ABSTRACT In order to relieve the problem of unbalanced energy consumption of sensor nodes near the base station in the wireless sensor network, this paper proposes a mobile multi-sink nodes path planning algorithm with energy balance (hexHPSO). An optimization model is established by considering the energy consumption of each group, network lifetime and movement path of the mobile sink nodes. Meanwhile a hybrid positive and negative particle swarm optimization algorithm (HPNPSOA) is proposed to solve the optimization model to obtain a path with optimal grid traversal order and optimal parking position. Compared with the DOSM algorithm, GLRM algorithm and RWM algorithm, the hexHPSO algorithm improves the network lifetime by 68%. The experimental results show that the hexHPSO algorithm can effectively balance the energy consumption, alleviate hotspot phenomenon and extend the network lifetime.

INDEX TERMS Wireless Sensor Networks; Mobile multi-Sink Nodes; Path Planning; Hybrid Positive and Negative Particle Swarm Optimization; Network Lifetime

I. INTRODUCTION

The many-to-one transmission mode results in consuming a large amount of energy near the base station in wireless sensor networks which makes many nodes die prematurely and ultimately shortens the network lifetime. In order to solve the above problems effectively, mobile sink nodes are employed to collect data in the network [1]-[2]. However, the mobility of the sink node increases the complexity of building routes as well as the latency of networks. On the one hand, it results in variety constantly of route of data transmission. On the other hand, it makes the movement path non-negligible, which means that the mobile delay should be considered. The researches pay attention to the mobile sink node mainly focus on two aspects: route optimization and path planning [3]-[9]. These works also can be divided into four categories: random movement model, controlled movement model, geographic movement model and predictive movement model.

The random movement model means that the moving direction and moving speed of the mobile sink node are randomly selected each time the movement is made, so the model has strong uncertainty. The RWM algorithm proposed in [10] is improved based on this model, so that the mobile sink node saves a first-in first-out list M that stores the area number that has been visited. Thus the next parking position of the mobile sink node is randomly selected for the area number that has not appeared in M. The proposed algorithm reduces the repeated selection of the parking position to some extent and alleviates hotspot phenomenon, but it still causes unbalanced energy consumption and large network delay. This is because any improvement based on this model cannot change the randomness in essence, therefore this model has significant limitations.

The controlled movement model refers to the control of the movement of the sink node using a certain control mechanism (such as buffer overflow time, etc.). Reference [11] decomposes the monitoring area into a plurality of triangles to obtain a circle passing through the three vertices of the triangle and the center of each circle serves as the parking position of the mobile sink node. Then the next parking position is determined by the greedy algorithm. In fact, the path determined by the greedy algorithm belongs to local optimization path, which results in a long movement path and a large network delay. The GLRM algorithm [12] divides the monitoring area into multiple virtual square grids and selects a cluster head within each grid. The center line of
the monitoring area serves as the movement path of the mobile sink node to establish the approximate grid-like routes from the cluster head to the sink node. The proposed algorithm causes the nodes in the grid near the center line to consume much more energy than other nodes. The controlled movement model is either complicated by control mechanisms or difficult to construct routes.

The geo-movement model means that the movement is limited by the actual geographical environment, and the mobile sink node encounters obstacles or boundary during the movement. In [13], a route planning method combining description route by use of Bezier curve and improved particle swarm optimization algorithm is proposed, which is aim to design an efficient algorithm focusing on how to avoid obstacles in route planning.

The predictive movement model means that the sensor node knows the moving path of the mobile sink node, so the sensor node goes into sleep mode until predicted time to transmit data, and then the sensor node enters the active mode and sends its data to the mobile sink node. The DOSM algorithm proposed in [14] divides the monitoring area into grids. Next, the algorithm selects the cluster head based on the energy in each grid in each round, then solves the centroid of all cluster head positions which is the position of mobile sink node. After that, it is able to establish the route and transmit the data. The VUGR algorithm proposed in [15] divides the low-energy-level grid cells into smaller units. In the case of not participating in the construction of the virtual high-level structure, the algorithm transmits the data by searching grid unit in high-energy-level, and the mobile sink node moves clockwise along the high-energy-level grid cells which is near to edge of the network. Although both of the DOSM algorithm and the VUGR algorithm consider the balance of energy, the routing update is frequent and the route construction is complicated. The algorithm proposed in [16] selects the sensor nodes with higher weights as the rendezvous points and establish the set of points, then find the path which can access all rendezvous points and does not exceed the maximum delay of data transmission. Based on the position information of the sensor nodes, the algorithm proposed in [17] determines the position of each cluster head which is the stop position of the mobile sink node by using range-constrained clustering algorithm, and uses the TSP algorithm to find the path that traverses all the staying positions. However, algorithms proposed in [16] and [17] only consider the optimization of single factor (delay, path length) in the path planning of mobile sink nodes, do not consider the combination optimization problem that include energy consumption of sensor nodes and the network lifetime. In [18], the proposed algorithm builds the network into multiple circles with different communication ranges according to each sensor node at first, then mobile sink node moves to the communication range of each sensor node to collect data, next, builds a model of energy consumption and movement path, and uses mixed immunity particle swarm algorithm to plan the path. But, although the combination optimization of energy consumption and moving path is considered, the mobile sink node stays frequently, and the total movement path and network delay are too large.

