process. As to the polynomial-based mutation operator, we will make detailed introduction in Subsection 3.4.

When each individual in each subpopulation finishes one evolution process, we consider the whole population completes one evolution. The whole population continuously repeats this evolutionary cycle until the termination condition is satisfied. The whole evolution process is detailed in Algorithm 2.

3.4 Polynomial-based Mutation

To increase the diversity of the population, avoid premature convergence and being trapped in local optima, after we have conducted mutation and crossover of each individual, a polynomial-based mutation operator is utilized to perturb variables selected by a certain probability to conduct a second-time mutation. The polynomial-based mutation perturbs the original value by exerting a small change, improving the diversity of the population [41]. This change value $v_{i,j}^{perm}$ can be represented as:

$$v_{i,j}^{perm} = \sigma_{i,j} \times B_j \tag{34}$$

where i denotes individual i, j represents variable j, and B_j represents the baseline value, which is generally set as (ub_j, lb_j), here ub_j and lb_j are the upper and lower boundaries of variable j, respectively, and $\sigma_{i,j}$ is the variation ratio, which can be calculated as follows:

$$\sigma_{i,j} = \begin{cases} \left[2u + (1 - 2u) \times \sigma_{i,j,1}^{n+1}\right]^{\frac{1}{n+1}} - 1, & \text{if } u \leq 0.5 \\ 1 - \left[2\left(1 - u\right) + (2u - 1) \times \sigma_{i,j,2}^{n+1}\right]^{\frac{1}{n+1}}, & \text{otherwise} \end{cases} \tag{35}$$

where u is a random number obeying uniform distribution, n denotes the mutation distribution index, and $\sigma_{i,j,1}$ and $\sigma_{i,j,2}$ are computed respectively as follows:

Algorithm 2: CCDEXSPM

Output: The final best vector: $Best$.
Initialize();
do
  for *all individuals in each subpopulation* **do**
    (1) Variables allocated to current subpopulation — mutation according to Eq. 27;
    (2) Remaining variables in current subpopulation — crossover according to Eq. 31;
    (3) Generated individual — polynomial-based mutation;
    (4) Selection:

$$x_i^{g+1} = \begin{cases} v_i^g, & v_i^g.cost > x_i^g.cost \\ x_i^g, & \text{otherwise} \end{cases} \tag{33}$$

    where x_i^g and x_i^{g+1} are the personal bests of the g-th and $(g+1)$-th generations, respectively;
    (5) Update F_i and CR_i;
    (6) Update the vector: $Best$;
  end
while *The termination condition is not met*;

$$\sigma_{i,j,1} = \frac{x_{i,j} - lb_j}{ub_j - lb_j} \tag{36}$$

$$\sigma_{i,j,2} = \frac{ub_j - x_{i,j}}{ub_j - lb_j} \tag{37}$$

where $x_{i,j}$ is the original value before mutation.

Finally, we can obtain the mutated value $x'_{i,j}$:

$$x'_{i,j} = x_{i,j} + v_{i,j}^{perm} \tag{38}$$

3.5 Parallelism Implementation

To improve the operation speed of the algorithm , we put forward a MPI-based distributed parallel algorithm. In designing the parallel strategy, we observed that the computation of QoC occupied a large proportion of the overall operation time, which includes multiple loops and the time complexity is $O(LEN \times WID \times DS)$, here LEN and WID represent the length and width of the terrain data, respectively. Thus, compared to the communication cost among processes, the computation cost of QoC is tremendous. Especially, when the terrain area expands and the number of sensor nodes increases (that is, the dimensionality of the feasible solution is enlarged), the corresponding time consumption will also increase rapidly. Considering these factors, we chiefly divide the computation process into multiple blocks and address them in parallel; then, the information transmission is completed through communication among processes. Specifically, it is to divide the data for computation according to the number of processes. The more the processes, the less the data to be processed in each block. The specific process is in Algorithm 3.

