Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Wireless Sensor Network Deployment of 3D Surface Based on Enhanced Grey Wolf Optimizer

ZHENDONG WANG, (Member, IEEE), HUAMAO XIE
College of Information Engineering, Jiangxi University of Science and Technology, Ganzhou Jiangxi 341000 China

Corresponding author: Huamao Xie (fhzm1995@163.com).

This work was supported by the National Natural Science Foundation of China (61562037, 61562038, 61563019, 61763017), the Natural Science Foundation of Jiangxi Province (20171BAB202026, 20181BBE58018), Science and Technology Project Founded by the Education Department of Jiangxi Province (GJJ150643, GJJ190460), Innovation Designated Fund for Graduate Student of Jiangxi Province (YC2018-S331).

ABSTRACT Aiming at the difficulty of deploying wireless sensor networks (WSNs) on three-dimensional (3D) surfaces, based on the grey wolf optimizer (GWO), an enhanced version of the grey wolf optimizer is proposed for deploying WSNs on 3D surfaces, namely the enhanced grey wolf optimizer (EGWO), which is characterized by enhanced exploitation and exploration ability of the algorithm. The novelty of EGWO is that the grey wolf population is divided into two parts, one part is responsible for the outer-layer encircle and the other is responsible for the inner-layer encircle, and the introduction of Tent mapping. The purpose of this is to enhance the exploitation and exploration ability of the algorithm respectively, so as to improve the convergence and optimization precision of the algorithm. In addition, in terms of WSN deployment in 3D surfaces, this paper improves the means of determining the perceived blind zone. Meanwhile, a novel method to calculate the WSNs coverage area of simple and complex 3D surfaces is presented by combining the grid and integral of the 3D surfaces. The EGWO is favorably compared with the GWO and three existing variants of the grey wolf optimizer when testing on 12 well-known benchmark functions. The simulation experiment results show that compared with the existing algorithms, EGWO can provide a very competitive search result in terms of optimization precision and convergence performance. Finally, this paper applies EGWO to the 3D surface deployment of WSN. Simulations show that compared with the other three deployment algorithms, EGWO can improve the network coverage of WSN, which can save network deployment costs. In addition, the probability of network connectivity deployed by EGWO is higher, that is, EGWO can provide a better deployment solution.

INDEX TERMS Wireless sensor network, Three-dimensional surface deployment, perceived blind zone, Grid and Integration, Optimization, Grey wolf optimizer, Inner-layer encircle

I. INTRODUCTION

With the development of 5G communication technology and the Internet of Things, wireless sensor networks (WSNs) have been widely used in the fields of medical health, smart home, underwater monitoring, and military operations [1-4]. In recent years, researchers have made remarkable research progress in routing protocols, positioning, and coverage in WSN [5-9], where coverage optimization is a key issue for WSNs because it directly affects Network Quality of Service (QoS) [10]. WSN deployment in a two-dimensional (2D) plane, which is a simplified version of three-dimensional (3D) deployment, has been studied extensively [11-13]. On the other hand, 3D deployment of WSNs is divided into two types, one is deployed in a 3D space [4,9], and the other is deployed on a surface [16-18]. There are less research works, but increasing research interest, about the deployment of WSNs on 3D surfaces due partly to its complexity and usefulness in practical applications. Some scholars have applied Voronoi diagrams commonly used in 2D environment deployment to 3D deployment [19,20], and have obtained good deployment results. The recent works in [21,22] present two self-deployed algorithms for the problem of deploying sensor nodes on surfaces. The widely used virtual force algorithm in 2D deployment has also been well applied in 3D surface deployment [22-25]. With the outstanding effect of swarm intelligence optimization algorithms on optimization problems in recent years, this class of algorithms have all achieved satisfactory deployment outcomes when applied in 3D deployments [26-29]. In addition, some in-depth, more quantitative research works have been conducted, e.g., on target coverage [29,30].

The Grey Wolf Optimizer (GWO), a swarm intelligence optimization algorithm that mimics the predation process of grey wolves, is a heuristic search algorithm proposed by Mirjalili et al. [31]. Because the algorithm has good convergence ability and high optimization precision, it has been widely used in neural network training [32,33], controller design [34,35], wireless sensor network coverage [36,37] and other research fields [38,39,40]. The optimization performance of the GWO algorithm, however, still has room for improvement. Aiming at the problems of slow convergence of GWO and insufficient precision of optimization, various improvement strategies have been proposed. Researchers mainly study how to strengthen the exploitation or exploration of GWO, and balance the exploitation and exploration of GWO. For instance, the algorithm performance has been improved by modifying the convergence factor [41,42]. The framework of the GWO is modified, or the position update strategy of the grey wolf is changed [43-46] for achieving good search results. In order to obtain good optimization precision, some scholars have proposed a hybrid grey wolf optimizer algorithm [47,48] that combines other optimization algorithms with the grey wolf optimizer algorithm, but new algorithms are often computationally complex. In addition, operations such as chaotic mapping, crossover, and mutation [49,50] also play a role in GWO variants.

In terms of deploying wireless sensor networks on 3D surfaces, swarm intelligence optimization algorithms have demonstrated the advantages of simplicity, ease of use, and no need of special modeling [27], compared with other deployment strategies. This paper proposes a method based on GWO to integrate wireless sensor networks deployed on 3D surfaces. The main novelties of this paper are summarized as follows.

1) Given that the performance of GWO can still be improved, this paper proposes an enhanced grey wolf optimizer (EGWO). We have added a new grey wolf position update strategy, which uses a mixture of inner and outer layers to surround the prey to improve the ability of the grey wolf to hunt for prey. That is, the original population is divided into two parts, with half of the population executing the original outer envelope command, and the other half executing the inner encircle strategy. The role of the inner-layer encircle strategy is to strengthen the exploitation ability of EGWO, while the outer-layer encircle mechanism and Tent mapping can slightly strengthen the exploration ability of EGWO. In addition, EGWO's inner-layer encircle strategy can promote the connectivity of the WSN to a certain extent, which increases the possibility of WSN connectivity. In order to verify the effectiveness and superiority of this strategy, we use the improved grey wolf optimizer algorithms in [42], [45], and [46] as the benchmark algorithms. All algorithms are tested on 12 well-known benchmark functions, and their advantages and disadvantages are analyzed.
2) To our knowledge, this work is the first to apply the improved grey wolf optimizer to 3D surface deployment. In order to verify that the proposed algorithm has strong adaptability, the complex surface deployment scenarios are set up. Meanwhile, in order to increase the possibility of network connectivity in a simple surface deployment scenario, some nodes are fixed at a certain height from the surface.
3) Sun et al. [29] proposed a method of judging the perceptual blind zone by mathematical geometry, which can be roughly summarized as solving the linear equation and the 3D surface equation simultaneously, then judging the perception by the number of solutions. However, this method does not consider the situation when the nodes are all on the surface, a case which is introduced in this work.
4) The grid method commonly used for deploying sensor nodes in a 2D environment cannot be directly applied to the deployment of 3D surfaces. Therefore, the grid method in [51] is refined to adapt to the deployment of 3D environments. At the same time, a method is proposed to calculate the coverage area by combining the grid and the surface integral.
5) In order to verify the effectiveness of EGWO deploying sensor networks on 3D surfaces, our work will also be compared with two existing 3D deployment methods, i.e., ABC-DSS (artificial bee colony algorithm with dynamic search strategy) in [28] and DEA (differential evolution algorithm) in [29].

The reminder of this paper is organized as follows. Some related works are introduced in Section II. The relevant model of WSN deployment and the problem to be solved and optimized are introduced in Section III. In Section IV, the basic grey wolf optimizer is described. Section V presents the EGWO algorithm. The simulation experiments and analysis are presented in Section VI. Finally, a brief conclusion and some future works are drawn in Section VII.

II. RELATED WORK

The main content of this paper is divided into two parts. One is the 3D surface deployment of WSN, and the other is the improved grey wolf optimizer. We therefore briefly discuss each circumstance in turn.

In terms of the deployment of WSN, researchers have also achieved fruitful results. In [11], Deng et al. combined the virtual spring force algorithm and hexagonal topology to deploy WSN in a 2D plane. This method improves the energy consumption of the entire sensor network and can be used in large-scale WSN. In [9] Han et al. proposed a method of deploying sensor nodes under a 3D space underwater for underwater surveillance and marine research, the author believes that the method can also be used in other deployment environments. In [19] Anand et al. proposed a deployment technology suitable for 3D terrain. The traditional Voronoi diagram algorithm was modified to partition the terrain, their simulation results verified the effectiveness of the algorithm. Nazarzehi and Savkin [22] proposed a 3D self-deployment algorithm (3DSD) which can satisfy the autonomous deployment of nodes with obstacles in 3D space. Simulation results show that the deployment process of 3DSD is relatively fast, with the nodes evenly distributed, and the coverage of 3DSD close to the theoretical maximum. Boufares et al. [24] proposed a 3D-DVFA-FSC (3D Distributed Virtual Forces Algorithm for Flat Surface Covering) to cover a complete 3D plane and maintain network connectivity. Simulation results show that 3D-DVFA-FSC can provide full coverage deployment regardless of the 3D planar shape, while maintaining network connectivity. However, the situation of blind spots is not considered in this article, and the deployment method is slow to deploy. Yang et al. [28] proposed an ABC-DSS (artificial bee colony algorithm with dynamic search strategy) to deploy 3D surfaces with limited sensor nodes. This algorithm matches the sensor deployment problems on 3D surface well. Simulation experiments show that its deployment coverage is higher than particle swarm optimization (PSO) and artificial bee colony algorithm (ABC), but this article does not consider the most basic connectivity issues in WSN. In [29], in order to solve the problem of target coverage in 3D surface, Sun and coauthors innovatively proposed the deployment strategy of the target points, using the differential evolution algorithm (DEA) to optimize the position of the nodes to achieve the coverage of the target points. In addition, the authors proposed the sensor blind spots in the process of detecting target points on the 3D surface, and the corresponding solutions. Simulation experiments have proved the effectiveness of DEA for 3D surface deployment. However, in terms of judging blind spots, the authors did not consider the situation where the target node and the detection node are on the surface.

The grey wolf optimizer (GWO) was proposed in 2014 [31], which has good optimization performance, and is widely used in many fields. In [33], to predict the spun yarns tensile strength, Hadavandi et al. presents a neural network simulator based on GWO, called "GWNN", for prediction of siro-spun yarn tensile strength. The role of GWO is to determine the value of multi-layer perceptron weights through global search. In the experimental part, compared with several other prediction models, it is verified that the prediction accuracy of GWNN is higher. In [35], Komathi et al. proposed the design of GWO algorithm based fractional order proportional integral (FOPI) controller. The role of GWO is to obtain the optimal parameters of the FOPI controller to improve the response speed of the system. Simulation with Matlab / Simulink software and comparison with other optimization techniques show that the controller based on GWO has advantages in stability, speed and accuracy in applications such as robot power supply. In [36], the author researched routing protocols and proposed a new clustering algorithm to solve the problem of energy consumption in WSN. In order to improve energy efficiency, GWO is used to select cluster head nodes, and the protocol algorithm is allowed to use the same cluster in multiple consecutive rounds. In addition, the author also provides a new two-hop routing algorithm for cluster head nodes far from the base station. In the experimental part, the authors compared the routing protocol algorithm with several other similar algorithms in terms of energy consumption and proved that the proposed routing algorithm can improve network lifetime. In [38], Cai applied GWO to influence maximization in social networks. The influence is first formulated into an optimization problem, and then the optimization problem is solved with GWO. Experiments show that GWO has better performance than the latest influence maximization algorithms, and has less computation time than other meta-heuristic methods.

Given that GWO is a useful algorithm, many improved variants of GWO have been proposed. In [42], Mittal et al. Changed the convergence factor to balance exploitation and exploration of GWO, and called it mGWO (modified GWO). In the benchmark function test, mGWO can improve the precision of the solution compared to GWO, but its convergence has not improved significantly. In the practical application of the WSN clustering problem, the stability and effectiveness of mGWO are demonstrated. In [43], Jayabarathi et al. proposed a hybrid grey wolf optimizer, which incorporates the mutation and crossover operations in the genetic algorithm into the GWO to enhance the ability of exploration, and obtained a good scheduling scheme on the economic scheduling problem. Long et al. [45] proposed the EEGWO (exploration-enhanced GWO) which not only improved the convergence factor of GWO, but also changed the position update equation. The purpose is to strengthen GWO's exploration ability. Their experiments verified that EEGWO has outstanding effects in multiple engineering applications. Once again, Long et al. [46] incorporated random opposition learning strategy into GWO (ROLGWO) to strengthen the exploration and exploitation ability of GWO, thereby improving the optimization precision and convergence performance of GWO. In 23 benchmark function test experiments, the experimental results verify the effectiveness of the strategy, but a minor disadvantage is that the type of the benchmark function is not rich enough, and the benchmark function whose optimal position is non-zero has not been tested. Simulation experiments of pressure vessel design and welded beam design show that ROLGWO's optimization effect is better than other algorithms. Aiming at the difficulty to deal with the optimal power flow problem, reference [48] proposed a hybrid algorithm based on enhance grey wolf optimizer and dragonfly algorithm (EGWO-DA).

The new algorithm combines the advantages of the grey wolf optimizer algorithm in local search in terms of exploitation ability and advantages of the dragonfly algorithm in global search in terms of exploration ability. Simulation experiments were performed on the IEEE 30 bus system. Experimental results prove that EGWO-DA is more effective than other algorithms in reducing costs and minimizing power consumption, but because it needs many iterative steps, its computing is complicated. In [49], Zhao et al. added chaos mapping to the population initialization of GWO to enrich the diversity of the population, thereby strengthening the global exploration ability of the algorithm, and named it CEGWO (the chaos enhanced grey wolf optimization). They then applied it to extreme learning machines (ELM) to identify patients with paraquat poisoning. The role of CEGWO is to search the optimal feature set to improve the recognition accuracy of ELM. Experiments have shown that CEGWO can promote the identification of poisoned paraquat patients with extreme learning machines.

III. MODEL AND COVERAGE OPTIMIZATION PROBLEM DESCRIPTION

In this section, we will introduce the WSN deployment model, which involves node attributes, coverage calculation, judgment methods of network connectivity, judgment methods of perceived blind spot, and goals of network deployment.

A. COVERAGE DESCRIPTION

In a WSN, the perception and communication radius of a sensor node are denoted as r^p and R^c respectively. To ensure the connectivity of the wireless sensor network, the communication radius of the node is greater than or equal to twice the perception radius of the node. When a WSN is deployed in a 2D plane, the coverage is the area of the plane, and in the 3D surface, the coverage is the area of the surface. In order to facilitate the calculation of the area covered, we use the boolean perception model and the grid method that are commonly used in 2D deployment. The monitoring point is at the center of each grid (small surface). Due to the uneven 3D surface and many slopes, the 3D grid method and the calculation method of node perception need to be improved. We will describe our approach in the next section. Suppose a WSN comprises a set of sensor nodes $\mathbf{S} = \{s_1, s_2, s_3, \ldots, s_n\}$, and a set of monitoring points $\mathbf{M} = \{m_1, m_2, m_3, \ldots, m_n\}$, (x_i, y_i, z_i) and (x_j, y_j, z_j) correspond to the 3D space coordinates of s_i and m_j in the set, respectively. Then, the Euclidean distance between the node and the monitoring point is:

$$d(s_i, m_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (1)$$

The probability that the monitoring point m_j is perceived by the node s_i is:

$$p_{cov}(s_i, m_j) = \begin{cases} 1 & if\ d(s_i, m_j) \le r^p \\ 0 & otherwise \end{cases} \quad (2)$$

The joint perception probability of all sensor nodes to point m_j is:

$$C_p(s_{all}, m_j) = 1 - \prod_{i=1}^{n} \left(1 - p_{cov}(s_i, m_j)\right), \quad (3)$$

where s_{all} represents all sensor nodes in the monitoring range, and L and W are the length and width of the rectangle formed when the surface (H) is vertically mapped to the two-dimensional (2D) plane. If the value of Eq. (3) is 1, it indicates that the small surface where the j-th monitoring point is located is covered by the sensor node, and the surface (H) area covered by WSN is the sum of the small surface area of all the detected monitoring points. The small surface area set is $\boldsymbol{S^b} = \{s_1^b 、 s_2^b 、 s_3^b, \ldots, s_n^b\}$, and surface area S_a of surface H is the sum of the surface area of all small surfaces. Each small surface area can be calculated by the surface integral formula, and the coverage ratio C_r can be expressed as follow:

$$C_r = \sum_{x=1}^{L} \sum_{y=1}^{W} C_p\left(s_{all}, m_{(x-1)\cdot W+y}\right) \times s^b_{(x-1)\cdot W+y} \Big/ S_a \quad (4)$$

B. GRID METHOD DESCRIPTION

In the 2D plane deployment, we can divide many small grids with equal area with equal spacing between x and y axes, and place a monitoring point in the center of each grid. However, in the surface deployment, it is difficult to divide the whole surface into small surfaces with the same area of each grid due to the winding and undulating surface. In fact, it is not necessary to divide the surface into grids with equal area. Therefore, our strategy can be described as follows.