The predictive mobile model is superior to the other three mobile models in the control of nodes and the energy consumption of the network. Therefore, based on the predictive moving model, this paper presents a mobile multi-sink node path planning algorithm with energy balance (hexHPSO). First, the algorithm divides the monitoring area into grid and then abstracts the network into a selective TSP problem which can solved by a hybrid positive and negative particle swarm optimization algorithm (HPNPSOA). Finally, it obtains an optimal path and the path is evenly distributed to multiple mobile sink nodes. The proposed algorithm balances the energy consumption of the network and effectively extends the network lifetime.

II. SYSTEM MODEL PRESENTATION

A. System Assumption

In wireless sensor networks, nodes are divided into mobile sink nodes and sensor nodes. We assume that:

1. \( n \) sensor nodes are randomly deployed within the two-dimensional monitoring area and the position is stationary. The \( k \) mobile sink nodes can move within the monitoring area and stay collecting data.

2. All sensor nodes have the same performance (such as communication radius, initial energy, energy consumption model, etc.);

3. All nodes can know their own position coordinates through GPS global positioning system or other positioning methods;

4. All sensor nodes have limited energy and cannot be supplemented, but all mobile sink nodes have infinite energy.

B. Energy Consumption Model

In this paper, the wireless communication energy consumption model similar to the reference [19] is adopted. Since only one-hop routing is considered to send data from the sensor node to the mobile sink node, the sensor node only has the energy consumption of transmitting data:

\[
E_{\text{tx}}(l, d) = \begin{cases} 
|E_{\text{tx}} + l\epsilon_{fs}d^2, d < d_0 \\
|E_{\text{tx}} + l\epsilon_{mp}d^4, d \geq d_0 
\end{cases} 
\]  
(1)

Where: \( E_{\text{tx}} \) represents the energy consumption of the transmission circuit; \( \epsilon_{fs} \) and \( \epsilon_{mp} \) represent the amplifier power consumption of the free-space model and multi-path attenuation model respectively; \( l \) represents the length of the data packet; \( d_0 \) represents the distance threshold.

\[
d_0 = \sqrt{\epsilon_{fs}/\epsilon_{mp}} 
\]  
(2)

The communication between the mobile sink node and the sensor node is limited to one virtual grid and the grid side length is \( R \) (\( R \) is less than \( d_0 \)), that is to say, to adopt one-hop mode. In consequence, the free-space model is adopted in this paper.

III. NETWORK MODEL PRESENTATION
A. Network Model

As illustrated in Figure 1, \( n \) sensor nodes are randomly deployed within the monitoring area with the side length of \( L \), and \( \mathcal{V}_n \) represents the sensor node set. The monitoring area is divided into \( N \) virtual regular hexagonal grids, where the grid side length is \( R \) (\( R \) is the sensor node communication radius), \( \mathcal{V}_g \) represents the set of nodes in the grid, \((x, y)\) represents the coordinates of the grid, and \( g(g = 1, 2L N) \) represents the grid number. Compared with the triangles, squares, etc., proposed in the reference [10]-[12], the regular hexagon has a larger cover area (the same coverage radius), which makes the mobile sink node collect more sensor node data in the parking position and obviously reduce the number of parking positions demanded in a whole network data collection. If there is a circle which radius is \( r \), we can work out the area of circle is \( S_1 = \pi r^2 \), and the area of square which involved in circle is \( S_2 = (\sqrt{2}r)^2 \), but the area of regular hexagon which involved in same circle is \( S_3 = \frac{2\sqrt{3}}{2}r^2 \), so that we can get the result that \( S_1 - S_2 > S_1 - S_3 \), it means the regular hexagon’s use ratio is higher than square as well as triangle. Meanwhile regular hexagon is the frequently-used shape in geometry model with excellent mathematics property.

Therefore, this article uses the regular hexagon to divide the grid. There are two candidate parking positions of the mobile sink node in each grid: the center point position of the entire grid and the centroid point position of the sensor node distribution. \( \mathcal{V}_{site} \) represents the set of candidate parking positions, and the mobile sink nodes selects only one of the parking positions in each grid to collect data of the sensor nodes. Let \( \mathcal{V}_s \) represents the set of candidate parking positions selected by the mobile sink node, \( |\mathcal{V}_s| = N \).

The key to dividing the virtual regular hexagonal grid is to determine which grid each sensor node belongs to, as illustrated in Figure 2, the sensor nodes are divided into two categories for judgment.