Algorithm 3: Parallelism Implementation

MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD,&rank);
MPI_Comm_size(MPI_COMM_WORLD,&size);
$block\_size = len/size$;
$begin\_row = rank \times block\_size$;
$end\_row = (rank+1) \times block\_size$;
MPI_Bcast();
for $a := begin\_row \rightarrow end\_row$ **do**
  for $b := 0 \rightarrow wid$ **do**
    Compute $local\_qoc$;
  end
end
MPI_Gather $local\_qoc$;
MPI_Finalize();
return cover();

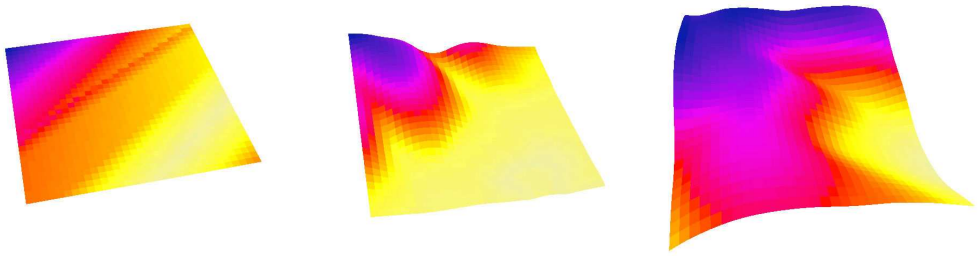
4 Experimental Results and Analysis

To assess the performance of the novel algorithm, we import real digital elevation model (DEM) raw data[1]. Then, we crop the DEM data to obtain three terrain data: mountainous, hilly and plain terrain data. Through resampling and pretreatment, we obtain three different type of terrain data with a size of $160m \times 160m$ and the resolution of $5m$. We will use these three types of 3D terrain data for experimentation. The plain, hilly, and mountainous terrain region that we extracted are illustrated in Fig. 4a, Fig. 4b, and Fig. 4c, respectively.

We implement MPI parallelism on the platform of Tianhe-2 supercomputer and conduct experiments on three terrains: plain, hilly and mountainous terrains. We compare the modified algorithm, CCDEXSPM, with MS-DE [42], jDE [43], GPSO [44], and CLPSO [45]. For the deployment problem, each algorithm repeats the operation 20 times. The parameter settings are listed in Table 1. We conduct experiments on the plain, hilly and mountainous terrains, and the results corresponding to nonparametric tests are shown in the following.

For the test results of the plain terrain, as shown in Table 9, our algorithm CCDEXSPM possesses superiority throughout the entire evolutionary process, and the mean fitness values have remained ahead, which gradually reach-

1. Geospatial Data Cloud, Computer Network Information Center, Chinese Academy of Sciences. (http://www.gscloud.cn).



(a) Plain Terrain (b) Hilly Terrain (c) Mountainous Terrain

Fig. 4. Three types of extracted terrains.

TABLE 1
Parameter Settings

Symbol	Quantity	Value
N_{DS}	Number of sensor nodes in plain	60
	Number of sensor nodes in hill	75
	Number of sensor nodes in mountain	90
N_{CH}	Number of CHs in plain	15
	Number of CHs in hill	20
	Number of CHs in mountain	25
NP	Population size of DE	100
	Population size of PSO	30
n_{FE}^{max}	Maximum number of function evaluations	$2.00E+06$
S_r	Predefined detection range	10 pixels [12]
U_r	Uncertain detection range	2 pixels [12]
θ_f	Predefined detection angle in horizontal direction	$\pi/4$ [12]
Φ_f	Predefined detection angle in vertical direction	$3pi/16$
θ_u	Uncertain detection angle	$0 \sim \pi/36$ [12]
α	Enviromental parameters	1
β	Enviromental parameters	0.5
λ	Fusion operator	-0.5 [33]
O_{th}	Sensing threshold	0.9 [33]
E_{rec}	Unit energy consumption for receiving data	$5.00E-08$
E_{agg}	Unit energy consumption for aggregating information	$5.00E-09$
ε_{fs}	Parameter for the free-space channel	$1.00E-11$
ε_{mp}	Parameter for the multi-path channel	$1.30E-15$
r_{cmp}	Data aggregation ratio	0.9
e	Initial energy of CHs	10
d_{th}	Channel selection threshold	87
l_0	Unit information size (bit)	200

es $8.57E-01$ gradually, far higher than the other four algorithms,and the maximum fitness value reaches $8.79E-01$. Similarly, from Fig. 5, we can observe that in the initial stage of evolution, GPSO has encountered premature convergence and is trapped in local optima, while CLPSO, jDE and MS-DE have poor convergence ability, although they have not converged so early, their exploration abilities are inferior to that of our algorithm, as their improvement of the fitness values are slow. In contrast, CCDEXSPM not only has converged early in the initial stage of evolution but also has a strong global search capability and updates the fitness value toward the global best solution. In the late stage of evolution, it performs the local search around the global best, and it finally achieves optimization results that are much better than those of the other four algorithms. For the Friedman test in Table 2, the rank of CCDEXSPM is ahead of the other four algorithms. Moreover, from results of the Wilcoxon test in Table 3, we find that all results achieved by CCDEXSPM are better than those of the other four algorithms ($Exact\ P-value$ is equal to $1.91E-06$ compares to CLPSO, GPSO, jDE, and $2.67E-05$ to MS-DE). In brief, the optimization performance of our algorithm is significantly better than that of GPSO, CLPSO, jDE and MS-DE.