1. Vertically map the entire 3D surface onto a 2D plane, and then divide the 2D plane into an equal-area grid. As shown in Fig. 1 (a).

2. As shown in Fig. 1 (b), the center point o of the small grid $efgh$ is marked, and then the small grid $efgh$ and the center point o of the grid in the 2D plane are mapped back to the tangent plane in the 3D space. In this step, the tangent plane $e'f'g'h'$ and the tangent plane center $o'$are obtained.

3. As shown in Fig. 1 (c), the small tangent plane $e'f'g'h'$ and the center point o' of the tangent plane are inversely mapped onto the surface H, and the surface H' and the surface center o'' are obtained. We think of small surfaces approximately as small tangent planes (the more grids are divided, the closer the surface area is to the tangent plane). By doing this, the approximate mapping between the 2D plane and the 3D surface is completed. Then a monitoring point is marked at point o'' and the surface area of each small surface is calculated.

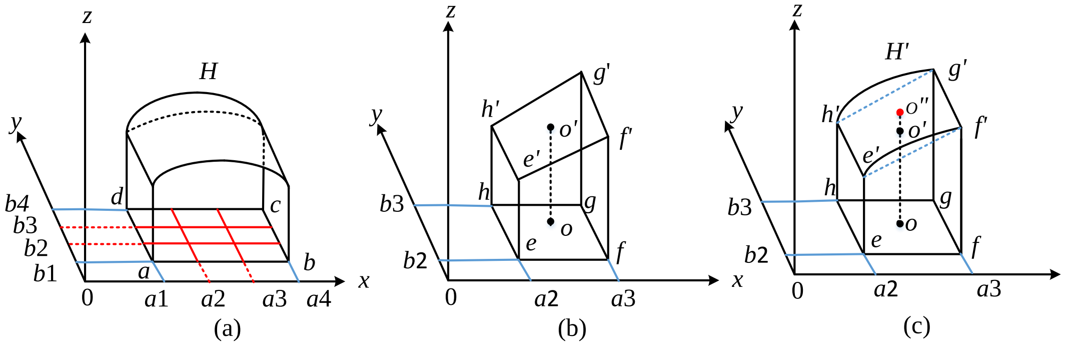


FIGURE 1. **Process diagram of grid method**

C. SURFACE AREA CALCULATION

The grid method divides the surface into a number of continuous small surfaces. Next, the surface integral formula is used to calculate the surface area of each small surface. The sum of which is the total area of the surface. Assuming that the surface is expressed by Eq. (5), then the surface area of the surface can be obtained by surface integral formula given by Eq. (6), and the range of D_{xy} is composed of $[x_1, x_2]$ and $[y_1, y_2]$.

$$z = f(x, y) \tag{5}$$

$$A = \iint_{D_{xy}} \sqrt{1 + \left(\frac{dz}{dx}\right)^2 + \left(\frac{dz}{dy}\right)^2}\, dxdy \tag{6}$$

D. 3D PERCEPTION BLIND ZONE

When a WSN is deployed on the surface, the boolean perception model in the 3D deployment is insufficient. The curve in Fig. 2 is a side view of the surface, and the circle is the theoretical sensing range of the wireless sensor node S. It can be seen from Eq. (2) that the monitoring points at points A, B, and C are within the sensing range of the node S. However, it should be noted that since the wave signal is propagating in a straight line, the wave signals transmitted to the points B and C are blocked by the slope where the point A is located, so the points B and C cannot be perceived by the node S. We call such a region (point) a 3D perceived blind zone (spot). It should be added that node S in the graph is on the surface, but, in our experiment, we have a few nodes placed at a certain height from the surface, which is a measure to connect the whole network.

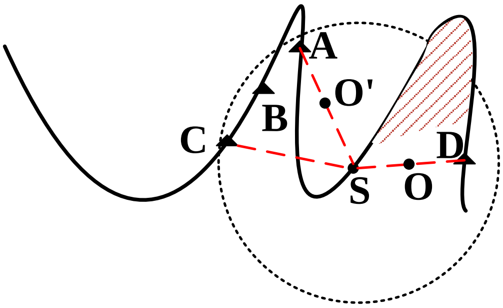


FIGURE 2. **3D perception blind zone**

In order to introduce the determination of the next part of the 3D perceived blind spot, we must first introduce the linear equation in 3D space. If a non-zero vector $\vec{S}$ is parallel to a known straight line L_1, the vector $\vec{S}$ is called the direction vector of the straight line L_1. The straight line L_1 passes through the point $M_1(x_1, y_1, z_1)$, $M_2(x_2, y_2, z_2)$, $\overrightarrow{M_1M_2} // \vec{S}$, so $\vec{S} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$. For a point N' on $\overrightarrow{M_1M_2}$, with the coordinates (x, y, z), then the linear equation L_1 can be expressed as:

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1} = t' \tag{7}$$

We can judge whether the wave signal is blocked according to the number of intersection points of the straight line equation and the surface equation. For example, the line segment SC has an intersection with the surface (except for the points S, C), and point C is a 3D perceived blind spot. Taking the monitoring point C and the sensing node S as an example, the spatial line segment SC expression is as shown in Fig. 2, (x_1, y_1, z_1) and (x_2, y_2, z_2) are the 3D coordinates of points S and C respectively. Theoretically, point C can be completely covered by node S. Based on the simultaneous line and surface equations, i.e., Eq. (8), the number of solutions is obtained.

$$\begin{cases} \dfrac{x - x_1}{x_2 - x_1} = \dfrac{y - y_1}{y_2 - y_1} = \dfrac{z - z_1}{z_2 - z_1} \\ z = f(x, y) \end{cases} \tag{8}$$

To facilitate the calculation, we can convert Eq. (8) into an equation containing only *x*, and find the solution of the equation in (x_1, x_2) (this is an open interval). It is judged whether there is a hillside between point S and point C according to the number of solutions, that is, whether the point C can be covered by the node S.

1. If Eq. (8) has no solution in the specified range, this means that there is no hillside between point S and point C, and node C can be covered.

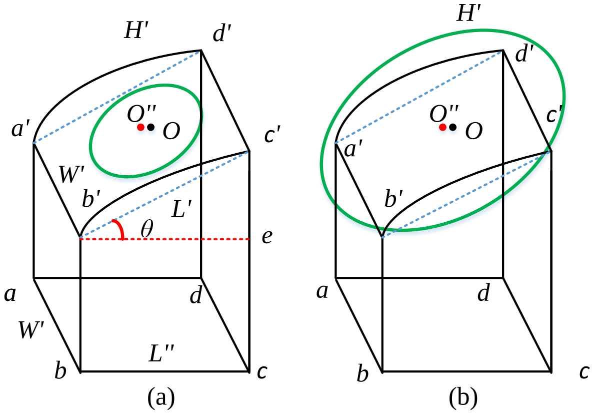
2. If the number of solutions is greater than or equal to 1, it means that there is an intersection point between the line equation and the surface, that is, there is a hillside, and the point C is not covered.

In Fig. 2, we know that point A can be perceived by node S, which illustrates the correctness of the proposed method. From the number of solutions, it can be determined whether there is an optimal perceptual blind spot, but there are still loopholes. For example, in the SD line segment of Fig. 2, except for the points S and D, the straight line and the curved surface have no intersection point, but the point D cannot be considered to be perceived by the node S. The shaded area in the figure is the hillside. Node S is on the west side of the hillside and point D is on the east side of the hillside. It can be seen from the figure that the sensing node S cannot cover the point D. The point O is the midpoint of the line segment SD, it is located below the curve. In the line segment SA, the midpoint O' is located above the curve. Therefore, based on this phenomenon, we supplement the method of judging the perceived blind spot in [27], that is, the third condition.

3. On the basis of satisfying the conditions 1 and 2, the position of the midpoint of the line segment needs to be determined. Take line segment SD as examples, assuming that point O is the midpoint of the line segment SD and its 3D coordinate is (x_3, y_3, z_3). If point O is below the curve, this means that the node S cannot cover the point D. This can be expressed by the Eq. (9).

$$p_{cov}(\mathrm{S},\mathrm{D}) = \begin{cases} 0 & if\ f(x_3, y_3) \geq z_3 \\ 1 & otherwise \end{cases} \tag{9}$$

Therefore, to determine whether a monitoring point is covered by a node or whether a node is perceived by another node, three conditions need to be met. First, the value of Eq. (2) must be 1. Second, the Eq. (8) has no solution. Finally, the value of Eq. (9) must be 1.



(a) (b)

FIGURE 3. Perceptual nodes cover small surfaces

It should be noted that we have to consider how many grids are suitable for dividing the 2D plane. Obviously, the more the grids are divided, the smaller the error between the calculated coverage area and the real coverage area will be. However, the calculation will be very heavy. On the opposite scenario, the calculation is light, but the error is large. In addition, the number of grids is also related to the perception radius of the sensor nodes. As shown in Fig. 3, $a'b'c'd'$ is a tangent plane of the small surface H', and the areas of both of them are considered to be approximately equal. O'' is the monitoring point of the surface H', and point O is the position of the sensor node, that is, the center of the circle O. In Fig. 3 (a), the node O covers the monitoring point O'', but it cannot be considered to cover the tangent plane $a'b'c'd'$, because the coverage area of the node O is too small. Therefore, we consider that when the coverage area of node O is larger than the area of the tangent plane $a'b'c'd'$, it is a reasonable coverage situation, as shown in Fig. 3 (b). Taking the Fig. 3. (a) as an example, in order to reduce the calculation error, the length (L') and width (W') of the tangent plane $a'b'c'd'$ should be less than 2 times the perception radius (r^p). The angle between the line segments $b'c'$ and $b'e$ is θ, then the length (L'') of the plane $abcd$ can represent $L' \cos\theta$. In general, the grid is divided by a 2D plane, and each grid must meet the following two conditions, which we call reasonable error conditions (REC), which can be expressed by the Eq. (10), and the error obtained at this time is considered to be within the accepted error range. Therefore, under reasonable error conditions, the entire 2D plane is divided into grids, and the appropriate number of grids can be obtained.

1. The area of the tangent plane $a'b'c'd'$must be smaller than the area of the circle O.

2. The length and width of the plane $abcd$ must be less than twice the perceived radius of the node.

$$\mathrm{REC} = \begin{cases} \dfrac{L' \cdot W'}{\pi(r^p)^2} < 1 \\ W' < 2r^p \\ L'' < 2r^p \end{cases} \tag{10}$$

E. NETWORK CONNECTIVITY DESCRIPTION

Network connectivity is the most basic requirement of a WSN. To meet the connectivity requirements, the communication radius (R^c) must be greater than or equal to twice the perception radius (r^p). For ease of calculation, assume $2r^p = R^c$. On the 3D surface, (x_1, y_1, z_1) and (x_2, y_2, z_2) are the 3D coordinates of point A and B respectively, point O is the midpoint of line segment BC, and its 3D coordinates are (x_3, y_3, z_3). As shown in Fig. 4, the Euclidean distance between AB is less than R^c, but there is a hillside between AB, so AB cannot communicate with each other. We call this phenomenon a communication blind spot. Similar to the perceived blind spot, the mathematical method can judge the communication state between the two nodes. First, the method has to meet Eq. (11), on the basis of which it has no solution for Eq. (8). Finally, the value satisfying Eq. (9) is 1. At this point, the two nodes can communicate with each other, otherwise, they cannot communicate. This method is consistent with perceived blind spot judgment, so it is just a simple statement.

$$d(A, B) \leq R^c \tag{11}$$

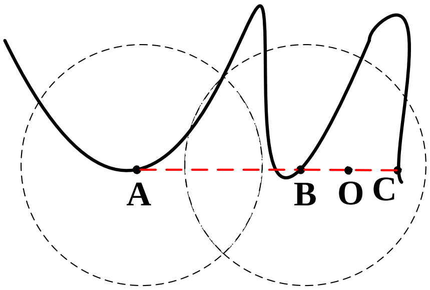


FIGURE 4. Communication blind spot

After obtaining the communication state between any two nodes, an undirected graph adjacency matrix vector $\boldsymbol{M_v}$ is established, which is used to store the connectivity of any two nodes, and then the connectivity is determined according to Eq. (12). $\boldsymbol{M}_v[i][j] = 1$ indicates that information can be transmitted between the i-th node and the j-th node, and when the value is 0, it indicates disconnection. Finally, the connectivity of the entire network is judged according to the matrix power algorithm in [52]. The matrix vector $\boldsymbol{S}_v$ is calculated by Eq. (13).

$$M_v[i][j] = \begin{cases} 1 & if\ s_i\ and\ s_j\ can\ communicate \\ 0 & otherwise \end{cases} \tag{12}$$

$$S_v = M_v + {M_v}^2 + {M_v}^3 \cdots + {M_v}^{n-1} \tag{13}$$

where n is the number of sensor nodes, if there is an element in $\boldsymbol{S}_v$ that is 0, the network is not connected, otherwise, it is connected. Based on the WSN connectivity, the Kruskal algorithm [53] is used to generate the minimum spanning tree.

F. SINGLE OBJECTIVE DEPLOYMENT OPTIMIZATION PROBLEM

When sensor nodes are deployed on a 3D surface, this paper assumes that the nodes can be charged by solar energy, so energy consumption is not considered. Therefore, we only need to optimize the coverage of the network. The greater the coverage, the better the deployment effect. $\boldsymbol{O}$ is a matrix of position coordinates, perception radius and communication radius of a group of sensor nodes. Except for a few fixed nodes, the position of each other node (s_i) should be on the surface (H), which can be expressed as: $A(s_i) \in H$. Because the constraints of network connectivity will bring extra running time to the algorithm, this paper does not consider the constraints of network connectivity, but uses the inner-layer encircle strategy mentioned in Section V A to increase the probability of network connectivity. Based on the description in Section III A, the problem is as follows:

$$\begin{cases} f_1(\boldsymbol{O}) = \mathrm{Max}\big(C_r(\boldsymbol{O})\big) \\ \quad \mathrm{s.t.} \quad A(s_i) \in H \end{cases} \tag{14}$$

IV. GREY WOLF OPTIMIZER

A. SOCIAL HIERARCHY

Inspired by the predation process of the grey wolf colony, Mirjalili et al proposed GWO through scientific mathematical abstraction [31]. Wolves usually go out for prey collectively, but at the same time they adhere to a strict social hierarchy. Wolves are divided into four levels, from high to low, α, β, δ, and ω. α, β and δ are the three wolves closest to the prey in the population, that is, the three wolves with the highest level in the hierarchical pyramid shown in Fig. 5, and the remaining grey wolf individuals are denoted as ω and have the lowest level. α, β and δ are responsible for leading ω to hunt for prey. It is worth noting that the social level of each wolf is not fixed. With the round-up, all grey wolves will be re-ranked according to the distance from the prey, that is, the social level will be re-ranked according to the strength of the fitness value.

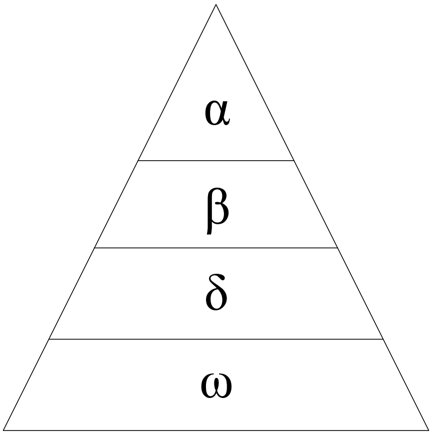


FIGURE 5. Grey wolf social hierarchy pyramid

B. ROUND HUNTING

The highest social level α, β and δ issue a hunting and killing command to ω, and ω approaches the prey from all directions to update its position.