1) VIRTUAL REGULAR HEXAGONAL GRID DIVISION

The grids which in boundary area is can be irregular or any shape in the proposed algorithm. It can explain the availability of algorithm adequately when the monitoring area is an irregular shape.
In Fig. 3 and Fig. 4, the red rectangle means the special area and blue point means the center point of the grid which is near the special area. When the sensor node is in the special odd column, the coordinates of the adjacent grid are \((2j, k + 1), (2j - 1, k)\) and \((2j, k)\). Then calculate the distance between the sensor node and the center point of its adjacent grids. Finally, the adjacent grid with the smallest distance serves as the grid of the sensor node. When the node is in a special even column, the coordinates of the adjacent grid are \((2j, k + 1), (2j, k)\) and \((2j + 1, k)\). Then calculate the distance between the sensor node and the center point of its adjacent grids. Finally, the adjacent grid with the smallest distance serves as the grid of the sensor node. The specific algorithm process is shown in Table I below:

**TABLE I**

<table>
<thead>
<tr>
<th>Algorithm 1: Sensor node attribution algorithm in the special area</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Sensor node set (S) and number (n),</td>
</tr>
<tr>
<td>Special area column number (col),</td>
</tr>
<tr>
<td>Special area row number (row),</td>
</tr>
<tr>
<td>Grid length (R)</td>
</tr>
<tr>
<td><strong>Output:</strong> Grid number of the sensor node and Grid coordinates of the sensor node</td>
</tr>
<tr>
<td>1: for (i = 1) to (n) do</td>
</tr>
<tr>
<td>2: for (j = 0) to (\text{col}/2) do</td>
</tr>
<tr>
<td>3: for (k = 1) to (n) do</td>
</tr>
<tr>
<td>4: if (\frac{S_{(i, xd)}}{R/2} = 3 + (j - 1) \times 6 &amp; \frac{S_{(i, xd)}}{\sqrt{3}R} \equiv k) then the node is in a special odd column</td>
</tr>
<tr>
<td>5: (d(2j, k + 1)d(2j - 1, k)). Calculate the distance between the node and the center point of its adjacent grids</td>
</tr>
<tr>
<td>6: (\text{misDis} \leftarrow \min{d(2j, k + 1), d(2j - 1, k), d(2j, k)}). Find the shortest distance</td>
</tr>
<tr>
<td>7: (gx, gy, gno), the coordinates and number of the nearest grid</td>
</tr>
<tr>
<td>8: if (\frac{S_{(i, xd)}}{R/2} = 3 + (j - 1) \times 6 &amp; \frac{S_{(i, xd)}}{\sqrt{3}R} \equiv k) then the node is in a special even column</td>
</tr>
<tr>
<td>9: (d(2j, k + 1)d(2j, k)). Calculate the distance between the node and the center point of its adjacent grids</td>
</tr>
<tr>
<td>10: (d(2j, k + 1), d(2j, k), d(2j + 1, k)) . Calculate the distance between the node and the center point of its adjacent grids</td>
</tr>
</tbody>
</table>

The sensor node attribution algorithm in the special area is used to assist the main algorithm to divide the monitoring area into grids. So we can learn from the paper that although the sensor node attribution algorithm in the special area has high computational complexity, it can make a great deal of contribution in judging grid each sensor node included.

2) **CANDIDATE PARKING POSITION**

There are two candidate parking positions for each grid: one is the center point position of the grid; the other is the centroid point position of the sensor nodes distributed within the grid. The reason why select these two positions as the candidate parking positions of the mobile sink node is that they are the least costly communication positions. On the one hand, the center point position ensures that all sensor nodes are within the communication radius of the mobile sink node, in other words, one-hop data transmission is assured. On the other hand, since the distribution of nodes in each grid is different, once the nodes in the grid are concentrated in a small area in the grid, the communication cost of the centroid point position is smaller than the center point position.

The calculation formula of the center coordinate \((x_c, y_c)\) and the centroid point coordinate \((x_z, y_z)\) is as follows:

\[
x_c = \frac{R + (x - 1) \times 3R}{2} = \frac{\sqrt{3}R \times (x \mod 2) + 2 \sqrt{3}R \times y - 1}{2} \tag{3}
\]

\[
y_c = \frac{\sqrt{3}R \times (x \mod 2) + 2 \sqrt{3}R \times y - 1}{2} \tag{4}
\]

\[
x_z = \frac{\sqrt{3}R \times (x \mod 2) + 2 \sqrt{3}R \times y - 1}{2} \tag{5}
\]

\[
y_z = \frac{\sqrt{3}R \times (x \mod 2) + 2 \sqrt{3}R \times y - 1}{2} \tag{6}
\]

\((x, y)\) represents the coordinates of the grid, \(h\) represents the number of nodes in the grid, \(S_{(i, xd)}\) represents the abscissa of the \(i\) node, and \(S_{(i, yd)}\) represents the ordinate of the \(i\) node.