Regarding the test results of the hilly terrain, as shown in Table 10 and Fig. 6, GPSO, CLPSO and jDE have similar performance compared to plain terrain, while MS-DE exhibits better performance, especially in the late stage of evolution. However, these four algorithms still show weak exploration ability (MS-DE performs better search ability) ; in contrast, our algorithm CCDEXSPM still exhibits strong exploration ability compared to the other four algorithms. For the Friedman test in Table 4, the rank of CCDEXSPM is still ahead of the other four algorithms, and the rank of GPSO is still the last one. From results of the Wilcoxon test in Table 5, we find that all results achieved by CCDEXSPM are

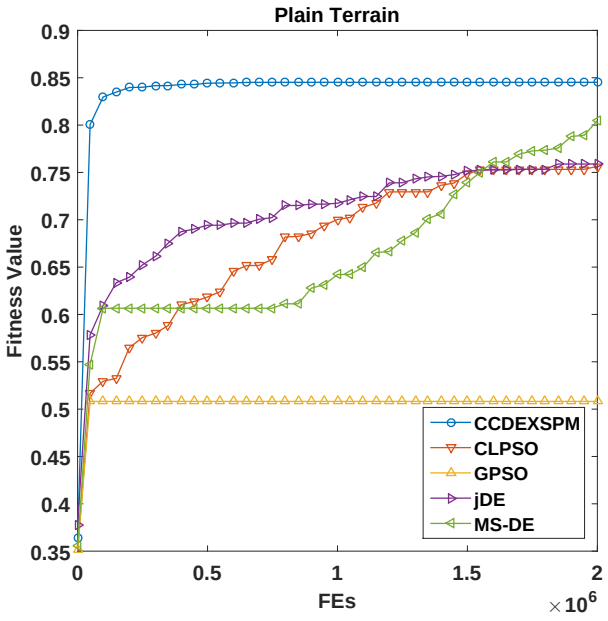


Fig. 5. Test results on plain terrain.

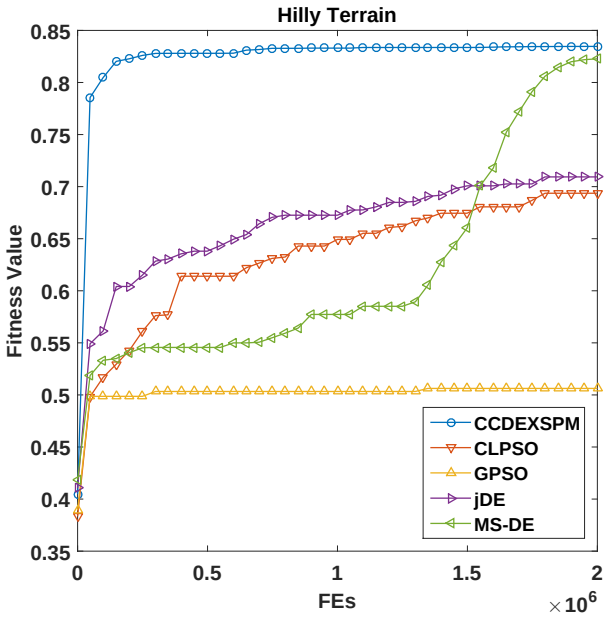


Fig. 6. Test results on hilly terrain.

TABLE 2
Friedman Test Results (Plain)

Algorithm	Ranking
CLPSO	3.1
GPSO	5
MS-DE	2.1
jDE	3.7
CCDEXSPM	1.1

TABLE 3
Wilcoxon Test Results (Plain)

VS	R+	R-	Exact P-value	Asymptotic P -value
CLPSO	210	0	1.91E-06	0.000082
GPSO	210	0	1.91E-06	0.000082
MS-DE	204	0	2.67E-05	0.000204
jDE	210	0	1.91E-06	0.000082

TABLE 4
Friedman Test Results (Hill)

Algorithm	Ranking
CLPSO	3.85
GPSO	5
MS-DE	2.1
jDE	3.05
CCDEXSPM	1

TABLE 5
Wilcoxon Test Results (Hill)

VS	R+	R-	Exact P-value	Asymptotic P -value
CLPSO	210	0	1.91E-06	0.000082
GPSO	210	0	1.91E-06	0.000082
MS-DE	210	0	1.91E-06	0.000082
jDE	210	0	1.91E-06	0.000082

still better than those of the other four algorithms ($Exact\ P-value$ is equal to $1.91E-06$). These indicate the effectiveness of CCDEXSPM in hilly terrain.