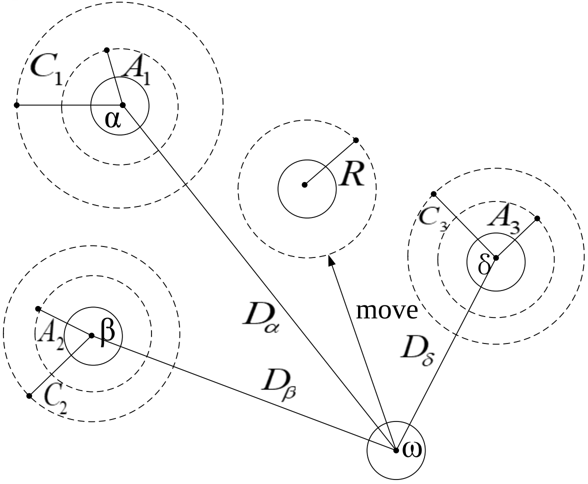


FIGURE 6. Grey Wolf location update

Fig. 6 is a schematic diagram of the updated position of the grey wolf. GWO simulates the approximate position of the prey by the positions of α, β and δ. When the ω receive the killing command, it approaches the prey, that is, updates its position. The distance between ω and α, β and δ can be expressed as follows:

$$\begin{cases} \vec{D}_\alpha = \left|\vec{C}_1 \cdot \vec{X}_\alpha(t) - \vec{X}_\omega(t)\right| \\ \vec{D}_\beta = \left|\vec{C}_2 \cdot \vec{X}_\beta(t) - \vec{X}_\omega(t)\right| \\ \vec{D}_\delta = \left|\vec{C}_3 \cdot \vec{X}_\delta(t) - \vec{X}_\omega(t)\right| \end{cases} \tag{15}$$

$$\begin{cases} \vec{C}_1 = 2 \cdot \vec{r}_1 \\ \vec{C}_2 = 2 \cdot \vec{r}_2 \\ \vec{C}_3 = 2 \cdot \vec{r}_3 \end{cases} \tag{16}$$

Where $\vec{D}_\alpha$ represents the distance between α and ω, $\vec{D}_\beta$, $\vec{D}_\delta$ and so on. $\vec{X}_\alpha(t)$ is the position of α in the t-th (ie, current) iteration, $\vec{X}_{\beta}(t)$, $\vec{X}_{\delta}(t)$, and $\vec{X}_{\omega}(t)$ and so on. $\vec{C}_1$ is the azimuth variable when ω moves to α, and $\vec{C}_2$, $\vec{C}_3$ and so on. $\vec{r}_1$ is a random number between [0, 1].

After learning the distances from α, β and δ, ω approaches each at a certain step, and its position update can be expressed as follows:

$$\begin{cases} \vec{X}_1 = \vec{X}_{\alpha} - \vec{A}_1 \cdot \vec{D}_{\alpha} \\ \vec{X}_2 = \vec{X}_{\beta} - \vec{A}_2 \cdot \vec{D}_{\beta} \\ \vec{X}_3 = \vec{X}_{\delta} - \vec{A}_3 \cdot \vec{D}_{\delta} \end{cases} \tag{17}$$

$$\begin{cases} \vec{A}_1 = 2a \cdot \vec{r}_4 - a \\ \vec{A}_2 = 2a \cdot \vec{r}_5 - a \\ \vec{A}_3 = 2a \cdot \vec{r}_6 - a \end{cases} \tag{18}$$

$$a = 2 - 2\left(\frac{t}{T}\right) \tag{19}$$

$$\vec{X}(t+1) = (\vec{X}_1 + \vec{X}_2 + \vec{X}_3)/3 \tag{20}$$

$\vec{X}_1$ indicates that if ω only uses α as its prey, its updated position, $\vec{X}_2$, $\vec{X}_3$ and so on. t is the current number of iterations, and T is the maximum number of iterations. a is the convergence factor, and its role is to reduce the distance between the grey wolf and its prey as the iteration progresses. $\vec{A}_1$ is the step vector of ω moving in the direction of α, $\vec{A}_2$, $\vec{A}_3$ and so on. Under the joint leadership of α, β and δ, its position can be expressed as the arithmetic mean of $\vec{X}_1$, $\vec{X}_2$and $\vec{X}_3$.

V. ENHANCED GREY WOLF OPTIMIZER (EGWO)

From the description of the GWO algorithm [29], it can be known that when $|\vec{A}| > 1$, the grey wolf group will expand the search range to find better candidate solutions, which is the global exploration ability of the GWO algorithm. $\vec{A}$ includes $\vec{A}_1$, $\vec{A}_2$, etc. When $|\vec{A}| < 1$, the grey wolf group narrow down the search space and conduct a fine search in the local area, which is the exploitation ability of the GWO algorithm. In order to improve the convergence ability of the algorithm, we try to strengthen the exploitation ability of GWO, which is called enhanced grey wolf optimizer algorithm (EGWO). In EGWO, we divided the entire grey wolf group, half of them were responsible for the outer layer of encircling and the other half were responsible for the inner layer of encircling. These two groups are called " inner-layer group " and " outer-layer group ", and the corresponding strategies to be executed are called " inner-layer encircle " and " outer-layer encircle " respectively. The reason why the number of inner-layer and outer-layer groups is the same is that EGWO can obtain good results in most cases through experiments. The role of inner-layer encircle is to enhance the grey wolf's ability to attack prey, that is, the exploitation ability of the algorithm. To improve the optimization precision of the algorithm, Tent mapping is used to enrich the diversity of the initial population. In addition, outer-layer encircle is to maintain the global exploration ability of GWO. Figure 7 is the inner and outer layers of the grey wolf group.

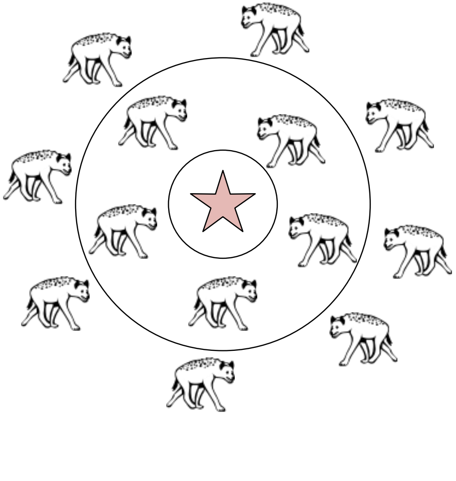


FIGURE 7. Inner and outer encircle diagram

A. INNER-LAYER ENCIRCLE

The local search of the algorithm emphasizes the use of the existing information to influence other search agents to perform fine searches in certain search space, which has a decisive effect on the convergence speed of the algorithm. Therefore, EGWO focuses on improving the exploitation ability of the algorithm. In the EGWO algorithm, α is more intelligent and wise, and a more detailed encircle order is arranged for the grey wolf group. The distance between the grey wolf group and the prey that executed the inner-layer encircle command is short, just follow the order of α. The position of α is used to simulate the position of the prey. Therefore, the grey wolf group that executes the inner-layer encircle only needs to move around α. Its position update strategy is similar to GWO, though still different.

$$\begin{cases} \vec{D}'_{\alpha} = \left|\vec{C}_4 \cdot \vec{X}_{\alpha}(t) - \vec{X}_{\alpha}(t)\right| \\ \vec{C}_4 = 2 \cdot \vec{r}_7 \\ \vec{X}'_1 = \vec{X}_{\alpha} - \vec{A}_4 \cdot \vec{D}'_{\alpha} \\ \vec{A}_4 = 2a \cdot \vec{r}_8 - a \\ \vec{X}(t+1) = \vec{X}'_1 \end{cases} \tag{21}$$

It can be seen from the above formula that the update strategy is similar to GWO. The difference is that in the inner-layer encircle, we cancelled the influence of β and δ on ω, and instead obeyed the command of α uniformly. $\vec{D}'_{\alpha}$ is the distance between ω and α, $\vec{r}_7$ is a random decimal vector between [0,1], $\vec{C}_4$ is the corresponding direction vector, and $\vec{A}_4$ is the corresponding step vector, so the updated position of the grey wolf can be expressed by the Eq. (21).

B. OUTER-LAYER ENCIRCLE

Global search means that the grey wolf group needs to explore a wider search space in order to find the global optimal solution. Therefore, the grey wolf group that executes the order of outer-layer encircle is approaching the prey slowly from all directions, from far to near. In order to avoid the algorithm falling into the local optimum, the outer-layer group is not only obeying the command of α, but also the commands of β and δ. To make a long story short, the position update strategy of the outer-layer group is consistent with GWO, that is, the Eqs. (15-20).

After the grey wolf group executes the inner-layer encircle and the outer-layer encircle strategies, the position of the entire grey wolf group is updated. Then in EGWO, the fitness value calculation method of each grey wolf can be expressed by Eq. (22), $\vec{X}$ is an individual of the grey wolf group, and fit is an abstract expression of the optimization problem. It should be noted that the mathematical expressions are different for different optimization problems. After obtaining the fitness of the entire grey wolf group, these grey wolves are ranked according to the size of the fitness values. Choose the best three wolves ($\vec{X}_\alpha$, $\vec{X}_\beta$ and $\vec{X}_\delta$) according to the nature of the optimization problem. For example, suppose the minimization problem is solved, then $\vec{X}_\alpha$, $\vec{X}_\beta$ and $\vec{X}_\delta$ are the three wolves with the smallest fitness values. Conversely, the three wolves with the highest fitness values.

$$fitness = fit(\vec{X}) \tag{22}$$

C. TENT MAPPING

To enrich the population diversity of search agents, many swarm intelligence optimization algorithms [54,55] use Tent mapping to generate search sequences. [54] showed that Tent mapping can enrich the diversity of the initial population, reduce the probability of falling into a local optimum, and then improve the global exploration ability of the algorithm, so the algorithm can find a better solution. The literature [55] pointed out that Tent mapping is the most outstanding one among many chaotic mapping functions. The chaotic sequences generated by Tent do not require much of the initial value of the distribution function, so it is widely used. Based on these advantages of Tent mapping, it is cited in this paper to enhance the exploration ability of the algorithm. Specifically, this paper cites the Tent equation of [55], as shown in Eq. (23). x_t represents the value of the t-th iteration, and x_{t+1}represents the value obtained by the $(t+1)$-th iteration. After multiple iterations, a rich search sequence can be generated.

$$x_{t+1} = \begin{cases} \dfrac{x_t}{0.7} & x_t < 0.7 \\ \dfrac{1-x_t}{0.3} & x_t \geq 0.7 \end{cases} \tag{23}$$

D. EGWO PSEUDO CODE

The EGWO pseudo code is as follows:

TABLE 1. Enhanced Grey Wolf Optimizer

```
Algorithm 1: Enhanced Grey Wolf Optimizer
Input: population size $N$, maximum number of iterations $T$
Output: Grey wolf $X_\alpha$
Initialize the grey wolf population vector W by Eq. (23)
$t = 1$
while $t < T$ do
    Initialize/update $a$, $A$ and $C$
    Calculate the fitness value of W by Eq. (22)
    The best grey wolf in W as $X_\alpha$
    The second best grey wolf in W as $X_\beta$
    The third best grey wolf in W as $X_\delta$
    for $i = 1$ to $N/2$ do
        Calculate $D_\alpha$, $D_\beta$, $D_\delta$, $X_1$, $X_2$ and $X_3$ by Eqs. (15-19)
        Update the position of ω by Eq. (20)
    end
    for $i = N/2+1$ to $N$ do
        Calculate $D'_\alpha$ and $X'_1$ by Eq. (21)
        Update the position of ω by Eq. (21)
    end
    $t = t + 1$
end
return $X_\alpha$
```

E. TIME COMPLEXITY ANALYSIS

Assume that the maximum number of iterations of the algorithm is T, the size of the population is N, the dimension of the optimization problem is D, and the time complexity of the initialization of the population is O(ND). In GWO, first, the fitness value of each grey wolf needs to be calculated and the best three grey wolves are found, with a time complexity of O(N). Later, the location of all grey wolves is updated, and its time complexity is O(N). Therefore, the total time complexity of each iteration is O(N+N), so the total time complexity required by GWO is O(T(N+N)+ND). That is at the level of O(TN). Compared with GWO, EGWO adds an inner-layer encircle mechanism to the location update strategy, and the time complexity of the grey wolf population location update is still O(N). In addition, the composition structure of GWO and EGWO is the same. Therefore, no additional time complexity is added, so the time complexity of EGWO is consistent with GWO, which is O(TN). The time complexity of the EEGWO, mGWO, and ROLGWO can be seen in Table 2, and the time complexity of the ABC-DSS, and DEA can be seen in Table 12.

F. STEPS FOR APPLYING EGWO TO WSN

It should be noted that when EGWO is applied to the WSN deployment of 3D surface, it is necessary to pay attention to the deployment of nodes under the inner-layer encircle strategy. In order to increase the possibility of network connectivity, only a few nodes are selected (in the simulation experiment, the number of nodes selected to update the position is 2) to update their positions according to Eq. (21). The steps for WSN deployment on 3D surfaces are as follows.

1). Set the number of sensor groups (population) N_1, the number of sensor nodes (dimension) D_1 and the range of monitoring area (surface H).

2). $\boldsymbol{O}_{s}$ represents position and radius of N_1 groups of sensor nodes, $\boldsymbol{O}_{s}$ is initialized first, that is, all node positions must be initialized on the surface H.
3) Set the current number of iterations $t = 1$ and the maximum number of iterations T
4). According to the description of Section III B and the Eq. (10), the surface H is divided into multiple small surfaces, and the position coordinates of all monitoring points on the surface H are calculated.
5). Calculate the coverage of each group of sensor nodes according to Eq. (4). The WSN deployment problem is optimized according to Eq. (14).
6) The grey wolf group is divided into two parts. The outer-layer group of the grey wolf updates the node position according to Eqs. (15-20), and the inner-layer group of the grey wolf updates the node position according to Eqs. (19) and (21). In addition, the position of the nodes must be constrained by the constraints of Eq. (14).
7). When t does not reach the maximum number of iterations T, go to step 5), otherwise the algorithm ends and a group of sensor node deployment solutions $\boldsymbol{O}_1$ with the largest coverage can be obtained.
8). Determine whether the network ($\boldsymbol{O}_1$) is connected according to the Eq. (13). If it is not connected, select a suboptimal set of deployment plans until the connected set of plans.
9). Generate the WSN minimum spanning tree according to the Kruskal algorithm mentioned in [53].

VI. SIMULATION EXPERIMENTS AND ANALYSIS

In this section, two sets of experiments are performed. The first experiment is to test all comparison algorithms on the benchmark function, the purpose is to test the effectiveness and superiority of EGWO. In Experiment 2, the comparative deployment algorithm is applied to the 3D surface deployment of WSN, and the experimental results are analyzed. All experiments were performed on an Intel Core i5 dual-core CPU running at 2.4GHz, 8GB of memory and Windows 10 operating system. The experimental simulation software is MATLAB 2014b.

A. EXPERIMENT 1: BENCHMARK FUNCTION TEST

In this part, we use 12 well-known benchmark functions to test the performance of EGWO. The 12 benchmark functions can be found in the literature [56,57], and are classic functions used by many researchers [45,46,55,58]. There are three types of benchmark functions. Including 3 unimodal functions, 3 multimodal functions, 6 fixed-dimensional multimodal functions. The detailed information of the benchmark functions can be viewed from the Tables 3-5. The attributes "optimum" in the table are theoretical optimal values of the function, "range" is the search range of the algorithm, and "best position" is the position corresponding to the function's optimal value. The characteristic of the unimodal function is that there is no local optimal value, so it is easier to find the global optimal value, which can be used to test the exploitation ability of the algorithm. Multimodal functions are characterized by multiple local minimums, and it is relatively difficult to find a global optimum. Multimodal fixed-dimension functions are characterized by mathematical equations that are repeated multiple times, which is equivalent to finding the maximum or minimum of the equation. These two types of benchmark functions can be used to test the exploration ability of the algorithm. Figs. 8-10 list the images of the functions when the search dimension is 2.

