IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 18, 2019, accepted December 2, 2019, date of publication December 5, 2019, date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2957834

Energy-Efficient Trajectory Planning Algorithm Based on Multi-Objective PSO for the Mobile Sink in Wireless Sensor Networks

XIAOLIN HE, XIUWEN FU, AND YONGSHENG YANG
Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China
Corresponding author: Xiuwen Fu (xwfu@shmtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61902238.

ABSTRACT In wireless sensor networks (WSNs), collecting data with mobile sinks is an effective way to solve the ''energy hole problem''. However, most of existing algorithms of mobile sinks ignore the load balance of rendezvous nodes, which will significantly shorten the network lifetime. Moreover, most mobile sinks are usually required to visit locations of sensor nodes without taking advantage of their communication ranges. Therefore, this paper proposes an energy-efficient trajectory planning algorithm (EETP) based on multi-objective particle swarm optimization (MOPSO) to shorten the trajectory length of the mobile sink and balance the load of rendezvous nodes. EETP aims to reduce the delay in data delivery and prolong the network lifetime. To shorten the trajectory length of the mobile sink, we design a mechanism to select potential visiting points within communication overlapping ranges of sensor nodes, rather than locations of sensor nodes. Additionally, according to trajectory characteristics of the mobile sink, we design an effective trajectory encoding method that can generate a trajectory containing an unfixed number of visiting points. The simulation results show that the proposed EETP is superior to existing WRP, CB and the MOPSO-based algorithm, in terms of delay in data delivery, network lifetime and energy consumption.

INDEX TERMS Mobile sink, MOPSO, load balance of rendezvous nodes, trajectory planning.

I. INTRODUCTION

The Internet of Things (IoT) has been widely used in environmental monitoring [1], industrial control [2], health care [3] and other fields. In the application of IoT, battery-powered sensors are used to sense the information of objects and the sensed data is transmitted to users for further processing. A large number of sensor nodes form a wireless sensor network (WSN) [4]. In traditional WSNs, sensor nodes near the static sink need to relay a lot of data so that they will consume more energy [5]. Due to the limited energy of battery-powered sensor nodes, excessive energy consumption of sensor nodes near the static sink will result in the network partition or even isolate the static sink, shortening the network lifetime (i.e., energy hole problem) [6]–[8]. Using the mobile sink to collect data is an effective way to solve this problem [9]–[11]. The mobile sink is to install an ordinary sink on the object that can move in the monitoring area. In this way, the data of sensor nodes can be transmitted to multiple sensor nodes to store and wait for the mobile sink to collect. Therefore, hop counts and energy consumption in multi-hop transmission can be reduced using the mobile sink. In early studies, the mobile sink collects data from sensor nodes individually [12], [13], but it will lead to a high delay in data delivery due to the long trajectory of the mobile sink, especially in large-size sensor networks. To tackle this problem, most current studies only allow the mobile sink to collect data from some of nodes called rendezvous nodes (RNs) and non-RN nodes transmit data via RNs [14]–[16]. The selection of RNs is a challenging issue. If the trajectory of the mobile sink is enough long, there are more RNs reducing energy consumption caused by multi-hop transmission and energy consumption can be further balanced. However, the too long trajectory of the mobile sink will cause a high delay in data delivery. The focus of most researches is to select a sufficient number of RNs under the trajectory length constraint of the mobile sink [17]–[19]. Therefore, in these studies, the trajectories are usually close to the maximum allowed length

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

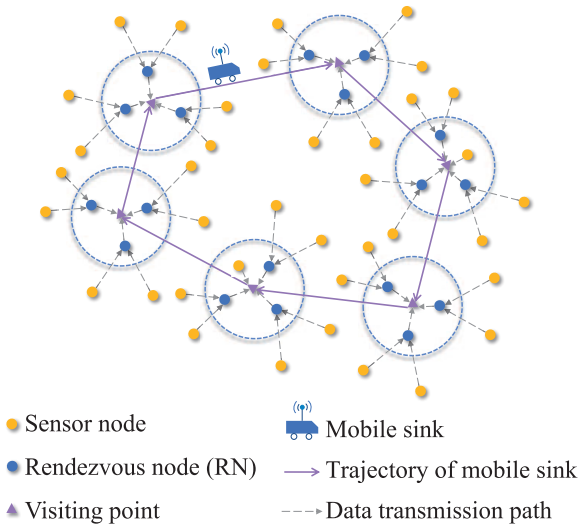

FIGURE 1. An example of the mobile sink collecting data.