For the mountainous terrain, as shown in Table 11 that these five algorithms show similar performance to that corresponding to plain terrain. The overall fitness values have decreased to some extent (the mean fitness reaches 8.57E-01 in plain terrain, while it is only 7.86E-01 in mountainous terrain). This result is mainly because that the undulating terrain obstruct the LOS, and the coverage, lifetime, connectivity of nodes will be affected to varying degrees. However, CCDEXSPM has still achieved evident superiority. The Friedman test in Table 6 shows that the rank of CCDEXSPM is still ahead of the other four algorithms, and the results of Wilcoxon test in Table 7 also indicate the excellent optimization performance of our algorithm.

The experimental results prove that our algorithm exhibits better performance that the other four algorithms in three different terrains. The different terrains would affect the fitness values, and the obtained fitness values on the mountainous terrain are inferior to those on the plain terrain. However, our algorithm exhibits stable search ability in different terrain and obtains satisfactory results in fitness. To verify the effect of MPI parallelism on reducing the computational time, for the deployment problem on the plain terrain, we conduct experiments by utilizing CCDEXSPM in the cases of 1, 2, 4, 8 and 16 processes. The experimental results are listed in Table 8 and Fig. 8.

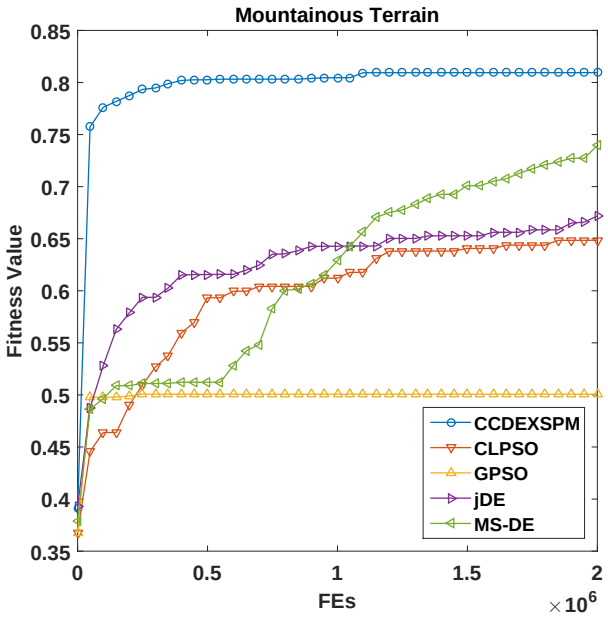


Fig. 7. Test results on mountainous terrain.

TABLE 6
Friedman Test Results (Mountain)

Algorithm	Ranking
CLPSO	3.7
GPSO	5
MS-DE	2.2
jDE	3.1
CCDEXSPM	1

TABLE 7
Wilcoxon Test Results (Mountain)

VS	R+	R-	Exact P-value	Asymptotic P -value
CLPSO	210	0	1.91E-06	0.000082
GPSO	210	0	1.91E-06	0.000082
MS-DE	210	0	1.91E-06	0.000082
jDE	210	0	1.91E-06	0.000082

TABLE 8
Operation Times

Process quantity	Average operation time (s)	Speedup ratio	Efficiency
1	21772.10823	1	1
2	11163.08569	1.950366	0.975183
4	6022.174545	3.615323	0.903831
8	3406.991099	6.390421	0.798803
16	1845.148276	11.799652	0.737478