**B Optimization Model**

According to the authoritative literatures, there are some different definition of network lifetime. In some literatures, author define network lifetime as the minimum value of the lifetime of nodes, and in other case, the network lifetime is defined as the maximum value of the lifetime of nodes, in addition, it also defined as the average of the lifetime of nodes. In this paper, we choose the one of definition with the algorithm analysis that the network is defined as the minimum value of the lifetime of nodes.

The whole network model can be abstracted into the parking position selection problem of the mobile sink node and the TSP (Traveling Salesman Problem) problem of caused by the mobile sink node to traverse the grid. The optimization
goal of this paper is to project a path with the shortest length, small delay, balanced energy consumption and maximizing the network lifetime. In order to reduce the delay as much as possible, a plurality of mobile sink nodes is used to share the excessive delay caused by only one mobile sink node. The delay in this paper mainly includes data transmission delay and mobile delay.

The data transmission delay refers to the time interval at which the sensor node sends data to the mobile sink node, which is related to the transmission rate $v_t$ of the sensor node and the distance $d_s$ to the sink node.

$$T_{s1} = d_s/v_c$$

The mobile delay refers to the time taken by the mobile sink node to move, which is related to the path length $d_s$ and the moving speed $v_m$ of the mobile sink.

$$T_{s2} = d_s/v_m$$

Let $k$ be the number of mobile sink nodes, and divide the $N$ grids into $K$ virtual groups [20]:

$$N = cLd$$

$c$ represents the quotient, $d$ represents the remainder, the first $d$ virtual groups are assigned $c + 1$ grids, and the rest of the groups are allocated $c$ grids.

Grid energy consumption $E_c$ refers to the sum of energy consumption of all nodes in the grid, that is,

$$E_c = \sum_{i=1}^{h} E_i$$

$h$ represents the number of nodes in the grid, $E_i$ represents the communication energy consumption of the node, and the energy consumption of different parking positions is different. According to the formula (1), we can see that $E_i = lE_{tx} + le_f d^2$.

Group energy consumption $E_p$ represents the sum of grid energy consumption within each group. In order to achieve the aim of the balanced energy consumption, the energy consumption of each group needs to be as similar as possible. Therefore, it is necessary to calculate the average group energy consumption $E_p$ and the energy consumption variance between groups $V(E_p)$:

$$E_p = \sum_{j=1}^{k} E_c(j)$$

$$E_p = \frac{\sum_{i=1}^{k} E_{pi}}{k}$$

$$V(E_p) = \frac{\sum_{i=1}^{k} (E_{pi} - E_p)^2}{k}$$

$t$ represents the number of grids in each group, and $k$ represents the number of mobile sink nodes.

Defining the node lifetime is the time it takes for its energy to run out, so the lifetime of the node $i$ [21] is:

$$T_i = \left\{ \frac{c_i}{E_{t1} E_{t2}}, i = 1, 2L n \right\}$$

$c_i$ represents the residual energy of the $i$ node, $E_{t1}$ represents the central point communication energy consumption of the $i$ node, and $E_{t2}$ represents the centroid point communication energy consumption of the $i$ node.

The network lifetime is the time it takes for the first node in the network to die, that is,

$$T = \min T_i (i = 1, 2L n)$$

According to the above analysis, the purpose of this paper is to minimize the path length and the energy variance between groups and maximize the network lifetime. Therefore, the following optimization model can be established:

$$\min (V(E_p)) \text{ s.t. } \min \left( \frac{d_{tSP}}{T} \right)$$

$s, t$ constraints: (10), (11), (12), (13), (14), (15)

$d_{tSP}$ in formula (16) represents the entire path length, which is the sum of paths of all mobile sink nodes.

IV. THE DETAIL OF HPNPSOA

For the general TSP problem, it can be solved by the hybrid particle swarm optimization algorithm (HPSOA) [22]. The HPSOA algorithm combined the traditional particle swarm optimization algorithm (PSOA) [23] and the genetic algorithm (GA) [24] adopts the crossover and mutation proposed in the GA algorithm to replace tracking optimum proposed in the PSOA algorithm to update particle swarms. Compared with GA algorithms, HPSOA algorithm has faster convergence rate and optimization results. However, the optimization model of this paper is not the general TSP problem, but the selective TSP problem cannot be solved by the HPSOA algorithm. This is because elements in particles proposed in the HPSOA algorithm merely represent a certain value. As a consequence, this paper proposes a hybrid positive and negative particle swarm optimization algorithm (HPNPSOA) to solve this optimization model.

A The Process of HPNPSOA

Analyze the optimization model to derive the objective function of the HPNPSOA algorithm:

$$F = \omega \times V(E_p) + \mu \times \frac{d_{tSP}}{T}$$

$\omega$ represents the weight of the energy variance between groups, $\mu$ represents the weight of the ratio of path to network lifetime. The higher the value of $\omega$, the more objective function results reflect the energy balance between different groups. The higher the value of $\mu$, the more objective function results reflect the energy balance of the entire network.
As illustrated in Figure 5, the detailed implementation steps of the HPNPSO algorithm are as follows:

Step 1 Particle swarm initialization. Initialize the parameters of the HPNPSO algorithm: initial value of iterations \( m \) \((m = 1)\), maximum iterations \( M \), number of positive and negative particle pairs \( D \), and so on. Initialize the positive and negative particles swarm so that the number of elements in each particle is \( N \). The positive particles represent the sequence of the grid number 1 to \( N \); the negative particles represent the parking position valued 0 or 1, 0 represents the center point position, and 1 represents the centroid point position.