1) DESIGN OF THE EXPERIMENT

In order to verify that the performance of EGWO has been significantly improved, we will compare EGWO with the original GWO. At the same time, EGWO will be compared with some recently improved grey wolf optimizers, including mGWO (modified GWO), EEGWO (exploration-enhanced GWO), and ROLGWO (random opposition learning strategy into GWO). In relation with the benchmark algorithm GWO, their improvements are listed in Table 2. Information on comparison algorithms can be found in Table 2. The effectiveness of EGWO is illustrated by comparing with them.

TABLE 2. Description of the comparison algorithms

Algorithm	Reference	Convergence factor	Location update	Time complexity
GWO	31	-	-	O(TN)
mGWO	42	Yes	No	O(TN)
EEGWO	45	Yes	Yes	O(TN)
ROLGWO	46	No	Yes	O(TN)
EGWO	This paper	No	Yes	O(TN)

By reviewing the grey wolf position update strategy in [45] and [46] and the results of corresponding experiments, it is obvious that EEGWO and ROLGWO have a natural optimization effect on the benchmark function with the optimal position of 0 and perform well. In addition, the two algorithms also have good search results on the benchmark function whose midpoint of the search range is 0. So from the "range" attribute in Tables 3-5, for some benchmark functions, we set the middle point of the search range not at 0, and at the same time, we have added benchmark functions whose optimal position is not 0, such as F7, F8, and so on. In the experiments of unimodal and multimodal benchmark functions, the dimension of the function is 30, the population size of all comparison algorithms is set to 30, and the maximum number of iterations is 500. The setting of these basic parameters is consistent with the literature [31,45,46]. The fixed-dimensional multimodal functions have an experimental dimension of 2. The population size of all comparison algorithms is 30, and a corresponding maximum number of iterations of 100. Each set of experiments runs for 20 times, and the average was finally compared. The parameters of the comparison algorithms are consistent with those in the literature [31,42,45,46]. In particular, in EEGWO, the local learning factor $b_1 = 0.1$, the global learning factor $b_1 = 0.9$, and the nonlinear modulation index $\mu = 1.5$.

TABLE 3. Unimodal benchmark functions

Function	Formula	Range	Best position	Optimum
Sphere	$F_1(x)=\sum_{i=1}^{D} x_i^2$	[-100,100], **[-80,100]**	$(0,0,0)^n$	$f_{\min}=0$
Schwefel2.22	$F_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	[-10,10], **[-8,10]**	$(0,0,0)^n$	$f_{\min}=0$
Rosenbrock	$F_3(x)=\sum_{i=1}^{n-1}\begin{bmatrix}100(x_{i+1}-x_i^2)^2\\+(x_i-1)^2\end{bmatrix}$	**[-30,25]**	$(0,0,0)^n$	$f_{\min}=0$

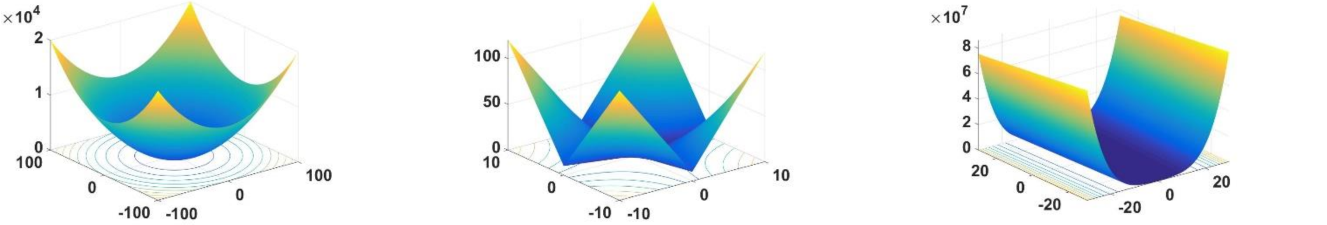
TABLE 4. Multimodal benchmark functions

Function	Formula	Range	Best position	Optimum
Ackley	$F_4(x)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+20+\mathrm{e}$	[-32,32], **[-28,32]**	$(0,0,0)^n$	$f_{\min}=0$
Griewank	$F_5(x)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[-600,600], **[-600,500]**	$(0,0,0)^n$	$f_{\min}=0$
Rastrigin	$F_6(x)=\sum_{i=1}^{D}[x_i^2-10\cos(2\pi x_i)+10]$	**[-5.12,4.8]**	$(0,0,0)^n$	$f_{\min}=0$

TABLE 5. Fixed-dimension multimodal benchmark functions

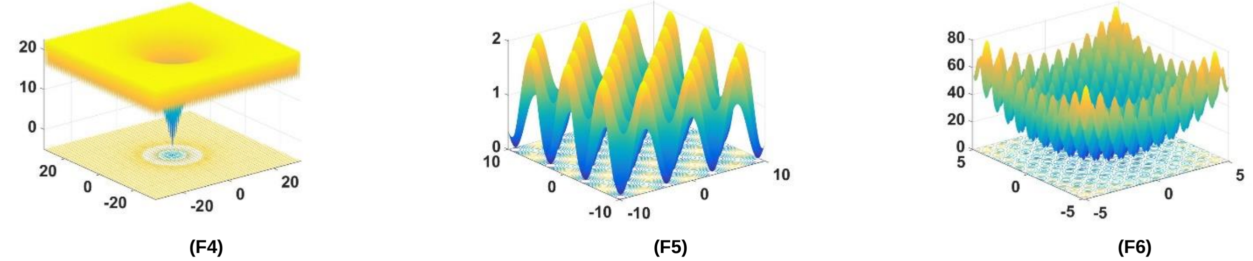
Function	Formula	Range	Best position	Optimum
Booth	$F_7(x)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$	[-10,10], **[-8,10]**	$(1, 3)^2$	$f_{\min}=0$
Easom	$F_8(x)=-\cos x_1\cos x_2\exp(-(x_1-\pi)^2-(x_2-\pi)^2)$	[-100,100], **[- 100,80]**	$(\pi, \pi)^2$	$f_{\min}=-1$
Goldstein &Price	$F_9(x)=\begin{bmatrix}1+(x_1+x_2+1)^2(19-14x_1\\+3x_1^2-14x_2+6x_1x_2+3x_2^2)\end{bmatrix}\times\begin{bmatrix}30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2\\+48x_2-36x_1x_2+27x_2^2)\end{bmatrix}$	**[-2,10]**	$(0, 1)^2$	$f_{\min}=3$
Branin	$F_{10}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10$	**[-5,10]**	$(\pi, 2.275)^2$	$f_{\min}=0.3979$
Hartmann	$F_{11}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2)$	[0,1]	$(0.1146, 0.5556,$ $0.8525)^3$	$f_{\min}=-3.8628$
Hump	$F_{12}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	**[-10,5]**	$(0.0898, -0.7126)^2$, $(-0.0898, 0.7126)^2$	$f_{\min}=-1.0316$

Tables 6-8 are the experimental results of 5 algorithms on each test function. The first column in Table 6 is the corresponding function number, but it should be noted that the search range of the function numbers with special marks is not symmetrical about 0. For example, F1', the search range is [-80,100]. " f_{best} ", " f_{mean}" and " f_{std} " represent the best results, average results, and variances of fitness values in the 20 independent experiments. " Rank " represents the performance ranking of 5 algorithms in a set of tests. This ranking is based on the optimization precision of the " f_{mean} " column. The number of effective evaluations refers to how many times the algorithm evaluates when it finds the optimal solution. For example, the number of population is 30, the maximum number of iterations is 500, and the algorithm finds the optimal solution at 100 generations, then the total number of evaluations is 15000, the number of valid evaluations is 3000, and the smaller the number of valid evaluations, the faster the algorithm finds the optimal solution.



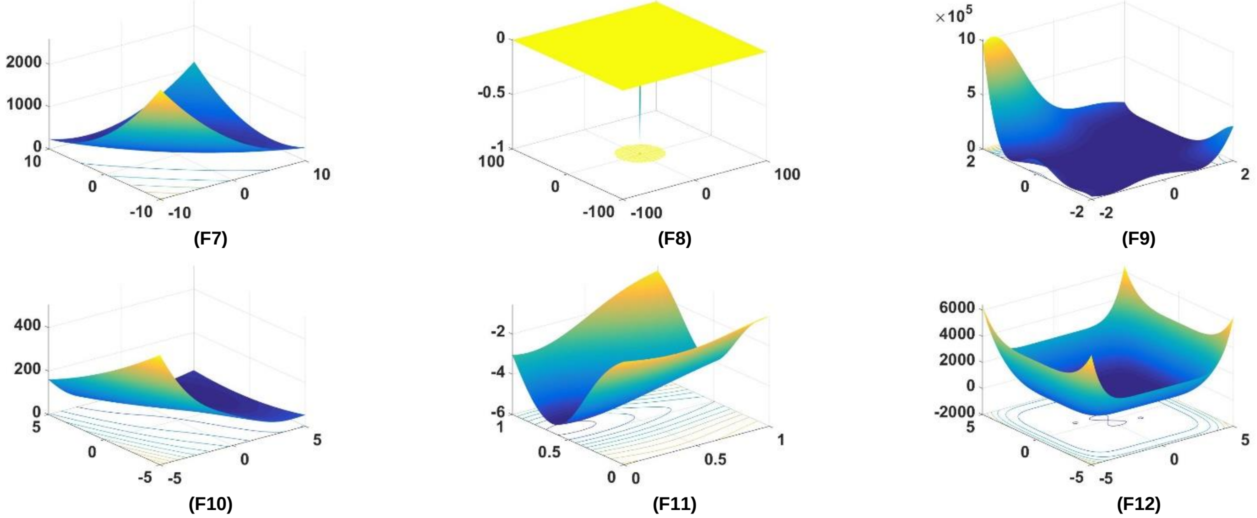
(F1) (F2) (F3)

FIGURE 8. 2-D versions of unimodal benchmark functions



(F4) (F5) (F6)

FIGURE 9. 2-D versions of multimodal benchmark functions



(F7) (F8) (F9)

(F10) (F11) (F12)

FIGURE 10. 2-D version of fixed-dimension multimodal benchmark functions

2) ANALYSIS OF EXPLOITATION AND EXPLORATION

As shown in Table 3, these unimodal benchmark functions with only a global optimal value are suitable for testing the exploitation ability of the algorithm. Table 6 shows the test results of the 5 algorithms on the unimodal benchmark function. From the "f_{mean}" or "Rank" column in Table 6, EGWO can provide competitive search results. Compared with GWO, EGWO can search better result in most benchmark functions, and outperforms mGWO. This shows that the inner-layer encircle strategy we proposed is helpful to strengthen the exploitation ability of the algorithm.

As shown in Tables 4-5, multimodal type benchmark function with multiple local optimums is the best choice for testing the algorithm's global exploration abilities. Tables 7-8 are the test results of 5 algorithms in these functions. It can be known from the "f_{mean}" column or the "Rank" column in the Table 7-8. EGWO can still find competitive results. Compared with GWO and mGWO, the algorithm performs better in most benchmark functions. In addition, EGWO shows a very competitive potential compared to EEGWO and ROLGWO, which in most cases is comparable to their performance and occasionally surpasses them, such as in F7, F7', F10', F11 and F12'. These results show that the EGWO algorithm has advantages in exploration. The EGWO can achieve such results, which shows that Tent mapping has played a role in enriching the initial population, strengthened the algorithm's global exploration ability, and reduced the probability of the algorithm falling into local optimal. It should be noted that the equation for the position update of the outer-layer group is consistent with the GWO, it is the basis for EGWO's good global exploration ability.

In order to better understand why the algorithm has a good or a bad optimization effect on the optimization problem, literatures [59,60] attempt to reveal the search process of the algorithm from the search space's exploitation rate and exploration rate. To evaluate the exploitation and exploration abilities of EGWO, this study draws two from three types of benchmark functions, namely F1 and F3 of unimodal type, F5 and F6 of multimodal type, and F9 and F11 of multimodal fixed dimension type. EGWO's search process on these 6 benchmark functions is shown in Fig. 11. It can be seen from the figure that in the unimodal and multimodal images, the exploitation rate curve has always been above the exploration rate curve, indicating that the exploitation ability of EGWO is outstanding, and it is verified that the proposed inner-layer encircle strategy indeed strengthens the exploitation ability of EGWO. As the iteration progressed, the two curves gradually stabilized, indicating that the exploration performance and exploitation performance of the algorithm can be balanced. On the other hand, it shows that using half of the grey wolf group for inner-layer encircle can still achieve a balance between exploration and exploitation. It can be seen from the F9 image that the exploitation rate and the exploration rate have stabilized shortly after the iteration, and in the end both sides accounted for 50%. In F11, the exploration rate curve and the exploitation rate curve alternate in the vicinity of 50%, and the two eventually reach 50%. In the F9 and F11 images, when the iteration is over, their exploitation rate and exploration rate have reached 50%, respectively. We think this phenomenon is caused by the low dimension of these two functions.

3) OPTIMIZATION PRECISION AND ALGORITHM STABILITY ANALYSIS

As can be seen from Table 6 to Table 8, the precision of the optimal solution searched by EGWO is better than other comparison algorithms in most cases. Taking F2 as an example, in the "f_{mean}" column, the precision of the EGWO solution is higher than that of GWO and mGWO, but lower than that of EEGWO and ROLGWO. Compared with the single-strategy GWO, the precision of the solution is significantly improved. And it is better than the mGWO of the nonlinear convergence factor. EGWO's excellent results verify that the Tent mapping is helpful to enhance the exploration ability of the algorithm. Tent mapping enriches the diversity of the initial population, that is, it expands the search space of the search agent, so the algorithm easily jumps out of the local optimum, and searches for a solution with higher optimization precision. ROLGWO achieved a good optimization effect, but in the F2' (not about 0 symmetrical search range) test, its optimization effect was greatly reduced. There are similar phenomena (F1 and F1', F4 and F4'), so it is supposed that two conditions are beneficial to ROLGWO. One is that the search range is symmetrical about 0, and the other is that the position of the optimal solution is 0. The second condition is deemed to have the biggest impact. For example, in Table 8 (fixed-dimension multimodal benchmark functions), the position of the optimal solution is not at the position of 0, this is not changed whether it is symmetrical or asymmetric about 0. EEGWO achieved good results on F2, and its optimization effect was not affected on F2' (asymmetric search range). This performance seems to be perfect, but in the test function where the optimal solution position is not 0, the effect is very poor. For example, in Table 8 (the optimal solution is not at the position of 0), EEGWO always cannot find the optimal solution, and the precision of the solution is the worst. This is because the EEGWO algorithm is always closer to the position where the direction is 0, so the optimal solution cannot be searched. However, the mGWO performance that only improves the convergence factor is satisfactory, and is not limited by the experimental parameters. Overall, EGWO performed well in terms of solution optimization precision. However, in the F9′ test, except for the good performance of mGWO, the other four algorithms performed poorly, and EGWO performed the worst. F9' is a multimodal benchmark function with fixed dimension that is difficult to find the optimal solution. Therefore, most comparison algorithms have difficulty in finding the optimal solution. EGWO performs the worst, we believe the reason is as follows. Compared with GWO, although Tent mapping has enriched the diversity of the population to some extent, the number of search agents that perform global exploration is reduced because half of the search agents of EGWO execute the inner-layer encircle command. Therefore, when EGWO optimizes complex optimization problems, its global exploration ability is weakened, then the algorithm is easy to fall into a local optimum, and the optimization effect is poor.

As can be seen from the "f_{std} " column in Tables 6-8, GWO performs poorly in most cases, and the mGWO performs slightly better than GWO. ROLGWO limited by experimental parameters has good stability. The EEGWO, which was limited by experimental parameters, performed poorly. The second column of Fig. 12 shows the optimal solutions of 20 separate experiments as boxplot. GWO and EEGWO have the highest box height, which indicates that its stability is poor, while the box height of EGWO is low, this shows that EGWO has good stable performance. The algorithm is stable, which means that the accuracy level of the solution it searches for is close to that in multiple independent experiments. In other words, the algorithm has a good global search ability, which obviously benefits from Tent mapping as it enriches the diversity of the population and indirectly enhances the algorithm's global exploration ability.