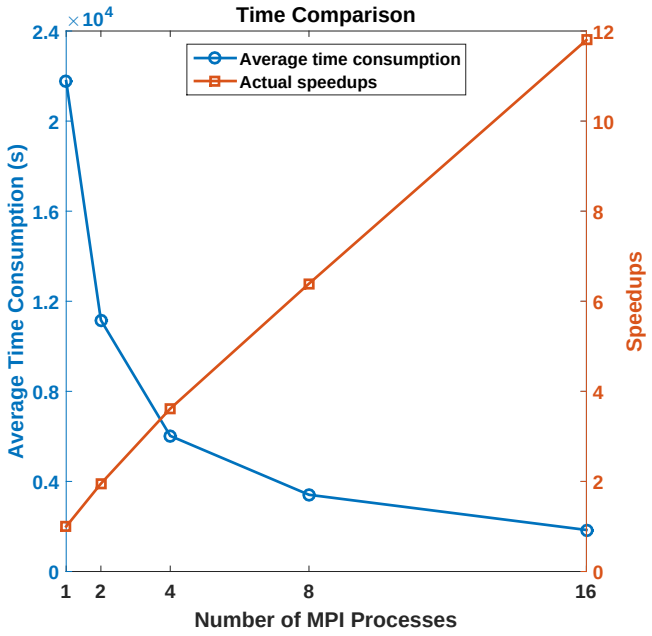


Fig. 8. Operation times.

As shown, as the number of processes increases, the overall operation time significantly decreases. When the number of processes is 16, the speedup ratio reaches approximately 12; however, with a greater number of processes, the use efficiency of each process is lower. When the number of processes is 16, the use efficiency of each process is reduced from approximately 0.97 for 2 processes to approximately 0.74. Therefore, if the number of processes further increases, computational resources will not be effectively explored due to the increased communication cost among MPI processes and the low use efficiency of each process. When the number of processes is 16, the operation time is effectively reduced, obtaining satisfactory results in operation time.

5 Conclusion

In this paper, we have primarily studied the deployment problem of DWSNs, we consider the coverage, lifetime, connectivity of sensor nodes, connectivity of CHs, and reliability in the deployment. We propose a modified DE algorithm with CR-sort and polynomial-based mutation by combining the CC framework, namely CCDEXSPM. We apply this algorithm to the deployment optimization problem of DWSNs, and compare it with GPSO, CLPSO, jDE and MS-DE. The experimental results show that the the performance of our modified DE algorithm is significantly better than that of these four algorithms. Additionally, we utilize MPI parallelism, effectively reducing the operation time.

For future research, there is still considerable work needs to be performed. We utilize weighted sum to combine the coverage rate, lifetime, connectivity of sensor nodes, connectivity of CHs, and reliability to a single objective function, while further research can be conducted by exploring multi-objective optimization.

TABLE 9
Algorithm Comparison Results (Plain)

FEs		CLPSO	GPSO	jDE	MSDE	CCDEXSPM
5.00E+05	MEAN	6.34E-01	5.12E-01	6.94E-01	6.10E-01	**8.53E-01**
	MAX	6.52E-01	5.51E-01	7.14E-01	6.91E-01	**8.74E-01**
	MIN	6.18E-01	4.66E-01	6.77E-01	5.80E-01	**8.42E-01**
	MEDIAN	6.35E-01	5.12E-01	6.93E-01	5.94E-01	**8.51E-01**
	STD	8.77E-03	2.22E-02	9.39E-03	3.34E-02	**8.25E-03**
1.00E+06	MEAN	7.06E-01	5.12E-01	7.27E-01	6.86E-01	**8.54E-01**
	MAX	7.34E-01	5.51E-01	7.41E-01	7.66E-01	**8.45E-01**
	MIN	6.94E-01	4.66E-01	7.16E-01	5.94E-01	**8.68E-01**
	MEDIAN	7.03E-01	5.12E-01	7.26E-01	6.75E-01	**8.54E-01**
	STD	1.06E-02	2.24E-02	**7.99E-03**	4.57E-02	8.22E-03
1.50E+06	MEAN	7.47E-01	5.12E-01	7.49E-01	7.59E-01	**8.56E-01**
	MAX	7.62E-01	5.51E-01	7.65E-01	8.34E-01	**8.78E-01**
	MIN	7.39E-01	4.66E-01	7.37E-01	6.56E-01	**8.45E-01**
	MEDIAN	7.47E-01	5.12E-01	7.48E-01	7.62E-01	**8.55E-01**
	STD	**5.87E-03**	2.24E-02	8.94E-03	4.66E-02	8.18E-03
2.00E+06	MEAN	7.69E-01	5.12E-01	7.61E-01	8.15E-01	**8.57E-01**
	MAX	7.84E-01	5.51E-01	7.72E-01	8.56E-01	**8.79E-01**
	MIN	7.55E-01	4.66E-01	7.50E-01	7.09E-01	**8.45E-01**
	MEDIAN	7.68E-01	5.12E-01	7.60E-01	8.13E-01	**8.56E-01**
	STD	8.41E-03	2.24E-02	**6.04E-03**	3.73E-02	8.29E-02