Step 2 Fitness values, also known as objective function value. Obtain the fitness values of each pair of particles according to the calculation formula (17) of the objective function. The smaller the fitness value, the more effective the optimization. Calculate the local optimum \( pbest \) and global optimum \( gbest \) of each pair of particles. If the current fitness value of each pair of particles is less than the local optimum or the global optimum, the local optimum and the global optimum are updated with the current fitness value.

Step 3 Crossover operation. First, each pair of particles crosses the positive and negative particles corresponding to the local optimum to update themselves. Then each pair of particles crosses the positive and negative particles corresponding to the global optimum to update themselves again. The bits used to cross \((c_1, c_2)\), \(1 \leq c_1 \leq c_2 \leq N\) and \((c_3, c_4)\), \(1 \leq c_3 \leq c_4 \leq N\) are generated randomly. The bits used to insert \( p_{flag} \) \(1 \leq p_{flag} \leq N - (c_2 - c_1) - 1\) and \( g_{flag} \), \(1 \leq g_{flag} \leq N - (c_4 - c_3) - 1\) are also generated randomly. The \( p_{flag} \) \( p_{flag} + (c_2 - c_1) \) elements of each pair of particles are replaced by \( c_2 \); \(c_4\) elements of each pair of local optimum particles, and the \( g_{flag} \) \( g_{flag} + (c_4 - c_3) \) elements of each pair of global optimum particles.

Step 4 Mutation operation. The bits used for mutation \((v_1, v_2)\), \(1 \leq v_1 \leq v_2 \leq N\) are generated randomly. The elements from the \( v_1 \) to \( v_2 \) positions of each pair of particles are reversed to insert the original \( v_1 \) to \( v_2 \) positions, and the rest remain unchanged.

Step 5 Iteration. Iterating until the number of iterations \( m \) is equal to the maximum number of iterations \( M \), resulting in a pair of optimal positive and negative particles, that is, the optimal grid traversal order and optimal parking position.

**B The Analysis of HPNPSO**

The HPNPSO algorithm improved the HPSOA algorithm combines the idea that the positive and negative particles attract each other in physics to solve the selective TSP problem.

The HPNPSO algorithm alters each particle in the HPSOA algorithm to a pair of positive and negative particles during the particle swarm initialization. The positive particles represent the order in which the mobile sink nodes traverse the grid, and the negative particles represent the choice of the parking positions within each grid. In other words, the positive and negative particle represent two factors of the route planning process, the two elements in the same position in each pair of positive and negative particles together can work out a certain position. Only two elements exist together can we confirm the position, so there were named by positive and negative particle compare to attraction in physics. The two elements in the same position in each pair of positive and negative particles together represent a certain position, and due to the attraction, they also change together during the crossover and mutation operations. Although this method of creating particle swarms can solve the selective TSP problem, compared with the HPSOA algorithm, the scale of particle swarms becomes huge. In order to ensure the algorithm has faster convergence rate and optimization results, the crossover and mutation operations have also been improved by the HPNPSO Algorithm.

The HPNPSO Algorithm abandons the method that the elements between bits used to cross of the optimum insert the end of the particle to adopt the method that the elements between bits used to cross of the optimum replace the elements between bits used to insert generated randomly during the crossover operation. When the mutation operation is performed, the method of exchanging the elements of two bits used for mutation generated randomly is no longer used, but the method that reverses the elements of between bits used for mutation generated randomly to replace the elements of between bits used for mutation generated randomly. With improved crossover and mutation operations, the stability of the algorithm can be improved, so that it will not converge too slowly, increase the running time of the algorithm, and will not fall into the local optimal results due to too fast convergence.

**V. SIMULATIONS**

**A Simulation Scene And Parameter Settings**
Simulation scene setting: 100 sensor nodes are randomly deployed within the square monitoring area of 100 100, the sensor nodes are stationary, and three mobile sink nodes are deployed within the monitoring area to move at a constant speed of 2. The simulation parameter settings are referenced in [25], as shown in Table II below:

### TABLE II

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor node initial energy $E_{0}/j$</td>
<td>0.1</td>
</tr>
<tr>
<td>Sensor node transmission rate $V_{e}/m \cdot s^{-1}$</td>
<td>1</td>
</tr>
<tr>
<td>Date packet size $l$/bit</td>
<td>2000</td>
</tr>
<tr>
<td>Energy consumption of the transmission circuit $E_{tx}/nj \cdot bit^{-1}$</td>
<td>50</td>
</tr>
<tr>
<td>Amplifier power consumption of the free-space model $\varepsilon_{f}/pj \cdot m^{2} \cdot bit^{-1}$</td>
<td>10</td>
</tr>
<tr>
<td>Communication radius $R$/m</td>
<td>20</td>
</tr>
<tr>
<td>Objective function weight $\omega$</td>
<td>120000</td>
</tr>
<tr>
<td>Objective function weight $\mu$</td>
<td>0.16</td>
</tr>
</tbody>
</table>