of the trajectory. Moreover, mobile sinks in these studies are only allowed to collect data at the location of sensor nodes. If mobile sinks can collect data in the communication range of sensor nodes, trajectories of mobile sinks can be further shortened and the delay in data delivery will be lower. In addition, the load-balance level of RNs (i.e., the number of child nodes of RNs) is not given too much attention in these studies. If the load of one RN is much larger than that of other RNs, this RN will quickly deplete its energy.

Therefore, we need a trade-off between the trajectory length of the mobile sink, the number of RNs and the load balance of RNs. In other words, the data collection with the mobile sink is a multi-objective and complex problem. For general multi-objective optimization algorithms, it may require a very high computational time. However, heuristic algorithms such as multi-objective particle swarm optimization (MOPSO) can solve similar problems and supply the entire Pareto set of optimal solutions [20], [21]. In this paper, we propose an energy-efficient trajectory planning algorithm (EETP) based on MOPSO to solve this multi-objective problem, as is shown in Figure 1. Non-RN nodes transmit data to RNs for temporary storage. The data is transmitted to the mobile sink when the mobile sink is in the communication range of RNs.

In fact, if the mobile sink is in communication overlapping ranges of sensor nodes, it can communicate directly with more sensor nodes. However, we should know the specific location of the visiting point in the communication overlapping range, because the trajectory length of the mobile sink needs to be calculated according to specific locations. In this paper, a mechanism is designed to select potential visiting points in communication overlapping ranges of sensor nodes rather than locations of sensor nodes. Then, MOPSO is applied to obtain the set of visiting points and plan the trajectory of the mobile sink, which can shorten the trajectory and reduce the delay in data delivery. In most studies, the trajectory planning of the mobile sink is generally divided into two stages: RNs selection and trajectory planning [17]–[19], [22]. In the RNs selection, the weight function is often used. After selecting the set of RNs, how to plan the trajectory for these selected RNs is actually a traveling salesman problem (TSP). Thus, algorithms for TSP are often used in the trajectory planning [13], [17], [18], [23]. If there is a maximum allowable trajectory length, the length of the trajectory will be calculated after selecting a RN to prevent exceeding the maximum allowable length. The trajectory planning proposed in this paper is essentially different from algorithms for TSP, because the set of RNs is uncertain before trajectory planning (RNs selection and trajectory planning are carried out simultaneously in our work). To achieve trajectory planning, an effective trajectory encoding method is designed to generate a closed-loop trajectory containing an unfixed number of visiting points. The main contributions of this paper are as follows:

- In this paper, we use the mobile sink to visit communication overlapping ranges of sensor nodes rather than locations of sensor nodes, so that the mobile sink can communicate directly with more sensor nodes (i.e., more nodes are selected as RNs).
- MOPSO is applied to obtain the set of visiting points and plan the trajectory of the mobile sink, which can shorten the trajectory and reduce the delay in data delivery.
- A trajectory encoding method is designed for the mobile sink, which can generate a closed-loop trajectory containing an unfixed number of visiting points.
- Extensive simulations have shown that the proposed EETP outperforms the existing WRP, CB and the MOPSO-based algorithm, in terms of delay in data delivery, network lifetime and energy consumption.

The rest of this paper is arranged as follows. Section II reviews related works. Section III presents the network environment and problem formulation. In section IV, the selection of potential visiting points and trajectory planning of the mobile sink are illustrated in detail. The performance evaluation is described in section V. Section VI gives the conclusion.

II. RELATED WORKS

There are two main goals in the research on mobile sinks: reducing the delay in data delivery and prolonging the network lifetime. Next, we review related works.