TABLE 10
Algorithm Comparison Results (Hill)

FEs		CLPSO	GPSO	jDE	MSDE	CCDEXSPM
5.00E+05	MEAN	6.04E-01	5.10E-01	6.52E-01	5.58E-01	**8.27E-01**
	MAX	6.18E-01	5.37E-01	6.67E-01	6.03E-01	**8.41E-01**
	MIN	5.90E-01	4.62E-01	6.38E-01	5.39E-01	**8.06E-01**
	MEDIAN	6.03E-01	5.09E-01	6.51E-01	5.52E-01	**8.26E-01**
	STD	8.23E-03	1.73E-02	**8.22E-03**	1.75E-02	8.27E-03
1.00E+06	MEAN	6.49E-01	5.11E-01	6.79E-01	6.24E-01	**8.30E-01**
	MAX	6.61E-01	5.46E-01	6.92E-01	7.49E-01	**8.43E-01**
	MIN	6.40E-01	4.62E-01	6.73E-01	5.50E-01	**8.10E-01**
	MEDIAN	6.48E-01	5.10E-01	6.78E-01	5.92E-01	**8.31E-01**
	STD	5.61E-03	1.81E-02	**5.44E-03**	7.03E-02	7.71E-03
1.50E+06	MEAN	6.76E-01	5.12E-01	6.95E-01	7.14E-01	**8.32E-01**
	MAX	6.86E-01	5.46E-01	7.09E-01	8.20E-01	**8.44E-01**
	MIN	6.66E-01	4.62E-01	6.85E-01	5.82E-01	**8.17E-01**
	MEDIAN	6.75E-01	5.10E-01	6.94E-01	7.15E-01	**8.32E-01**
	STD	**5.75E-03**	1.87E-02	6.14E-03	8.60E-02	7.06E-03
2.00E+06	MEAN	6.93E-01	5.14E-01	7.07E-01	7.95E-01	**8.33E-01**
	MAX	7.08E-01	5.46E-01	7.20E-01	8.28E-01	**8.45E-01**
	MIN	6.81E-01	4.62E-01	6.97E-01	6.19E-01	**8.18E-01**
	MEDIAN	6.94E-01	5.13E-01	7.07E-01	8.10E-01	**8.34E-01**
	STD	6.25E-03	1.85E-02	**5.74E-03**	4.75E-02	7.10E-03

TABLE 11
Algorithm Comparison Results (Mountain)

FEs		CLPSO	GPSO	jDE	MSDE	CCDEXSPM
5.00E+05	MEAN	5.20E-01	4.29E-01	5.57E-01	4.27E-01	**7.79E-01**
	MAX	5.35E-01	4.85E-01	5.71E-01	4.47E-01	**8.18E-01**
	MIN	5.06E-01	3.85E-01	5.40E-01	4.14E-01	**7.59E-01**
	MEDIAN	5.18E-01	4.28E-01	5.56E-01	4.26E-01	**7.76E-01**
	STD	8.00E-03	2.51E-02	8.45E-03	**7.78E-03**	1.49E-02
1.00E+06	MEAN	5.65E-01	4.30E-01	5.87E-01	4.35E-01	**7.83E-01**
	MAX	5.75E-01	4.85E-01	6.04E-01	4.55E-01	**8.19E-01**
	MIN	5.50E-01	3.85E-01	5.73E-01	4.22E-01	**7.64E-01**
	MEDIAN	5.67E-01	4.30E-01	5.85E-01	4.32E-01	**7.80E-01**
	STD	**6.96E-03**	2.54E-02	8.33E-03	9.20E-03	1.43E-02
1.50E+06	MEAN	5.95E-01	4.30E-01	6.04E-01	4.44E-01	**7.85E-01**
	MAX	6.09E-01	4.85E-01	6.19E-01	4.94E-01	**8.21E-01**
	MIN	5.83E-01	3.85E-01	5.94E-01	4.22E-01	**7.64E-01**
	MEDIAN	5.95E-01	4.30E-01	6.03E-01	4.39E-01	**7.82E-01**
	STD	7.46E-03	2.60E-02	**7.38E-03**	1.90E-02	1.46E-02
2.00E+06	MEAN	6.12E-01	4.33E-01	6.16E-01	4.73E-01	**7.86E-01**
	MAX	6.25E-01	4.85E-01	6.26E-01	6.85E-01	**8.23E-01**
	MIN	6.00E-01	3.85E-01	6.03E-01	4.29E-01	**7.64E-01**
	MEDIAN	6.13E-01	4.30E-01	6.18E-01	4.44E-01	**7.82E-01**
	STD	8.13E-03	2.48E-02	**6.70E-03**	7.13E-02	1.45E-02