TABLE 6. Results of unimodal benchmark functions

Function	Dimension	Algorithm	f_{best}	f_{worst}	f_{mean}	f_{std}	f_{mean} of population	Number of evaluations	Rank	Running time (s)
F1	30	GWO	1.15e-29	2.07e-26	2.10e-27	4.73e-27	2.10e-27	14985	5	0.43
		mGWO	3.33e-38	9.93e-36	1.29e-36	2.34e-36	1.29e-36	14980	4	**0.41**
		EEGWO	1.91e-144	8.57e-140	6.76e-141	1.99e-140	4.50e-136	14952	2	0.70
		ROLGWO	**0**	**0**	**0**	**0**	**0**	**12160**	**1**	0.92
		EGWO	3.18e-61	2.81e-55	1.69e-56	6.26e-56	1.70e-56	14993	3	0.46
F1'	30	GWO	7.34e-29	6.70e-27	9.3437e-28	1.48e-27	9.35e-28	14984	5	0.47
		mGWO	1.36e-38	5.44e-36	1.04e-36	1.42e-36	1.04e-36	14983	4	0.44
		EEGWO	**1.86e-144**	**4.55e-139**	**3.09e-140**	**1.05e-139**	**3.54e-135**	**14964**	**1**	0.76
		ROLGWO	1.67e-41	5.80e-39	8.6725e-40	1.46e-39	8.68e-40	14979	3	0.91
		EGWO	2.63e-62	1.37e-56	2.0854e-57	3.63e-57	2.09e-57	14994	2	**0.42**
F2	30	GWO	2.07e-17	2.83e-16	6.85e-17	5.86e-17	6.86e-17	14984	5	0.36
		mGWO	1.57e-22	2.83e-21	9.23e-22	7.24e-22	9.24e-22	14987	4	0.34
		EEGWO	1.86e-73	2.83e-71	6.66e-72	7.09e-72	1.47e-70	**14967**	2	0.61
		ROLGWO	**1.42e-207**	**8.97e-205**	**1.56e-205**	**0**	**2.04e-205**	14986	**1**	0.54
		EGWO	4.80e-39	2.51e-35	2.25e-36	6.13e-36	2.25e-36	14993	3	**0.30**

F2'	30	GWO	1.89e-17	2.40e-16	6.35e-17	5.77e-17	6.36e-17	14983	5	0.37
		mGWO	8.73e-23	3.74e-21	7.79e-22	8.06e-22	7.80e-22	14983	4	**0.34**
		EEGWO	**1.21e-73**	**8.34e-71**	**1.17e-71**	**2.14e-71**	**1.32e-70**	**14971**	**1**	0.62
		ROLGWO	8.82e-25	9.12e-23	2.43e-23	2.46e-23	2.43e-23	14985	3	0.60
		EGWO	3.58e-39	1.98e-35	2.04e-36	5.94e-36	2.04e-36	14993	2	0.36
F3'	30	GWO	26.1051	**27.9676**	**26.7191**	0.67589	**26.7194**	14980	**1**	0.47
		mGWO	**25.7601**	27.9869	27.0889	0.72708	27.0895	14979	2	**0.40**
		EEGWO	28.9286	28.9775	28.96	**0.012688**	28.9971	**778**	5	0.77
		ROLGWO	25.9870	28.7880	27.6501	0.71132	27.6502	14985	4	0.92
		EGWO	26.1936	28.8194	27.4310	0.78164	27.4312	14984	3	0.42

TABLE 7. Results of multimodal benchmark functions

Function	Dimension	Algorithm	f_{best}	f_{worst}	f_{mean}	f_{std}	f_{mean} of population	Number of evaluations	Rank	Running time (s)
F4	30	GWO	7.55e-14	1.57e-13	1.09e-13	1.99e-14	1.13e-13	11141	5	0.39
		mGWO	1.51e-14	2.93e-14	2.29e-14	5.10e-15	2.65e-14	8967	4	0.41
		EEGWO	**8.88e-16**	**8.88e-16**	**8.88e-16**	**0**	**9.41e-16**	3445	**1**	0.63
		ROLGWO	**8.88e-16**	4.44e-15	1.95e-15	1.67e-15	3.77e-15	**2203**	2	0.76
		EGWO	4.44e-15	7.99e-15	7.46e-15	1.30e-15	9.06e-15	4843	3	**0.35**
F4'	30	GWO	6.48e-14	1.18e-13	9.66e-14	1.43e-14	1.00e-13	11142	5	0.47
		mGWO	1.51e-14	2.93e-14	2.24e-14	4.07e-15	2.61e-14	9115	4	0.42
		EEGWO	**8.88e-16**	**8.88e-16**	**8.88e-16**	**0**	**9.12e-16**	**3452**	**1**	0.77
		ROLGWO	1.15e-14	2.22e-14	1.69e-14	3.16e-15	2.01e-14	6847	3	0.86
		EGWO	4.44e-15	7.99e-15	6.22e-15	1.82e-15	7.82e-15	4492	2	**0.40**
F5	30	GWO	0	0.0194	0.0016	0.0050	0.0016	6530	5	0.40
		mGWO	0	0.0105	0.0005	0.0023	0.0005	6471	4	0.37
		EEGWO	0	0	0	0	**0**	2145	**1**	0.71
		ROLGWO	0	0	0	0	5.55e-19	**868**	**1**	1.10
		EGWO	0	0	0	0	2.76e-17	2969	**1**	**0.36**
F5'	30	GWO	0	0.0253	0.0013	0.0057	0.0013	6366	5	**0.48**
		mGWO	0	0.0144	0.0007	0.0032	0.0007	6530	4	0.49
		EEGWO	0	**0**	**0**	**0**	**0**	**2122**	**1**	0.71
		ROLGWO	0	**0**	**0**	**0**	4.87e-17	4442	**1**	0.85
		EGWO	0	**0**	**0**	**0**	3.89e-17	3707	**1**	**0.48**
F6'	30	GWO	0	20.6566	2.4385	5.0792	2.4394	11292	5	**0.61**
		mGWO	0	5.17e-12	2.58e-13	1.15e-12	2.60e-13	7528	4	0.64
		EEGWO	0	**0**	**0**	**0**	**0**	**1983**	**1**	0.97
		ROLGWO	0	**0**	**0**	**0**	1.15e-15	4459	**1**	1.21
		EGWO	0	**0**	**0**	**0**	7.99e-16	2880	**1**	**0.61**

TABLE 8. Results of fixed-dimension multimodal benchmark functions

Function	Dimension	Algorithm	f_{best}	f_{worst}	f_{mean}	f_{std}	f_{mean} of population	Number of evaluations	Rank	Running time (s)
F7	2	GWO	**1.17e-08**	**1.14e-05**	4.33e-06	3.32e-06	0.0008	2937	3	0.04
		mGWO	2.90e-07	6.82e-05	2.01e-05	1.69e-05	0.0028	2963	4	0.04
		EEGWO	0.1051	1.7650	0.7185	0.4568	67.9587	**86**	5	0.11
		ROLGWO	8.92e-07	1.93e-05	6.89e-06	5.29e-06	**0.0007**	2970	2	0.07
		EGWO	2.28e-07	1.30e-05	**2.92e-06**	**3.15e-06**	0.0014	2793	**1**	0.04
F7'	2	GWO	1.35e-06	**1.61e-05**	5.54e-06	**4.44e-06**	0.0008	2951	2	0.06
		mGWO	**5.84e-08**	8.31e-05	1.54e-05	1.84e-05	0.0032	2910	4	**0.04**
		EEGWO	0.0301	5.5010	1.5118	1.3460	67.6946	**92**	5	0.10
		ROLGWO	3.99e-07	2.27e-05	7.87e-06	5.98e-06	**0.0007**	2885	3	0.06
		EGWO	4.89e-07	2.30e-05	**5.44e-06**	5.92e-06	0.0014	2906	**1**	**0.04**
F8	2	GWO	-1	**-0.9999**	-0.9999	**9.51e-06**	-0.9995	2961	**1**	0.05
		mGWO	-1	**-0.9999**	-0.9999	2.22e-05	-0.9983	2957	**1**	**0.04**
		EEGWO	-0.9839	-0.0186	-0.4499	0.3351	-5.79e-08	**209**	5	0.10
		ROLGWO	-1	-0.99996	-0.9999	1.01e-05	-0.9995	2955	**1**	0.07
		EGWO	-1	-0.99996	-0.9999	1.06e-05	**-0.9996**	2946	**1**	**0.04**
F8'	2	GWO	-1	-0.9999	-0.9999	8.49e-06	-0.9995	2973	**1**	**0.04**
		mGWO	-1	-0.9999	-0.9999	2.25e-05	-0.9982	2887	**1**	0.05
		EEGWO	-0.9796	-0.0189	-0.4815	0.2826	-6.95e-08	**212**	5	0.10
		ROLGWO	-1	-0.9999	-0.9999	**5.77e-06**	**-0.9996**	2920	**1**	0.07
		EGWO	-1	-0.9999	-0.9999	1.01e-05	-0.9991	2933	**1**	**0.04**
F9'	2	GWO	3	84.0001	7.0511	18.1119	7.0626	2509	3	0.06
		mGWO	3	**3.0071**	**3.0011**	**0.0020**	**3.0263**	2728	**1**	0.05
		EEGWO	3.0560	33.8879	7.5276	6.9715	579.4853	**159**	2	0.10
		ROLGWO	3	84.0014	19.2011	33.2415	19.2444	2873	4	0.07
		EGWO	3	84.0024	23.2503	35.9855	23.3530	2820	5	0.04
F10'	2	GWO	0.3979	0.4022	0.3982	0.0010	2.7846	2666	3	0.05
		mGWO	0.3979	0.3981	0.3979	3.72e-05	1.1951	2672	2	**0.04**
		EEGWO	0.4074	4.0957	2.2184	1.4071	51.5816	**70**	5	0.14

		ROLGWO	0.3979	0.4057	0.3988	0.0022	2.4922	2692	4	0.07
		EGWO	0.3979	**0.3980**	**0.3979**	**3.19e-05**	**0.4002**	2886	**1**	**0.04**
F11	3	GWO	**-3.8628**	-3.8574	-3.8619	0.0016	**-3.8605**	2827	2	0.09
		mGWO	**-3.8628**	-3.852	-3.8610	0.0026	-3.8566	2970	4	0.08
		EEGWO	-3.8591	-3.7401	-3.8133	0.0375	-0.4029	**1375**	5	0.15
		ROLGWO	-3.8627	-3.8529	-3.8615	0.0023	-3.8602	2901	3	0.18
		EGWO	-3.8627	**-3.8589**	**-3.8621**	**0.0012**	-3.8595	2960	**1**	**0.07**
F12'	2	GWO	-1.0316	**-1.0316**	**-1.0316**	3.33e-07	**-1.0316**	2896	**1**	**0.04**
		mGWO	-1.0316	**-1.0316**	**-1.0316**	1.05e-06	-1.0314	2804	**1**	0.05
		EEGWO	-1.0283	-0.8916	-1.0017	0.0288	-0.0098	**196**	5	0.10
		ROLGWO	-1.0316	-1.0299	-1.0315	0.0004	-0.9912	2785	4	0.07
		EGWO	-1.0316	**-1.0316**	**-1.0316**	**2.71e-07**	**-1.0316**	2869	**1**	**0.04**

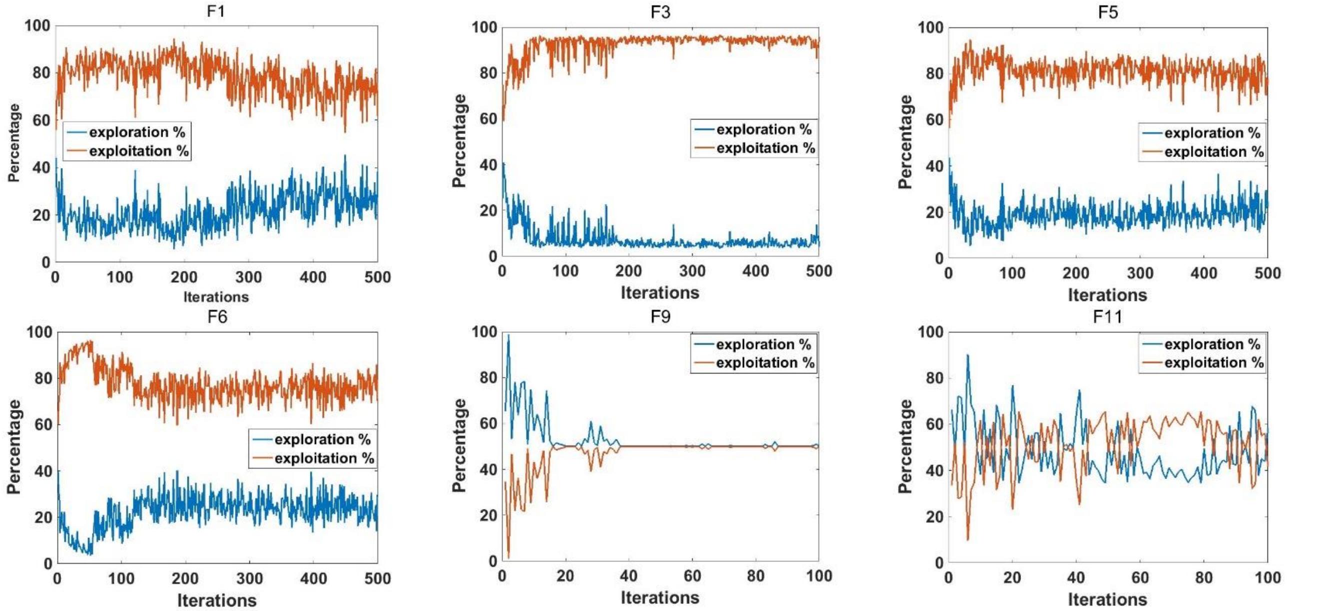
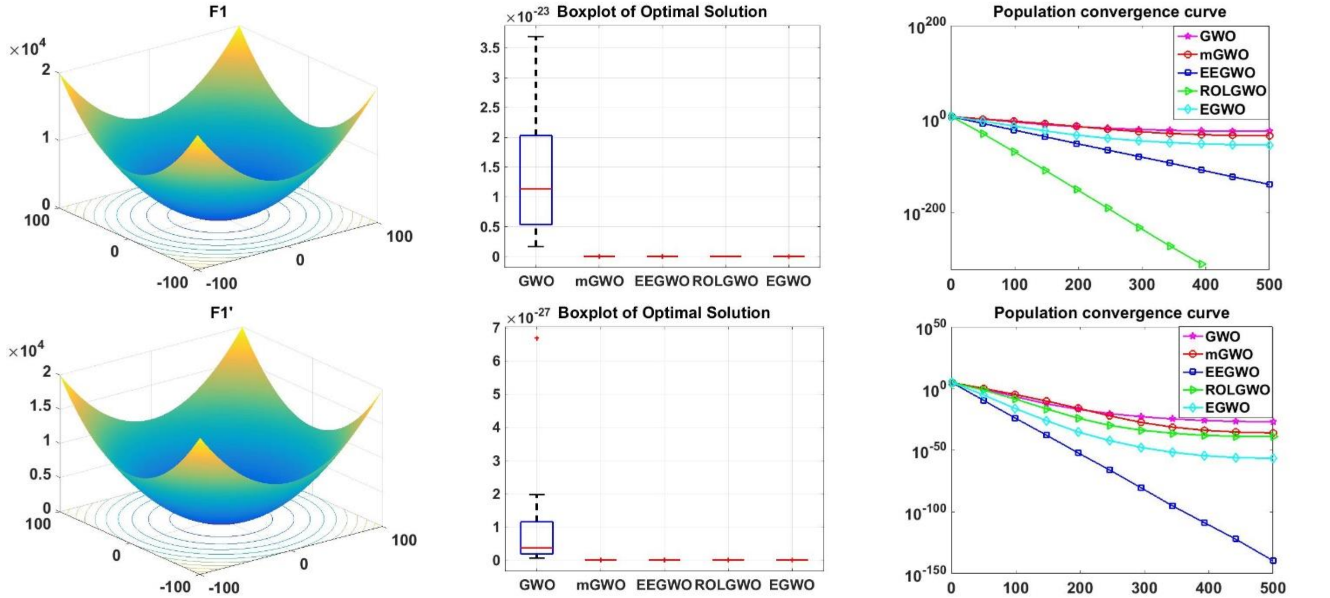