Meanwhile, considering the energy balance between different groups and the energy balance of the entire network, therefore the values of $\omega \times V(\overline{E_{F}})$ and $\mu \times d_{TSP}/T$ in the objective function (17) are almost the same, that is,

$$\frac{\omega}{\mu} = \frac{d_{TSP}}{T} \cdot \frac{V(\overline{E_{F}})}{N(\overline{E_{F}})}$$

(18)

The values of $\omega$ and $\mu$ were obtained experimentally as shown in Table III below.

### TABLE III

<table>
<thead>
<tr>
<th>Objective function weight values</th>
<th>Parameter</th>
<th>$\omega$</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V(\overline{E_{F}})$ mean</td>
<td>$d_{TSP}/T$ mean</td>
<td>$\frac{\omega}{\mu}$</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>6.7925e-10</td>
<td>8.1899e-03</td>
<td>1.2057e+06</td>
</tr>
<tr>
<td>400</td>
<td>4.1001e-10</td>
<td>7.9043e-04</td>
<td>1.9278e+06</td>
</tr>
<tr>
<td>600</td>
<td>1.1014e-10</td>
<td>5.6722e-04</td>
<td>5.1500e+06</td>
</tr>
<tr>
<td>800</td>
<td>2.5740e-11</td>
<td>4.5889e-04</td>
<td>1.7828e+07</td>
</tr>
<tr>
<td>1000</td>
<td>3.2335e-11</td>
<td>4.0489e-04</td>
<td>1.2522e+07</td>
</tr>
</tbody>
</table>

From Table III, the average value of $\omega/\mu$ is $7.7267e+06$, so $\omega = 1200000$ and $\mu = 0.16$ can be obtained, so that the equation $\omega/\mu = 7.7267e+06$ is established.

### B The hexHPSO Algorithm Performance Analysis

The performance of the swarm intelligence optimization algorithm such as GA algorithm and PSOA algorithm depends mainly on the convergence, stability and optimization target value of the algorithm. As shown in Figure 6, the hexHPSO algorithm proposed in this paper is compared with the classical hybrid particle swarm optimization (HPSO) algorithm. From the perspective of iterations, the classical HPSO algorithm iterates to 52 times to get the optimal target value, while the hexHPSO algorithm only needs iterate to 23 times. The convergence of hexHPSO algorithm is obviously better than the classical HPSO algorithm. From the perspective of optimization target value, the optimization result of classical HPSO algorithm is 0.039, and the optimization result of hexHPSO algorithm is 0.033. The optimization result shows that the hexHPSO algorithm has stronger ability to get the optimum solution compared with the classical HPSO algorithm. After obtaining the optimal solution, the fitness value of the hexHPSO algorithm no longer changes with the iterations, indicating that the algorithm has good stability. Comprehensive analysis, hexHPSO algorithm has good convergence, stability and strong ability to get the optimum solution.

The red circle means centroid point of each area and the blue asterisk means center point of each area. The number of mobile sink nodes is related to the size of the monitoring area and the size of the grid. If the path of the mobile sink nodes is too long, the mobile delay will increase, so the number of mobile sink nodes cannot be too small. Certainly, the mobile sink nodes is not too many, which will result in increasing costs and wasting resources. As illustrated in Figure 7, the monitoring area is divided into $N = 14$ grids (the maximum number of mobile sink nodes is 14), and the correspondence between the number of mobile sink nodes and the minimum number of grids traversed by the mobile sink node is as shown in Table IV below:

### TABLE IV

<table>
<thead>
<tr>
<th>Minimum number of grids</th>
<th>Number of mobile sink nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>14</td>
</tr>
<tr>
<td>1200</td>
<td>13</td>
</tr>
<tr>
<td>1500</td>
<td>12</td>
</tr>
<tr>
<td>1800</td>
<td>11</td>
</tr>
<tr>
<td>2100</td>
<td>10</td>
</tr>
<tr>
<td>2400</td>
<td>9</td>
</tr>
<tr>
<td>2700</td>
<td>8</td>
</tr>
<tr>
<td>3000</td>
<td>7</td>
</tr>
<tr>
<td>3300</td>
<td>6</td>
</tr>
<tr>
<td>3600</td>
<td>5</td>
</tr>
<tr>
<td>3900</td>
<td>4</td>
</tr>
<tr>
<td>4200</td>
<td>3</td>
</tr>
<tr>
<td>4500</td>
<td>2</td>
</tr>
<tr>
<td>4800</td>
<td>1</td>
</tr>
<tr>
<td>Particle swarm size</td>
<td>1</td>
</tr>
<tr>
<td>---------------------</td>
<td>---</td>
</tr>
<tr>
<td>Minimum number of grids</td>
<td>14</td>
</tr>
</tbody>
</table>

After the number of mobile sink nodes exceeds 8, the minimum number of grids traversed by the mobile sink node is 1, hence it is obvious that the number of mobile sink nodes is too large. It can be seen from Table IV that the minimum number of grids traversed by the mobile sink node is the set \{1, 2, 3, 4, 7, 14\}, so the optimal minimum number of grids is 4 (4 to 7 is closer to the mean value). As a result, the number of mobile sink nodes is three.