REFERENCES

[1] M. Chernyshev, Z. Baig, O. Bello, and S. Zeadally, "Internet of things (iot): Research, simulators, and testbeds," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[2] K. Wu, Y. Gao, F. Li, and Y. Xiao, "Lightweight deployment-aware scheduling for wireless sensor networks," *Mobile Networks and Applications*, vol. 10, no. 6, pp. 837–852, Dec 2005.

[3] G. Jia, G. Han, H. Rao, and L. Shu, "Edge computing-based intelligent manhole cover management system for smart cities," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[4] E. Aguirre, P. Lopez-Iturri, L. Azpilicueta, A. Redondo, J. J. Astrain, J. Villadangos, A. Bahillo, A. Perallos, and F. Falcone, "Design and implementation of context aware applications with wireless sensor network support in urban train transportation environments," *IEEE Sensors Journal*, vol. 17, no. 1, pp. 169–178, Jan 2017.

[5] C. Fosalau, C. Zet, and D. Petrisor, "Implementation of a landslide monitoring system as a wireless sensor network," in *2016 IEEE 7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, Oct 2016, pp. 1–6.

[6] S. Egea, A. Rego, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Intelligent iot traffic classification using novel search strategy for fast based-correlation feature selection in industrial environments," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[7] O. M. Alia and A. Al-Ajouri, "Maximizing wireless sensor network coverage with minimum cost using harmony search algorithm," *IEEE Sensors Journal*, vol. 17, no. 3, pp. 882–896, Feb 2017.

[8] Manju, S. Chand, and B. Kumar, "Maximising network lifetime for target coverage problem in wireless sensor networks," *IET Wireless Sensor Systems*, vol. 6, no. 6, pp. 192–197, 2016.

[9] E. Tuba, M. Tuba, and D. Simian, "Wireless sensor network coverage problem using modified fireworks algorithm," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Sept 2016, pp. 696–701.

[10] Y. Tan, *Fireworks Algorithm: A Novel Swarm Intelligence Optimization Method*, 1st ed. Springer Publishing Company, Incorporated, 2015.

[11] H. Ma and Y. Liu, *On Coverage Problems of Directional Sensor Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 721–731.

[12] H. Teng, C. D. Wu, Y. Z. Zhang, and N. Hu, "Design of probabilistic sensing model for directional sensor node," *Journal of Jiangnan University (Natural Science Edition)*, vol. 11, no. 4, pp. 391–395, 2012.

[13] T.-W. Sung and C.-S. Yang, "Voronoi-based coverage improvement approach for wireless directional sensor networks," *Journal of Network and Computer Applications*, vol. 39, no. Supplement C, pp. 202 – 213, 2014.

[14] B. Cao, J. Zhao, Z. Lv, X. Liu, X. Kang, and S. Yang, "Deployment optimization for 3d industrial wireless sensor networks based on particle swarm optimizers with distributed parallelism," *Journal of Network and Computer Applications*, vol. 103, no. Supplement C, pp. 225 – 238, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517302722

[15] P. Kuila and P. K. Jana, "Energy efficient clustering and routing algorithms for wireless sensor networks: Particle swarm optimization approach," *Engineering Applications of Artificial Intelligence*, vol. 33, no. Supplement C, pp. 127 – 140, 2014.

[16] S. Halder and S. D. Bit, "Enhancement of wireless sensor network lifetime by deploying heterogeneous nodes," *Journal of Network and Computer Applications*, vol. 38, no. Supplement C, pp. 106 – 124, 2014.

[17] X. Chu and H. Sethu, "Cooperative topology control with adaptation for improved lifetime in wireless ad hoc networks," in *2012 Proceedings IEEE INFOCOM*, March 2012, pp. 262–270.

[18] G. Hacioglu, V. F. A. Kand, and E. Sesli, "Multi objective clustering for wireless sensor networks," *Expert Syst. Appl.*, vol. 59, no. C, pp. 86–100, Oct. 2016.

[19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr 2002.

[20] J. Li and M. Chen, "Multiobjective topology optimization based on mapping matrix and nsga-ii for switched industrial internet of things," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1235–1245, Dec 2016.