FIGURE 11. Exploration and exploitation phases for EGWO in 6 benchmark functions



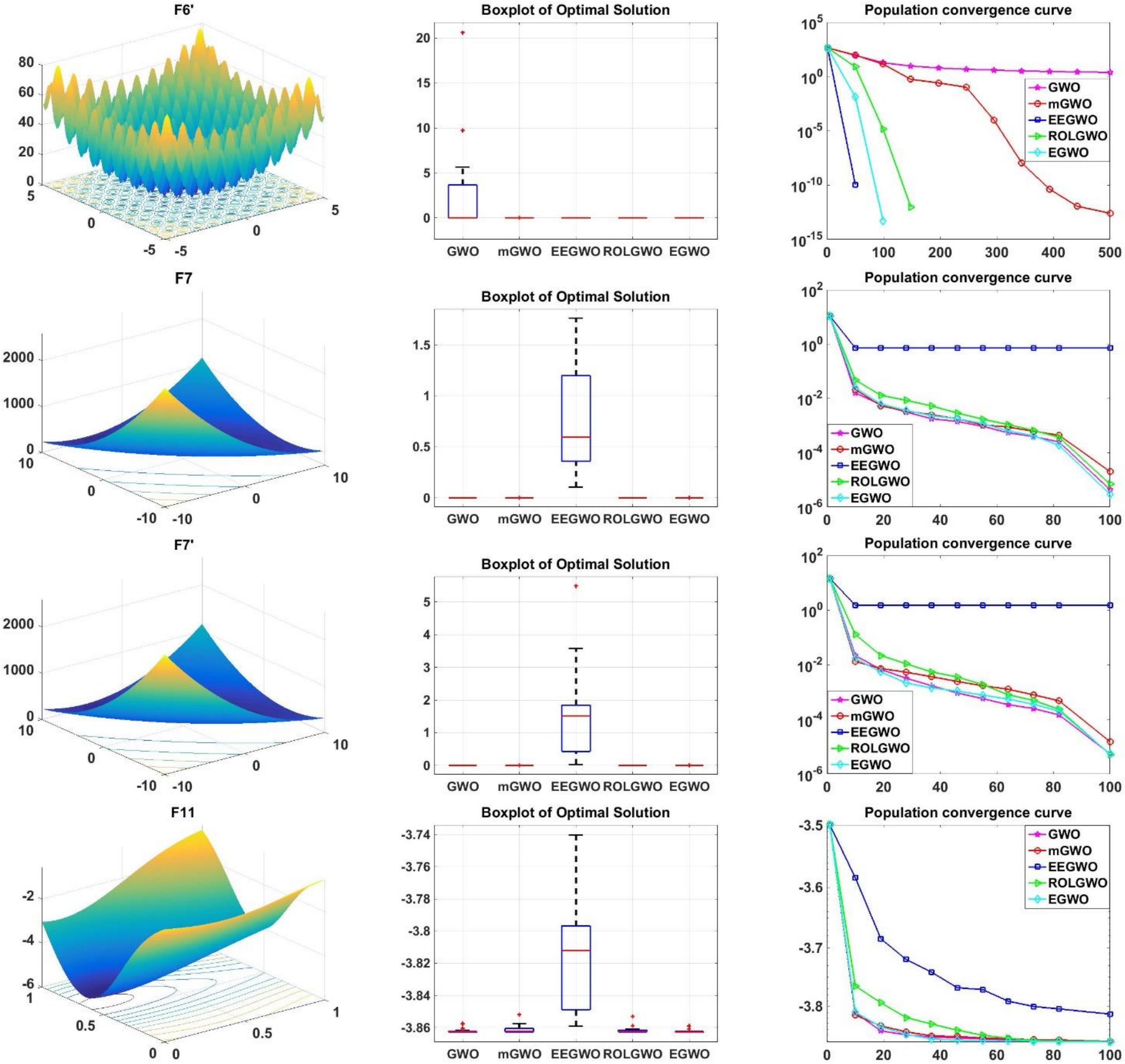


FIGURE 12. **Boxplot and population convergence**

4) ANALYSIS OF ALGORITHM CONVERGENCE

In this section, we consider the convergence performance of the population for the algorithm. From Tables 6-8, in the column of "f_{mean} of population", GWO performs the worst in most cases. The mGWO has better convergence performance than GWO. In Tables 6 and 7, EEGWO and ROLGWO perform well because the algorithms are always close to a certain position, so the convergence is faster, especially EEGWO. In Table 8, EEGWO performs the worst, even worse than GWO. The interesting point is that in Table 8, in the column of "number of evaluations", the number of valid evaluations of EEGWO is always the least, which indicates that the diversity of the grey wolf population has been lost, and the search can no longer be performed. The EGWO added with inner-layer encircle performs well. For example, in whatever type of benchmark function, its population average fitness value is better than other algorithms in most cases, which indicates that almost all grey wolves are already very close to the global optimum.

In order to further illustrate the above analysis, we show the population convergence graph of the algorithm in comparison, in some representative benchmark functions in the third column of Fig. 12. These representative benchmark functions cover three benchmark function types, including F1 and F1' for unimodal type functions, F6' for multimodal type functions, and F7, F7', and F11 for multimodal type functions with fixed dimensions. We plot the average fitness value of the population for each generation as a curve. As can be seen from the figure, in the convergence curve of F6', the convergence curves of EEGWO, EGWO and ROLGWO did not complete the last iteration, because these three algorithms have found the global optimal value before the maximum number of iterations preset in the experiment. It shows that the three algorithms have the fastest convergence speed, reaching full convergence in the 50th, 100th, and 150th generations. Moreover, ROLGWO has excellent convergence performance in F1, has a greatly reduced performance in F1'. EEGWO, whose convergence performance has always been excellent, has the worst performance when the position of the optimal solution is not 0, and falls into a locally optimal state prematurely. The convergence performance of EGWO and mGWO has been relatively stable, and EGWO is better than mGWO. EGWO performs well in convergence, which shows that the inner-layer encircle strategy is effective to strengthen the local exploitation ability of the algorithm. In each iteration, the search agent executing the inner-layer encircle command performs a fine search near the current optimal solution. Therefore, the current optimal solution provided by each iteration of EGWO is likely to be better than other algorithms. In other words, this strategy improved the convergence ability of the algorithm to a certain extent.

5) NON-PARAMETRIC TEST ANALYSIS

We are accumulating the ranking of each algorithm in 18 sets of experiments. The results are shown in Table 9. EGWO ranks the lowest in the sum and first overall. Therefore, we consider it a very competitive search algorithm. The worst was GWO, which ranked fifth. In addition, EGWO has a very short running time while maintaining competitiveness. As can be seen from the " Running time " column in Table 6-8, compared with GWO, the running time of the other three improved algorithms has increased, while the running time of EGWO is almost the same as or less than GWO. This is because the steps of EGWO are the same as GWO, but the calculation is a little bit less.

TABLE 9. Ranking of algorithms

Algorithm	Sum of ranks	Overall ranking
GWO	62	5
mGWO	56	4
EEGWO	53	3
ROLGWO	42	2
EGWO	33	**1**

TABLE 10. Ranking of algorithms on three types of functions

Algorithm	Unimodal function	Multimodal function	Fixed-dimension multimodal function
GWO	5,5,5,5,1	5,5,5,5,5	3,2,1,1,3,3,2,1
mGWO	4,4,4,4,2	4,4,4,4,4	4,4,1,1,1,2,4,1
EEGWO	2,1,2,1,5	1,1,1,1,1	5,5,5,5,2,5,5,5
ROLGWO	1,3,1,3,4	2,3,1,1,1	2,3,1,1,4,4,3,4
EGWO	3,2,3,2,3	3,2,1,1,1	1,1,1,1,5,1,1,1

Derrac et al. [61] proposed that for the evaluation of improved evolutionary algorithm performance, statistical tests should be performed. The purpose is to verify that the proposed improved algorithm has significant advantages over the comparison algorithms. In order to determine whether EGWO is significantly different from the best results of other algorithms in statistics, each row of data in Table 10 is used as a distribution column, and Wilcoxon rank-sum [62] is used at a significance level of 5%. Table 11 shows the *p*-values calculated by EGWO and other algorithms in the test. It can be seen from the table that the corresponding *p*-values of "EGWO vs GWO" and "EGWO vs mGWO" are less than 0.05, which shows that the superiority of EGWO is statistically significant compared to the two algorithms. That is to say, EGWO algorithm has higher optimization precision than these two algorithms. The *p*-value of "EGWO vs EEGWO" is 0.1116, which indicates that the performance of EGWO is significantly different from that of EEGWO, but the probability of error in this assertion is 11.16%. In the comparison of "EGWO vs ROLGWO", if the performance of EGWO is considered to be significantly different from that of ROLGWO, the probability of error in this assertion is 19.30%. Therefore, in general, the performance of EGWO is significantly superior to that of GWO and mGWO. Compared with EEGEO and ROLGWO, the superiority of EGWO is not significant.

TABLE 11. Wilcoxon rank-sum test

Algorithm comparison	*p*-value
EGWO vs GWO	0.0059
EGWO vs mGWO	0.0048
EGWO vs EEGWO	0.1116
EGWO vs ROLGWO	0.1930

B. EXPERIMENT 2: 3D SENSOR SIMULATION EXPERIMENT AND ANALYSIS

In order to verify that the EGWO algorithm can effectively deploy a 3D sensor network, we first set the deployment environment to a simple 3D surface. In order to verify that EGWO is highly adaptable compared to other deployment algorithms, complex 3D surfaces will be simulated as well. The comparison algorithms in the experiment are GWO, DEA (differential evolution algorithm) and ABC-DSS (artificial bee colony algorithm with dynamic search strategy), and their information is shown in Table 12. The parameters of the comparison algorithms are consistent with those in the literature [28,29,31]. It should be noted that in DEA, mutation factor $F = 0.9$ and cross-factor $CR = 0.1$. The maximum number of iterations for all algorithms in comparison is set to 300, the population size is set to 50, and the number of experiments is 20. The average results are compared for clarity.

TABLE 12. Description of the comparison algorithms

Algorithm	Reference	Coverage type	Time complexity
GWO	31	-	O(*TN*)
EGWO	This paper	-	O(*TN*)
ABC-DSS	28	Area coverage	O(TN^2)
DEA	29	Target coverage	O(*TN*)

1) SIMPLE 3D SURFACE SIMULATION EXPERIMENT

In this section, we use the more commonly used "saddle-shaped surface" as the deployment surface. The surface equation is shown in Eq. (24). In order to reduce the difficulty of simulation, the perceptual radius and surface area of the experiment are reduced. The perceptual radius of all nodes in the experiment is 0.4m, the communication radius is 0.8m, the 2D length of the surface is 2m and the width 2m, the total area of the surface is 6.20m^2. In order to increase the possibility of network connectivity and network coverage, we plan to place a fixed node at or near the center of each slope of the "saddle-shaped surface" and the connection between a slope and another slope, and hang them at a height from the surface 0.4m (the size of the node's perceived radius). The coordinates of the 6 nodes are (-0.7, -0.7, 0.4), (-0.7, 0, 0.89), (-0.7, 0.7, 0.4), (0.7, 0.7), 0.4), (0.7, 0, 0.89), and (0.7, -0.7, 0.4). The maximum coverage area of a node is $\pi(r^p)^2 = 0.5024$m^2. In the most ideal state, the entire surface needs $6.20/0.5024 = 13$, so we set the number of nodes to 13. In the experiment, according to the grid method mentioned in Section III B and the conditions for meshing of Eq. (10), the "saddle-shaped surface" was divided into 400 small mesh surfaces.

$$z = x^2 - y^2 \tag{24}$$

Figure 13 shows the location of the sensor nodes after the EGWO deployment when the number of nodes is 13. At this time, the network coverage has reached 44.26%, which is still far from the ideal full coverage. It can be seen that compared with the 2D plane, the coverage efficiency is lower due to the presence of perceived blind areas and unevenness of the coverage surface in the deployment of the 3D surface. Figures 14-18 describe the coverage optimization effect of 50 sensor nodes. Figure 14 shows the deployment effect of the nodes in a random spraying mode. The red dots are the locations of the nodes. It can be seen that the nodes are unevenly distributed. In order to understand the distribution and coverage of the nodes, we map them to a 2D grid. As shown in Figure 15, the grid where the black hollow circle is located means the area covered by the WSN. It can be seen that the coverage area is small, which is 3.40m^2, and the corresponding coverage rate is 54.73%. After using EGWO to optimize WSN deployment, the final deployment is shown in Figure 16. Figure 17 is its corresponding 2D grid overlay. Compared with the initial deployment, the nodes are more evenly distributed, the coverage area is larger, and the deployment effect is significant. At this time, the coverage area reaches 5.92m^2, and the coverage rate is 95.45% which is increased by 40.72%. Figure 18 shows the minimum spanning tree of the network generated by the Kruskal algorithm mentioned in [53]. It can be seen from the figure that the entire network is connected. The network can benefit from two points. Firstly, the algorithm has exerted its optimized performance. Secondly, six nodes fixed on the surface are set in the simulation experiment to make the network easier to communicate. The analysis shows that EGWO can provide an effective and optimized solution for the network deployment problem of 3D surfaces.

Figure 19 compares the coverage of WSN deployment optimization with the four algorithms where the experimental parameters are the same. It can be seen from the figure that after the algorithms converge, the WSN coverage of GWO optimization is 91.33%, ABC-DSS is 91.4%, and EGWO reaches 95.42%, which is 4.09% higher than GWO, meanwhile, it has an increase of 9.22% compared with the worst DEA (86.20%). It can be seen that the precision of the optimal solution of EGWO is the highest. This is because EGWO is based on the outer-layer encircle in the aspect of global exploration. Under the role of Tent mapping, it enriches the diversity of node deployment space and avoids node clustering. Therefore, the global exploration ability of the algorithm is improved to a certain extent. The DEA optimization precision evolved by the differential calculation of any two search agents is lower because the population diversity decreases in the later iterations, causing the algorithm to enter a premature convergence state. On the other hand, it also shows that DEA, which performs well in target coverage, might not be suitable working as the deployment model. It can be seen from Figure 19 that the coverage curve of EGWO is always above the other four curves, which indicates that EGWO has the fastest convergence speed. The reason why EGWO converges fast is that some search agents (wolves) search near the optimal solution (prey) instead of blind search, which improves the search efficiency and makes the search more thorough. In other words, the inner-layer encircle strategy proposed in this paper improves the local exploitation ability of the algorithm, so the search precision is high. Therefore, when EGWO is applied to WSN deployment, it can make the network coverage higher with fewer iterations. However, ABC-DSS maintains a similarly fast convergence speed with EGWO in the early stage of iteration. After 50 generations, the convergence speed is worse than EGWO. At the end of the iteration, with the exception of DEA, other algorithms do not fully converge, but instead tend to converge. This phenomenon occurs because the maximum number of iterations we set is small.

Figure 20 shows the comparison of the coverage of the four algorithms when the number of nodes is different. When the number of nodes is 80, the solution deployed by EGWO can cover the entire monitoring area with 100% coverage. ABC-DSS requires 90, while the remaining two algorithms require 100 nodes. When deploying the network with other numbers of nodes, EGWO's coverage optimization capabilities are consistently better than those of the other three algorithms. Therefore, under the same coverage requirements, the number of nodes required for EGWO deployment is smaller, thereby saving deployment costs. It can also be seen from the figure that as the number of nodes increases, the coverage of each algorithm also increases but becomes slower. This is because the increasing number of nodes leads to the occurrence of redundancy and the optimization efficiency decreases. Between 20 nodes and 30 nodes, the coverage increases at the fastest pace, because at this time the fewer nodes results in more coverage holes. Hence the increase in the number of nodes at this region brings the greatest deployment benefit. Taking EGWO as an example, when the number of nodes is 20, the coverage rate is 58.91%, and when it is 30, it reaches 74.68%, an increase of 15.77%.