In [12], the mobile agent with a sink acts as the mobile sink. Before data overflow of sensor nodes, the mobile sink completes data collection by visiting each sensor node. However, due to the long trajectory, the delay in data delivery is large. The authors in [13] converted visiting points from locations of sensor nodes to the boundary of their communication ranges. At some visiting points, the mobile sink can communicate with multiple sensor nodes within one-hop distance. Without multi-hop transmission, the trajectory of the mobile sink is still too long and the delay in data delivery is high. Therefore, the mobile sink is only allowed to visit some of sensor nodes (i.e., multi-hop transmission is adopted). The authors

TABLE 1. Summary of related works.

Algorithms		Using TSP algorithms	Visiting the position of sensor nodes	Minimizing the trajectory length of the mobile sink	Considering the load balance of RNs
Previous	TCBDGA [14]	✓	✓		
	SHA [13]	✓		✓	
	CB [17]	✓	✓		
	WRP [18]	✓	✓		
	EAPC [19]		✓		
	PFOA [22]	✓	✓		
	MOPSO-based [23]		✓		
Proposed	EETP			✓	✓

in [14] proposed a tree-cluster-based data gathering algorithm TCBDGA. Initially, the algorithm establishes a tree according to the weight of sensor nodes, which considers the distance to the root node, the number of its two-hop neighbors and the residual energy of its one-hop neighbors. Then, the root node is selected as RN and other nodes are selected as RNs according to the traffic load of sensor nodes and hops to root nodes. After selecting RNs, TSP algorithm is used to plan the trajectory of the mobile sink. TCBDGA focuses on reducing the overall load, but without considering load balance of RNs and the maximum allowable trajectory length of the mobile sink. The authors in [17] proposed a cluster-based algorithm CB, which is similar to the k-means algorithm. Under the trajectory length constraint of the mobile sink, the maximum number of clusters is determined. Then, the central node is selected as RN in each cluster, according to the minimum total hop distance to other nodes. After selecting RNs, TSP algorithm is used to plan the trajectory of the mobile sink. However, selected RNs are too few and CB does not consider the load balance of RNs. To obtain most RNs, the authors in [19] proposed an algorithm EAPC, which considers the distance between two consecutive RNs. Initially, RNs are selected according to the number of the saved data packet and the distance between two consecutive RNs. Then, the convex polygon is adopted to plan the trajectory of the mobile sink. EAPC can get many RNs, but without considering the load imbalance of RNs, it will shorten the network lifetime. The authors in [23] proposed a MOPSO-based algorithm. There are two objective functions in this algorithm. One is to minimize the maximum average transmission distance of RNs so that non-RN nodes can be relatively concentrated around RNs. The other objective function is to minimize the maximum total hop counts of RNs, which can shorten the path of multi-hop data transmission. To achieve these two goals, MOPSO is applied to select a fixed number of RNs. Then, the algorithm for TSP is used to plan the trajectory of the mobile sink. However, the trajectory length is not limited. If the trajectory of the mobile sink is too long, the delay in data delivery will increase. In [22], the author proposed an energy-aware data collection algorithm PFOA. First, sensor nodes are divided into several clusters. Then, the node with high energy acts as the cluster head in each cluster. Finally, PFOA combines ant colony algorithm and evolutionary algorithm to plan the trajectory of the mobile sink. PFOA effectively reduces hop counts, but the load imbalance of RNs still remains unsolved.

Table 1 summarizes related works. Most studies apply TSP algorithms to plan the trajectory of the mobile sink and their time complexities are generally high. In addition, most of them ignore the load balance of RNs. Therefore, this paper proposes an effective trajectory planning algorithm to shorten the trajectory length of the mobile sink and balance the load of RNs. Due to communication ranges of sensor nodes, the mobile sink in this work visits communication overlapping ranges of sensor nodes rather than their own locations, which can shorten the trajectory length of the mobile sink.