**FIGURE 8. Mobile sink node movement path**

Figure 8 shows the movement path of three mobile sink nodes planned by the hexHPSO algorithm. It is shown in formula 9 in this paper that one movement path is divided into three parts in the way that the first virtual group and the second virtual group are assigned five grids, and the third group are allocated four grids. It can be seen that for the sink node moving in the edge group, the location of the centroid point can balance the energy consumption of the nodes, prolong the network lifetime, and minimize the movement path of the sink node to reduce the delay. When the distance between the centroid point and the center point is small, the sink node moving in the internal group can select the position of the center point to further reduce the movement path. Therefore, the hexHPSO algorithm can obtain the path with the shortest length, small delay, balanced energy consumption and maximum network lifetime. The sensor node turn into operative mode when the mobile sink nodes move into the grid which is the sensor node located in, then it transmit the data each other with the mobile sink nodes. Another case is that when the mobile sink nodes move out from the grid which is the sensor node located in, the sensor node turn into sleep mode in order to save the energy.

**2) NETWORK LIFETIME**

**FIGURE 9. Network lifetime**

Figure 9 shows the network lifetime comparison graph of the hexHPSO algorithm, the DOSM algorithm, the GLRM algorithm, and the RWM algorithm. Defining the network lifetime is the time when the first node in the network dies. From Figure 9, it can be seen that the network lifetime of the hexHPSO algorithm is prolonged to 155%, 86%, and 68%, respectively, compared with the DOSM algorithm, the GLRM algorithm, and the RWM algorithm. The hexHPSO algorithm effectively extend the network lifetime. Compared with GLRM algorithm and RWM algorithm, the hexHPSO algorithm curve is steeper, and the interval between the number of rounds where the first node died and the number of rounds where all nodes died is very short, indicating that the hexHPSO algorithm has more balanced energy consumption in the whole network.

**3) COMPARISON OF DEAD NODES**

**FIGURE 10. The comparison of the dead nodes**

Figure 10 shows a comparison histogram of the number of rounds died of first node, half of nodes and all nodes in the hexHPSO algorithm, the DOSM algorithm, the GLRM algorithm, and the RWM algorithm. As can be seen from Figure 10, the hexHPSO algorithm has a longer lifetime than other algorithms in all three time periods. From the perspective of the number of death rounds for the first node, the hexHPSO algorithm is 155%, 86%, and 68% higher than the DOSM algorithm, GLRM algorithm, and RWM algorithm, respectively. From the perspective of the number of death rounds for the half of nodes, the hexHPSO algorithm is 176%, 21%, and 20% higher than the DOSM algorithm, GLRM algorithm, and RWM algorithm, respectively.
algorithm, and RWM algorithm, respectively. From the perspective of the number of death rounds for the all nodes, the hexHPSO algorithm is 113%, 10%, and 6% higher than the DOSM algorithm, GLRM algorithm, and RWM algorithm, respectively. In conclusion, the hexHPSO algorithm improves the lifetime of nodes throughout the network cycle and extends the network lifetime.

4) NETWORK RESIDUAL ENERGY

![Network Residual Energy Comparison](image)

**FIGURE 11.** The comparison of network residual energy

Figure 11 shows the network residual energy comparison graph of the hexHPSO algorithm, the DOSM algorithm, the GLRM algorithm, and the RWM algorithm. It can be seen from the figure that the hexHPSO algorithm has the most residual energy in each rounds, which indicates that this algorithm can greatly reduce the energy consumption of the network. The network residual energy comparison graph is consistent with the network lifetime comparison graph in Figure 9.

5) ENERGY CONSUMPTION BALANCE AND HOTSPOT PHENOMENON

![Energy Consumption Comparison](image)

**FIGURE 12.** The comparison of energy consumption

Figure 12 shows a three-dimensional comparison of the node average energy consumption of the hexHPSO algorithm, the DOSM algorithm, the GLRM algorithm, and the RWM algorithm. There are four diagrams in figure 12 which are used to show the comparison of energy consumption. In the diagram, the monitoring area which is 100*100 is divided into 100 equal units, and each unit is 10*10. In this way, we can calculate the energy consumption in each unit so that analyse the comparison of four algorithm more clearly. It can be clearly seen from the figure that the hexHPSO algorithm has the lowest energy consumption and the best energy consumption balance. The DOSM algorithm has a large energy consumption, but its energy consumption balance is relatively good. The GLRM algorithm has obvious hotspot phenomenon and the RWM algorithm has a large and unbalanced energy consumption. The hexHPSO algorithm comprehensively considers the distribution of nodes, energy balance and network lifetime to plan the path of mobile sink nodes, balances the energy consumption of the entire network effectively, and alleviates the hotspot phenomenon.