TABLE 13. Statistics of network connectivity

Number of nodes	Algorithm	Network connectivity success rate	Average coverage with network connectivity
30	GWO	0.3	0.6949
	EGWO	0.65	**0.7395**
	ABC-DSS	0.35	0.7086
	DEA	0.35	0.6671
40	GWO	0.75	0.8253
	EGWO	**1**	**0.8723**
	ABC-DSS	0.85	0.8389
	DEA	0.75	0.7944
50	GWO	**1**	0.9033
	EGWO	**1**	**0.9542**
	ABC-DSS	**1**	0.9190
	DEA	**1**	0.8720

In order to compare the performance of each deployment algorithm on network connectivity, each deployment algorithm was tested 20 times under the number of nodes of 30, 40, and 50, and the resulting deployment scheme was judged for network connectivity. After that, the average coverage rate of each deployed network in the connected case was calculated, and the results are shown in Table 13. As can be seen from the table, when the number of nodes is less than 40, all deployment algorithms cannot ensure network connectivity. EGWO can ensure network connectivity under 40 nodes, while other deployment algorithms can ensure connectivity only at 50 nodes. When other deployment algorithms are at 50 nodes, the main reason for the network to be connected is that there are too many nodes and the nodes appear redundant, which increases the possibility of network connectivity. In general, in the case of different node deployments, the connectivity rate of EGWO and the average coverage rate under network connectivity are higher than other deployment algorithms. The deployment of WSN is particularly sensitive to node locations, especially network connectivity. Assume that the network coverage rate under a certain deployment scheme is very high, but because the network is not connected, then this deployment scheme is also flawed. In EGWO's inner-layer encircle mechanism, the search agent always updates the location near the current optimal search agent. Therefore, when deploying WSN, the search agent that executes the inner-layer encircle strategy only updates the positions of two nodes. Therefore, the position of the node changes very little, which reduces the damage to the network connectivity, thereby reducing the possibility of network disconnection. As the iteration progresses, the search agent approaches the optimal solution position, and the nodes move closer to each other rather than discretely. Therefore, with the same number of nodes, the connectivity of EGWO is higher than other deployment algorithms, and with a certain number of sensor nodes, EGWO can ensure the connectivity of WSN.

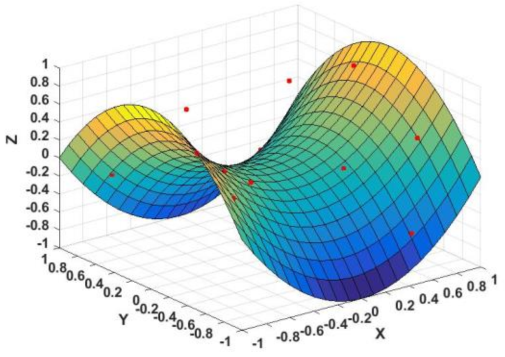


FIGURE 13. 13 nodes deployment diagram

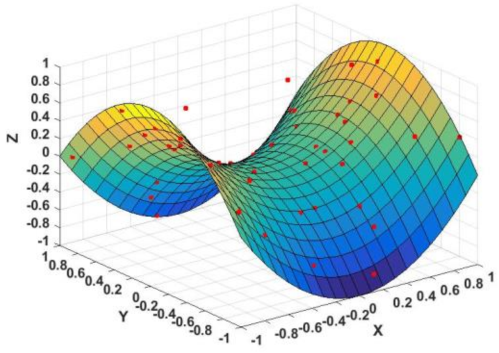


FIGURE 14. Initial deployment

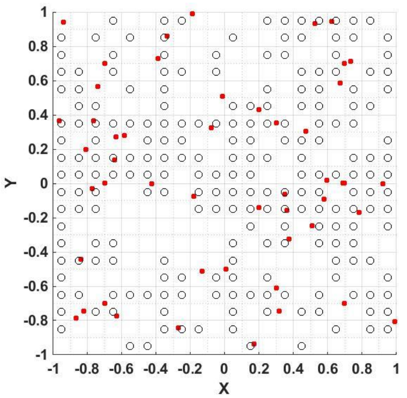


FIGURE 15. 2D perspective view of initial deployment

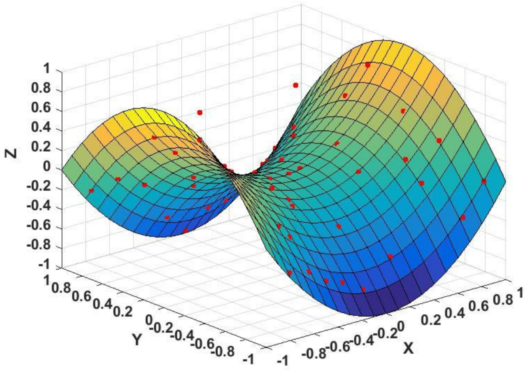


FIGURE 16. Final deployment

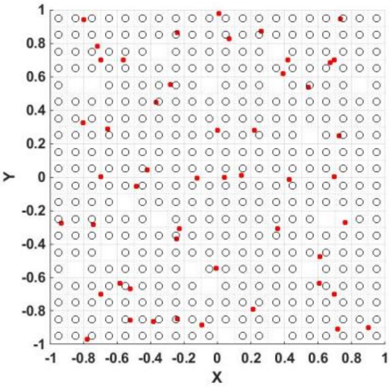


FIGURE 17. 2D perspective of final deployment

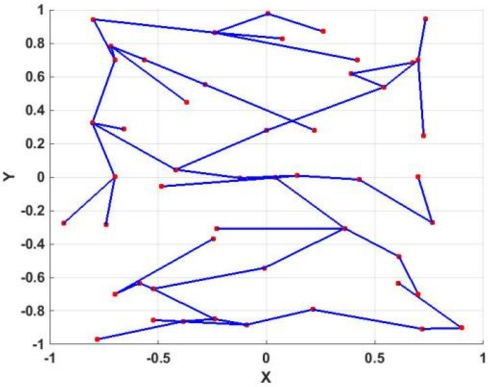


FIGURE 18. WSN minimum spanning tree

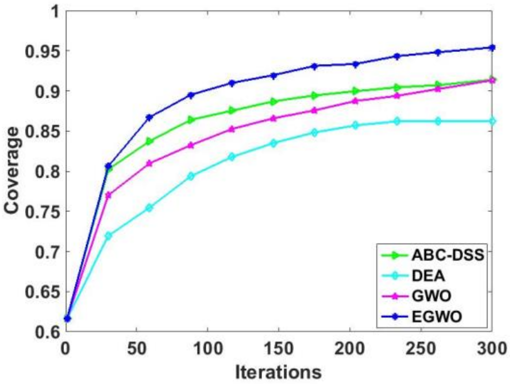


FIGURE 19. Coverage comparison

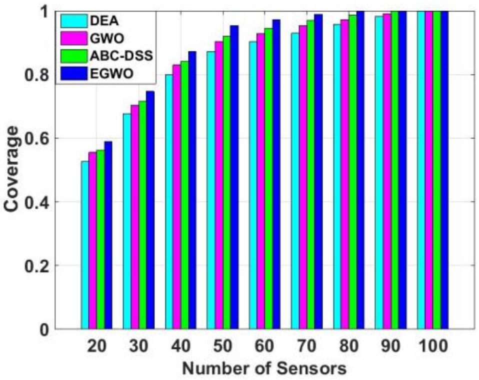


FIGURE 20. Number of nodes and coverage

2) COMPLEX 3D SURFACE SIMULATION EXPERIMENT

In order to better reflect the deployment performance of EGWO, we use a more complex surface as the deployment surface. The surface equation is shown in Eq. (25). In the experiment, the perceptual radius of all nodes is 1m, the communication radius is 2m, the 2D length of the surface is 10m and the width 10m, the total area of the surface is 122.07m^2, and the maximum coverage area of a node is $\pi(r^p)^2 = 3.14$m^2. In the most ideal state, the entire surface needs $122.07/3.14 = 39$. It can be known from the simple surface deployment experiment that the coverage efficiency is low due to the perception of blind spots and the unevenness of the coverage. In this experiment, we will use 100 nodes to deploy WSN, and the surface will be divided into 400 small mesh surfaces.

$$z = \sin(y)\sin(x) \tag{25}$$

Figures 21-24 describe the coverage optimization effect of 100 sensor nodes. Figure 21 shows the deployment effect of the node in a random spraying mode. In order to see the coverage of the WSN clearly, Figure 22 is a 2D map of the sensor nodes. It can be seen that the total area of the grid where the black hollow circle is located is small, covering an area of 74.60m^2, and the corresponding coverage is 61.11%. After using EGWO to optimize WSN deployment, the final deployment is shown in Figure 23. Figure 24 is its corresponding 2D grid overlay. Compared with the initial deployment, the coverage area is larger and the deployment effect is more significant. The coverage is 89.73%, increased by 28.62%, and the coverage area is 109.53m^2. It is worth mentioning that when deploying WSN using EGWO, the nodes are more difficult to deploy on the convex parts of the surface. As shown in Figure 23, there are many blank spaces in the figure, and the blank spaces correspond to the convex parts of the surface in Figure 22. It might be possible to fix some nodes on the convex part of the surface to solve this problem, which will remain as the future work.

Figure 25 is a diagram to compare WSN deployment optimization with the four algorithms with the same experimental parameters. As can be seen from the figure, the convergence curve of the algorithm is almost the same as that of Figure 19, but there are still differences. After the algorithms converge, GWO optimized WSN coverage can reach 87.21%, while EGWP can reach 89.62%, which is 2.41% higher than GWO, 0.82% higher than ABC-DSS which achieves 88.80%, while DEA only reaches 85.37%. Apparently, DEA has the worst performance. It can be seen that among the four algorithms, EGWO maintains the best deployment performance, which indicates that EGWO with its inner and outer layers encircle strategies has a good optimization performance even when the network is deployed on a complex surface. Figure 25 differs from Figure 19 in that the four deployment algorithms are nearly converged after 300 iterations. This is because the complex surface causes the optimization performance of the algorithms to decline, and quickly falls into a state of local optimum. DEA, which performs poorly on simple surface deployment, has converged after about 150 iterations in complex surface deployment, making it impossible to perform a useful search as it gradually falls into a local optimum. The performance of the other three algorithms is almost consistent with the experimental results of simple surface deployment. The experiment shows that EGWO has good adaptability, can still maintain good deployment performance even in a complex surface, and the inner-layer encircle strategy and Tent mapping have made great contributions.

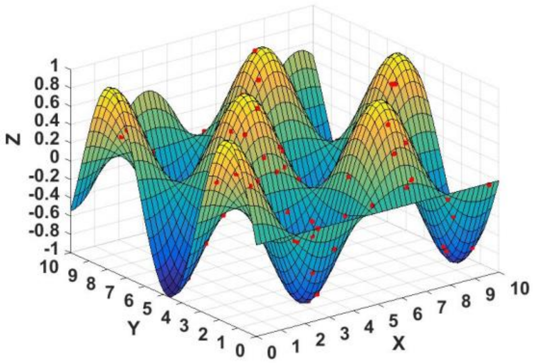


FIGURE 21. Initial deployment

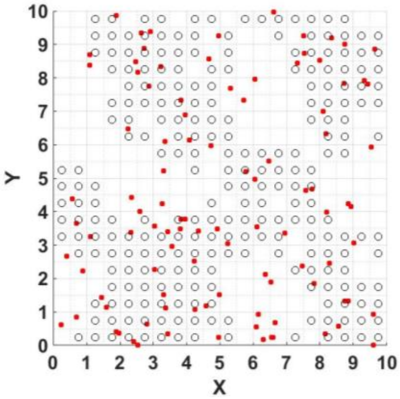


FIGURE 22. 2D perspective view of initial deployment

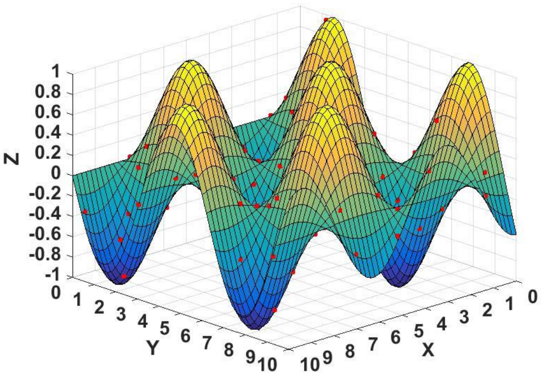


FIGURE 23. Final deployment

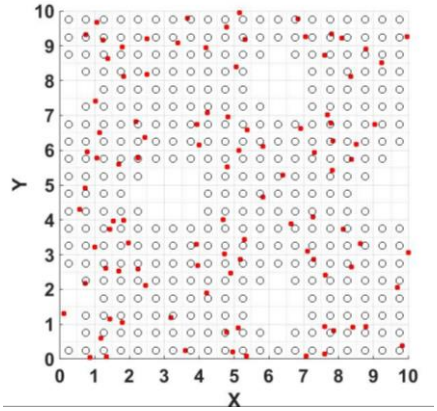


FIGURE 24. 2D perspective of final deployment

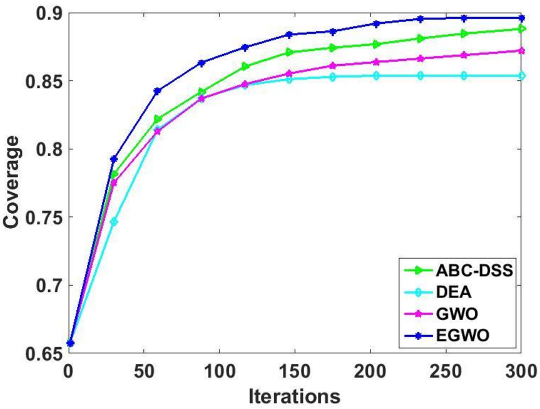


FIGURE 25. Coverage comparison

In this part of experiments, we compare our proposed algorithm with some existing algorithms. The first experiment is to test the exploitation ability, exploration ability, convergence, stability and optimization precision of the algorithm on the benchmark functions of unimodal, multimodal and multimodal fixed dimensions. On the one hand, in addition to analyzing the excellent performance of the EGWO algorithm, it is also verified that our proposed strategies have a certain promotion effect on the local exploitation ability and global exploration ability of the algorithm. In addition, the reasons for the excellent performance of EEGWO and ROLGWO algorithms are analyzed, as well as their shortcomings. In second experiment, EGWO and other swarm intelligence deployment algorithms are tested in a 3D surface network deployment to demonstrate that the EGWO still has excellent performance in practical applications. In simple 3D surface deployment, by fixing some nodes, when the number of deployed nodes is sufficient, EGWO's optimized deployment scheme can ensure network connectivity, while other deployment methods cannot. In addition, EGWO's optimized deployment plan has the highest coverage of WSN, so the entire surface can be completely covered with the minimum number of nodes. Using complex 3D surfaces, it can be proven that EGWO is still the best deployment algorithm, which can provide the deployment plan with the largest coverage.

From the above experimental results, we can see that the advantages of EGWO compared with other comparison algorithms can be summarized as follows:

1. In the benchmark function experiment, compared with other algorithms, EGWO shows advantages in both convergence performance and optimization performance. In general, EGWO is better than GWO and mGWO, and has good competitiveness compared with EEGWO and ROLGWO. In the evaluation of algorithm stability, EGWO showed satisfactory results.

2. In the WSN deployment experiment, the optimization performance (WSN coverage) and convergence performance of EGWO are better than GWO, DEA, and ABC-DSS. At the same time, the probability of network connectivity deployed by EGWO is higher than other deployment algorithms.

VII. CONCLUSION AND FUTURE WORK

The main work of this paper is to propose an EGWO (enhanced grey wolf optimizer) based on the GWO, which is used to deploy a WSN on a 3D surface. By strengthening the exploitation and exploration abilities of EGWO, respectively, the convergence performance and optimization precision of the algorithm are improved. Firstly, the grey wolf group is divided into two groups: the inner-layer group and the outer-layer group, and the inner-layer group improves the exploitation of the algorithm through a fine local search. Secondly, the outer-layer group is consistent with GWO and continues to conduct global exploration. Finally, in order to improve the global exploration ability of EGWO, Tent mapping was introduced to enrich the diversity of the grey wolf population and reduce the possibility of the algorithm falling into a local optimum. In order to verify the effectiveness of EGWO, 12 well-known benchmark functions are used to test the performances of the algorithm. Comparison algorithms include GWO, mGWO, EEGWO and ROLGWO. Firstly, the test results of the unimodal benchmark functions and multimodal type benchmark functions show that the exploitation and exploration abilities of EGWO have been significantly improved. Secondly, the experimental results of the optimization precision, convergence and stability of the algorithm show that EGWO is a competitive algorithm and has an excellent performance in most benchmark functions. Finally, the results of non-parametric test experiments prove that EGWO is superior to GWO and mGWO, and EGWO can provide a competitive performance compared to EEGWO and ROLGWO.