[21] L. Wang, X. Fu, J. Fang, H. Wang, and M. Fei, "Optimal node placement in industrial wireless sensor networks using adaptive mutation probability binary particle swarm optimization algorithm," in *2011 Seventh International Conference on Natural Computation*, vol. 4, July 2011, pp. 2199–2203.

[22] Y. Li, Y. Q. Song, Y. h. Zhu, and R. Schott, "Deploying wireless sensors for differentiated coverage and probabilistic connectivity," in *2010 IEEE Wireless Communication and Networking Conference*, April 2010, pp. 1–6.

[23] Z. Khalfallah, I. Fajjari, N. Aitsaadi, P. Rubin, and G. Pujolle, "A novel 3D underwater WSN deployment strategy for full-coverage

and connectivity in rivers," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–7.

[24] B. Cao, J. Zhao, P. Yang, Z. Lv, X. Liu, X. Kang, S. Yang, K. Kang, and A. Anvari-Moghaddam, "Distributed parallel cooperative coevolutionary multi-objective large-scale immune algorithm for deployment of wireless sensor networks," *Future Generation Computer Systems*, pp. –, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17313523

[25] B. Cao, J. Zhao, Z. Lv, and X. Liu, "3d terrain multiobjective deployment optimization of heterogeneous directional sensor networks in security monitoring," *IEEE Transactions on Big Data*, vol. PP, no. 99, pp. 1–1, 2017.

[26] X. Li, J. Peng, J. Niu, F. Wu, J. Liao, and K. K. R. Choo, "A robust and energy efficient authentication protocol for industrial internet of things," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[27] D. S. Deif and Y. Gadallah, "An ant colony optimization approach for the deployment of reliable wireless sensor networks," *IEEE Access*, vol. 5, pp. 10 744–10 756, 2017.

[28] R. Machado and S. Tekinay, "Diffusion-based approach to deploying wireless sensors to satisfy coverage, connectivity and reliability," in *2007 Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking Services (MobiQuitous)*, Aug 2007, pp. 1–8.

[29] M. A. Potter and K. A. De Jong, *A cooperative coevolutionary approach to function optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 249–257.

[30] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec 1997.

[31] Y. Z. Zhou, W. C. Yi, L. Gao, and X. Y. Li, "Adaptive differential evolution with sorting crossover rate for continuous optimization problems," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2742–2753, Sept 2017.

[32] B. Cao, W. Li, J. Zhao, S. Yang, X. Kang, Y. Ling, and Z. Lv, "Spark-based parallel cooperative co-evolution particle swarm optimization algorithm," in *2016 IEEE International Conference on Web Services (ICWS)*, June 2016, pp. 570–577.

[33] R. Wang, W. G. Wan, and X. Z. Wang, "Non-additive collaborative information coverage for cellular-model deployment in sensor networks," in *IET International Communication Conference on Wireless Mobile and Computing (CCWMC 2009)*, Dec 2009, pp. 49–52.

[34] R. Wang, W. Cao, and W. Xie, "Fuzzy coverage for sensor networks," *Chinese Journal of Scientific Instrument*, vol. 30, no. 5, pp. 954–959, 2009.

[35] J. Jia, *Coverage Control and Node Deployment Technologies in Wireless Sensor Networks*. Shenyang: Northeastern University Press, 2013.

[36] L. Wang, X. Fu, J. Fang, H. Wang, and M. Fei, "Optimal node placement in industrial wireless sensor networks using adaptive mutation probability binary particle swarm optimization algorithm," in *2011 Seventh International Conference on Natural Computation*, vol. 4, July 2011, pp. 2199–2203.

[37] H. Teng, C. D. Wu, Y. Z. Zhang, and N. Hu, "Design of probabilistic sensing model for directional sensor node," *Journal of Jiangnan University (Natural Science Edition)*, vol. 11, no. 4, pp. 391–395, 2012.

[38] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 210–224, 2012.

[39] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[40] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, Oct 2009.

[41] Q. Lin, J. Chen, Z. H. Zhan, W. N. Chen, C. A. C. Coello, Y. Yin, C. M. Lin, and J. Zhang, "A hybrid evolutionary immune algorithm for multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 711–729, Oct 2016.

[42] J. Wang, J. Liao, Y. Zhou, and Y. Cai, "Differential evolution enhanced with multiobjective sorting-based mutation operators," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2792–2805, Dec 2014.

[43] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, Dec 2006.

[44] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, May 1998, pp. 69–73.

[45] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, June 2006.