6) NETWORK DELAY

![Network Delay Comparison](image)

**FIGURE 13.** The comparison of the number of nodes - time delay

Figure 13 shows the number of nodes - time delay comparison graph of the hexHPSO algorithm, the DOSM algorithm, the GLRM algorithm, and the RWM algorithm. The network size is 200 200. It can be seen from the graph that the larger the number of nodes, the larger the delay in data transmission and then the larger the total delay. The hexHPSO algorithm has less delay than the DOSM algorithm and GLRM under different node numbers. However, after the number of nodes exceeds 400, the delay of the hexHPSO algorithm begins to be larger than that of the RWM algorithm. The reason is that with the increasement of sensor nodes the nodes in the grid are distributed densely and the difference between the centroid point and the center point is reduced. Meanwhile, it lead the difference of data transmission delay become smaller which costed by hexHPSO algorithm and RWM algorithm. In other words, because the decrease of the data transmission delay in total delay proportion, it makes the advantage of the hexHPSO algorithm in reducing delay decrease slightly to some extent. However, delay is not the main optimization goal of this paper, but low and balanced energy consumption. Introducing mobility to reduce and balance network energy consumption is originally achieved at the cost of increasing network latency, so this paper only reduces the delay as much as possible.
obtaining the optimal order to traverse grids and getting the hybrid positive and negative particle swarm optimization network model is abstracted as a special traveling salesman lifetime and movement path. It is proposed that the entire factors include balance of group energy consumption, network collect the data of sensor nodes in each grids. At the same time, mobile sink node stays at the one of the stop positions to position and centroid point position) in each grid. Then, the grids, and stores two candidate stop positions (center point divides the monitoring area into plurality of regular hexagonal (hexHPSO) is proposed. At first, the proposed algorithm multi-sink nodes path planning algorithm with energy balance distribution of nodes and fixed location of sink node, a mobile of sensor nodes and "empty hole" caused by uneven distribution of nodes, extends the network lifetime and minimizes network latency.

VI. CONCLUSIONS

For solving the problem of unbalanced energy consumption of sensor nodes and "empty hole" caused by uneven distribution of nodes and fixed location of sink node, a mobile multi-sink nodes path planning algorithm with energy balance (hexHPSO) is proposed. At first, the proposed algorithm divides the monitoring area into plurality of regular hexagonal grids, and stores two candidate stop positions (center point position and centroid point position) in each grid. Then, the mobile sink node stays at the one of the stop positions to collect the data of sensor nodes in each grids. At the same time, it divides into groups according to the number of mobile sink node, and builds an optimization model which combine the factors include balance of group energy consumption, network lifetime and movement path. It is proposed that the entire network model is abstracted as a special traveling salesman problem with selection. Aiming at resolving such problem, a hybrid positive and negative particle swarm optimization algorithm (HPNPSOA) is proposed. HPNPSOA is used for obtaining the optimal order to traverse grids and getting the path of stop positions selected by using improved crossover and mutation operations. Finally, we assign optimal path to different mobile sink nodes in groups, so that each mobile sink node moves in its own group. In order to ensure the synchronization of all mobile sink nodes, the time which is spend longest in data collection by mobile sink nodes in each round is defined as the data collection time of the total network every round. The proposed algorithm solves the "empty hole" problem effectively and balance the energy consumption of nodes, extends the network lifetime and minimizes network latency.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program (NO. 2017YFB1401800).

The authors express their sincere appreciation to the editors and the anonymous reviewers for their helpful comments.

REFERENCES


Jianping Zhao received the master’s and Ph.D. degrees from the Changchun University of Science and Technology, China. He is currently a Professor with the School of Computer Science and Technology, Changchun University of Science and Technology.

Xin Feng received a M.Sc. degree from the School of Computer Science and Information Technology, Northeast Normal University, P.R. China in 2006. He is an associate professor in Changchun University of Science and Technology school of computer and technology. His research area includes internet of Things, network security and body area networks.

Jing Zhang received a PhD. degree and M.Sc. degree from the College of Computer Science and Technology, Jilin University, P.R. China in 2015 and 2012. She is a lecturer in Changchun University of Science and Technology school of computer and technology. Her research interests include wireless sensor networks and complex networks.

Jiaqi Luo is currently pursuing the master’s degree with the College of Computer Science and Technology, Changchun University of Science and Technology. Her research interests are wireless sensor network and routing algorithm.

Supeng Sun received the master’s degree from the Changchun University of Science and Technology, China. He is currently an Engineer with the School of Computer Science and Technology, Changchun University of Science and Technology.