In the WSN deployment of 3D surfaces, we have improved the method of estimating the perceptual blind zone by using algebraic geometry, improved the grid method often used in 2D deployment, and applied them to the calculation of the network coverage of 3D surfaces. EGWO is then applied to the network deployment of 3D surfaces, and compared with the three deployment methods of GWO, ABC-DSS and DEA. The simulation experiment results of WSN deployment show that EGWO improves the coverage of WSN. Compared with other deployment methods, EGWO deployment has the highest network coverage and can cover the entire surface with the fewest nodes, thereby saving network deployment costs. With a certain number of nodes, it can ensure network connectivity in a simple surface deployment, and provides better solutions to network deployment. In general, EGWO is a very competitive search algorithm in terms of its performance using benchmark functions and 3D WSN deployment in practical applications.

In future work, we will improve the optimization performance of the EGWO on complex surfaces, and study measures to ensure network connectivity.

REFERENCES

[1] Li Y, Chen C S, Chi K *et al*., "Two-tiered relay node placement for WSN-based home health monitoring system," *Peer-to-Peer Networking and Applications*., vol. 12, no. 3, pp. 589-603, May. 2019.

[2] Li M and Lin H J, "Design and implementation of smart home control systems based on wireless sensor networks and power line communications," *IEEE Transactions on Industrial Electronics*., vol. 62, no. 7, pp. 4430-4442, Jul. 2015.

[3] Topcuoglu H R, Ermis M, Sifyan M *et al*., "Positioning and utilizing sensors on a 3-D terrain part II—Solving with a hybrid evolutionary algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*., vol. 41, no. 4, pp. 470-480, Aug. 2011.

[4] Alam S M N and Haas Z J, "Coverage and connectivity in three-dimensional networks with random node deployment," *Ad Hoc Networks*., vol. 34, pp. 157-169, Nov. 2015.

[5] Houssein E H, Saad M R, Hussain K *et al*., "Optimal Sink Node Placement in Large Scale Wireless Sensor Networks Based on Harris' Hawk Optimization Algorithm," *IEEE Access*., vol. 8, pp. 19381-19397, Jan. 2020.

[6] Nguyen T T, Pan J S, Dao T K *et al*., "An improved flower pollination algorithm for optimizing layouts of nodes in wireless sensor network," *IEEE Access*., vol. 7, pp. 75985-75998, Jun. 2019.

[7] Zhu C, Leung V C M, Yang L T *et al.*, "Collaborative location-based sleep scheduling for wireless sensor networks integrated with mobile cloud computing," *IEEE Transactions on Computers*., vol. 64, no. 7, pp. 1844-1856, Jul. 2014.

[8] Hai D T and Le Vinh T, "Novel fuzzy clustering scheme for 3D wireless sensor networks," *Applied Soft Computing*., vol. 54, pp. 141-149, May. 2017.

[9] Han G, Zhang C, Shu L *et al., "*A survey on deployment algorithms in underwater acoustic sensor networks," *International Journal of Distributed Sensor Networks*., [Online]. Available: https://journals.sagepub.com/doi/pdf/10.1155/2013/314049.

[10] Mini S, Udgata S K, Sabat S L *et al*., "Sensor deployment and scheduling for target coverage problem in wireless sensor networks," *IEEE sensors journal*., vol. 14, no. 3, pp. 636-644, Mar. 2014.

[11] Deng X, Yu Z, Tang R *et al.*, "An Optimized Node Deployment Solution Based on a Virtual Spring Force Algorithm for Wireless Sensor Network Applications," *Sensors*., vol. 19, no. 8, pp. 1817-1831, Apr. 2019.

[12] Benatia M A, Sahnoun M, Baudry D *et al.*, "Multi-objective WSN deployment using genetic algorithms under cost, coverage, and

connectivity constraints," *Wireless Personal Communications.*, vol. 94, no. 4, pp. 2739-2768, Feb. 2017.

[13] Xu Y, Ding O, Qu R *et al.*, "Hybrid multi-objective evolutionary algorithms based on decomposition for wireless sensor network coverage optimization," *Applied Soft Computing.*, vol. 68, pp. 268-282, Jul. 2018.

[14] Xiang Y, Xuan Z, Tang M *et al.*, "3D space detection and coverage of wireless sensor network based on spatial correlation," *Journal of Network and Computer Applications.*, vol. 61, pp. 93-101, Feb. 2016.

[15] Hajizadeh N, Jahanbazi P, Javidan *et al.*, "R Controlled deployment in wireless sensor networks based on a novel Multi Objective Bee Swarm Optimization algorithm," *2018 3rd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, Bam, Iran, 2018, pp. 1-5.

[16] Kong L, Zhao M, Liu X Y *et al.*, "Surface coverage in sensor networks." *IEEE Transactions on Parallel and Distributed Systems.*, vol. 25, no. 1, pp. 234-243, Jan. 2013.

[17] Unaldi N, Temel S, Asari V K *et al.*, "Method for optimal sensor deployment on 3D terrains utilizing a steady state genetic algorithm with a guided walk mutation operator based on the wavelet transform," *Sensors.*, vol. 12, no. 4, pp. 5116-5133, Apr. 2012.

[18] Li F, Luo J, Wang W *et al.*, "Autonomous Deployment for Load Balancing k-Surface Coverage in Sensor Networks,". *IEEE Transactions on Wireless Communications.*, vol. 14. no. 1, pp. 279-293, Jan. 2015.

[19] Anand N, Ranjan R, Rai B S, *et al.*, "A novel computational geometry-based node deployment scheme in 3D wireless sensor network," *International Journal of Sensor Networks.*, vol. 25, no. 3, pp. 135-145, Nov. 2017.

[20] Tan L, Tang X, Yang M *et al.*, "A Weighted Voronoi Diagram Based Self-Deployment Algorithm for Heterogeneous Mobile Sensor Network in Three-Dimensional Space," *China Conference on Wireless Sensor Networks*, Chongqing, China, 2019, pp. 19-34.

[21] Miao C, Dai G, Zhao X *et al.*, "3D self-deployment algorithm in mobile wireless sensor networks," *International Journal of Distributed Sensor Networks.*, vol. 11, no. 4, pp. 721921-721931, Apr. 2015.

[22] Nazarzehi V and Savkin A V, "Decentralized control of mobile three-dimensional sensor networks for complete coverage self-deployment and forming specific shapes," *2015 IEEE Conference on Control Applications (CCA)*, Sydney, NSW, Australia, 2015, pp. 127-132.

[23] Li X, Ci L, Yang M *et al.*, "Deploying three-dimensional mobile sensor networks based on virtual forces algorithm," *China Conference on Wireless Sensor Networks*, Huangshan, China, 2012, pp. 204-216.

[24] Boufares N, Minet P, Khoufi I *et al.*, "Covering a 3D flat surface with autonomous and mobile wireless sensor nodes," *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Valencia, Spain, 2017, pp. 1628-1633.

[25] Boufares N, Saied Y B, Saidane L A *et al.*, "Improved Distributed Virtual Forces Algorithm for 3D Terrains Coverage in Mobile Wireless Sensor Networks," *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, Aqaba, Jordan, 2018, pp. 1-8.

[26] Temel S, Unaldi N, Kaynak O *et al.*, "On deployment of wireless sensors on 3-D terrains to maximize sensing coverage by utilizing cat swarm optimization with wavelet transform," *IEEE Transactions on Systems, Man, and Cybernetics: Systems.*, vol. 44, no. 1, pp. 111-120, Jan. 2014.

[27] Ru J, Jia Z, Yang Y *et al.*, "A 3D Coverage Algorithm Based on Complex Surfaces for UAVs in Wireless Multimedia Sensor Networks," *Sensors.*, vol. 19, no. 8, pp. 1902-1921, Apr. 2019.

[28] Yang H, Li X, Wang Z *et al.*, "A novel sensor deployment method based on image processing and wavelet transform to optimize the surface coverage in WSNs," *Chinese Journal of Electronics.*, vol. 25, no. 3, pp. 495-502, May. 2016.

[29] Sun S, Sun L, Chen S *et al.*, "Research on the target coverage algorithms for 3D curved surface," *Chaos, Solitons & Fractals.*, vol. 89, pp. 397-404, Mar. 2016.

[30] Dang X, Shao C, Hao Z *et al.*, "Target Detection Coverage Algorithm Based on 3D-Voronoi Partition for Three-Dimensional Wireless Sensor Networks," *Mobile Information Systems.*, to be published. DOI: 10.1155/2019/7542324.

[31] Mirjalili S, Mirjalili S M, Lewis A. *et al.*, "Grey wolf optimizer," *Advances in engineering software.*, vol. 69, pp. 46-61, Jan. 2014.

[32] Mirjalili S, "How effective is the Grey Wolf optimizer in training multi-layer perceptrons," *Applied Intelligence.*, vol. 43, no. 1, pp. 150-161, Jan. 2015.

[33] Hadavandi E, Mostafayi S, Soltani P *et al.*, "A Grey Wolf Optimizer-based neural network coupled with response surface method for modeling the strength of siro-spun yarn in spinning mills," *Applied Soft Computing.*, vol. 72, pp. 1-13, Nov. 2018.

[34] Li S X and Wang J S, "Dynamic modeling of steam condenser and design of PI controller based on grey wolf optimizer," Mathematical Problems in Engineering., [Online]. Available: http://downloads.hindawi.com/journals/mpe/2015/120975.pdf.

[35] Komathi C and Umamaheswari M G, "Design of Gray Wolf Optimizer Algorithm-Based Fractional Order PI Controller for Power Factor Correction in SMPS Applications," *IEEE Transactions on Power Electronics.*, vol. 35, no. 2, pp. 2100-2118, Feb. 2019.

[36] Daneshvar S M M H, Mohajer P A A, Mazinani S M *et al.*, "Energy-Efficient Routing in WSN: A Centralized Cluster-Based Approach via Grey Wolf Optimizer," *IEEE Access.*, vol. 7, pp. 170019-170031, Jan. 2020.

[37] Lipare A, Edla D R, Kuppili V *et al.*, "Energy efficient load balancing approach for avoiding energy hole problem in WSN using Grey Wolf Optimizer with novel fitness function," *Applied Soft Computing.*, [Online]. Available: https://doi.org/10.1016/j.asoc.2019.105706.

[38] Cai Z, Gu J, Luo J, *et al.*, "Evolving an optimal kernel extreme learning machine by using an enhanced grey wolf optimization strategy," *Expert Systems with Applications.*, [Online]. Available: https://doi.org/10.1016/j.eswa.2019.07.031.

[39] Heidari A A, Aljarah I, Faris H *et al.*, "An enhanced associative learning-based exploratory whale optimizer for global optimization," *Neural Computing and Applications.*, to be published. DOI: 10.1007/s00521-019-04015-0.

[40] Sultana U, Khairuddin A B, Mokhtar A S, *et al.* "Grey wolf optimizer based placement and sizing of multiple distributed generation in the distribution system," *Energy.*, vol. 111, pp. 525-536, Sep. 2016.

[41] Saxena A, Kumar R, Das S *et al.*, "β-Chaotic map enabled grey wolf optimizer," *Applied Soft Computing.*, vol. 75, pp. 84-105, Feb. 2019.

[42] Mittal N, Singh U, Sohi B S et al., "Modified grey wolf optimizer for global engineering optimization," *Applied Computational Intelligence and Soft Computing.*, [Online]. Available: http://downloads.hindawi.com/journals/acisc/2016/7950348.pdf

[43] Jayabarathi T, Raghunathan T, Adarsh B R *et al.*, "Economic dispatch using hybrid grey wolf optimizer," *Energy.*, vol. 111, pp. 630-641, Sep. 2016.

[44] Abdel-Basset M, El-Shahat D, El-henawy I *et al.*, "A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection," *Expert Systems with Applications.*, to be published. DOI: 1 0.1016/j.eswa.2019.112824.

[45] Long W, Jiao J, Liang X *et al.*, "An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization," *Engineering Applications of Artificial Intelligence.*, vol. 68, pp. 63-80, Jan. 2018.

[46] Long W, Jiao J, Liang X *et al.*, "A random opposition-based learning grey wolf optimizer," *IEEE Access.*, vol. 7, pp. 113810-113825, Aug. 2019.

[47] Al-Betar M A, Awadallah M A, Krishan M M *et al.*, "A non-convex economic load dispatch problem with valve loading effect using a hybrid grey wolf optimizer," *Neural Computing and Applications.*, [Online]. Available: https://doi.org/10.1007/s00521-019-04284-9.

[48] Shilaja C and Arunprasath T, "Internet of medical things-load optimization of power flow based on hybrid enhanced grey wolf optimization and dragonfly algorithm," *Future Generation Computer Systems.*, vol. 98, pp. 319-330, Sep. 2019.

[49] Zhao X, Zhang X, Cai Z, *et al.*, "Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients," *Computational biology and chemistry*, vol. 78, pp. 481-490, Feb. 2019.

[50] Saxena A, Kumar R, Mirjalili S *et al.*, "A harmonic estimator design with evolutionary operators equipped grey wolf optimizer," *Expert Systems with Applications.*, [Online]. Available: https://doi.org/10.1016/j.eswa.2019.113125

[51] Wang Z, Xie H, He D *et al.*, “Wireless Sensor Network Deployment Optimization Based on Two Flower Pollination Algorithms,” *IEEE Access.*, vol. 7, pp. 180590-180608, Dec. 2019.

[52] Graph theoretic methods in multiagent networks, Mesbahi M and Egerstedt M, eds., Princeton, NJ, USA: Univ. Princeton Press, 2010. [Online]. Available: https://pdfs.semanticscholar.org/7208/0c14258cf 8ca884329d50fdbe51790670b5a.pdf.

[53] Zhou Y, Wu J, Chen Q *et al.*, “Kruskal algorithm based assessment on power system structural vulnerability,” *Power System Technology.*, vol. 37, no. 11, pp. 3172-3177, Nov. 2013.

[54] Toreini, Ehsan, Maryam Mehrnejad *et al.*, “A novel method in fuzzy data clustering based on chaotic PSO,” *2011 International Conference for Internet Technology and Secured Transactions*, Abu Dhabi, United Arab Emirates, 2011, pp. 335-340.

[55] Kaur G and Arora S, “Chaotic whale optimization algorithm,” *Journal of Computational Design and Engineering.*, vol. 5, no. 3, pp. 275-284, Jul. 2018.

[56] Global optimization, Törn, Aimo, and Antanas Žilinskas, Heidelberg, Berlin, Germany: Springer, 1989. [Online]. Available: https://doi.org/10.1007/3-540-50871-6_8

[57] Hedar A R, “Test functions for unconstrained global optimization,” *System Optimization Laboratory, Kyoto University*., [Online]. Available: http://www-optima. amp. i. kyotou. ac. jp/member/student/hedar/Hedar_files/TestGO. htm, 2013.

[58] Hedar, A.R. and Fukushima M *et al.*, “Minimizing multimodal functions by simplex coding genetic algorithm,” *Optimization Methods and Software*., vol. 18, no. 3, pp. 265-282, Feb. 2003.

[59] Hussain K, Salleh M N M, Cheng S *et al.*, “On the exploration and exploitation in popular swarm-based metaheuristic algorithms,” *Neural Computing and Applications.*, vol. 31, no. 11, pp. 7665-7683, Nov. 2019.

[60] Hashim F A, Houssein E H, Mabrouk M S *et al.*, “Henry gas solubility optimization: A novel physics-based algorithm,” *Future Generation Computer Systems.*, vol. 101, pp. 646-667, Dec. 2019.

[61] Derrac J, García S, Molina D *et al.*, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” *Swarm and Evolutionary Computation.*, vol. 1, no. 1, pp. 3-18, Feb. 2011.

[62] Individual comparisons by ranking methods, Wilcoxon F, New York, NY, USA: Springer, 1992. [Online]. Available: https://xs.scihub.ltd/https://doi.org/10.1007/978-1-4612-4380-9_16