III. NETWORK ENVIRONMENT AND PROBLEM FORMULATION

In a monitoring area, we assume that n static sensor nodes $S=\{s_1, s_2, s_3, \cdots, s_n\}$ are randomly deployed. Once these sensor nodes are placed, they can no longer be moved. We assume that the communication range R of the mobile sink is the same as that of all sensor nodes. Non-RN nodes transmit data to RNs for temporary storage. When the mobile sink is in communication ranges of RNs, the stored data can be directly transmitted to the mobile sink. The mobile sink starts from its initial position with a speed v to collect data and eventually returns to its initial position. The data transmission speed is much faster than that of the mobile sink, so the delay in data delivery depends on the trajectory length of the mobile sink. The shorter the trajectory, the lower the delay in data delivery. To shorten the trajectory, we set the first objective function as is shown in Equation (1). To balance the load among RNs, we set the second objective function as is shown in Equation (2) (i.e., minimize the standard deviation of child nodes among RNs), which helps to prolong the network lifetime. To obtain a sufficient number of RNs to reduce the energy consumption caused by multi-hop transmission, we set the third objective function (convert the maximization into a minimization by adding a minus), as is shown in Equation (3).

$$\text{Minimize:} \quad L = \sum_{\forall i,j \in P, i \neq j} (D_{ij} x_{ij}), \tag{1}$$

TABLE 2. Summary of notations.

Notation	Description
n	Number of sensor nodes
S	Set of sensor nodes: $\{s_1, s_2, s_3 \cdots, s_n\}$
v	Speed of the mobile sink
N	Number of potential visiting points
W_s	Standard deviation of child nodes among RNs
G	Negative value of the number of RNs
A	Set of non-RN nodes
B	Set of RNs
P	Set of potential visiting points: $\{p_1, p_2, p_3 \cdots, p_N\}$
x_{ij}	If the mobile sink directly passes by p_i and p_j, x_{ij} is 1, otherwise 0
D_{ij}	Distance between p_i and p_j
y_j	If the p_i is visited, y_j is 1, otherwise 0
u_i	Rate of data generation for sensor node s_i
h_{ij}	If the sensor node s_i transmit its data to the mobile sink through s_j, h_{ij} is 1, otherwise 0
L	Trajectory length of the mobile sink
L_{max}	Allowed trajectory length of the mobile sink
d_{ij}	Distance between s_i and s_j
c_i	Number of child nodes of $s_i (i \in B)$
$\bar{c}$	Average number of child nodes among RNs
b_j	Cache space of the sensor node s_j

$$W_s = \sqrt{\frac{1}{|B|}\sum_{\forall i \in B}(c_i - \bar{c})^2}, \tag{2}$$

$$G = -\sum_{i=1}^{n} y_i, \tag{3}$$

$$\text{Subject to: } b_j \geq \sum_{\forall i \in A}(h_{ij} u_j L/v), \quad \forall j \in B, \tag{4}$$

$$\sum_{\forall i,j \in P, i \neq j}(D_{ij} x_{ij}) \leqslant L_{max}, \tag{5}$$

$$\sum_{\forall i,j \in P}(x_{ij} + x_{ji}) = 2y_i, \tag{6}$$

$$\sum_{\forall i,j \in P}(x_{0i} + x_{i0}) = 2, \tag{7}$$

where constraint (4) ensures that cache spaces of RNs do not overflow. Constraint (5) ensures that the trajectory length of the mobile sink will not exceed the allowed length L_{max}. Constraint (6) ensures that the obtained trajectory is continuous and complete, not fractured. Constraint (7) ensures that the mobile sink returns to its initial position. In addition, descriptions of notations used in this paper are summarized in Table 2.

In this work, the energy model in [24] is adopted to update energy consumption of sensor nodes. If one data packet (ι bits) is sent from node s_i to s_j, energy consumption of node s_i and s_j can be calculated by Equation (8) and Equation (9), respectively.

$$E_t(\iota, d_{ij}) = \begin{cases} \iota E_{elec} + \alpha_{fs} \iota d_{ij}^2, & d_{ij} < d_r, \\ \iota E_{elec} + \alpha_{mp} \iota d_{ij}^4, & d_{ij} \geqslant d_r, \end{cases} \tag{8}$$

$$E_r = \iota E_{elec}, \tag{9}$$

where d_{ij} is the distance between node s_i and s_j. E_{elec} is energy consumption per bit for the electronics circuit. α_{fs} and α_{mp} are the energy consumption factors of amplification for the free space and multipath radio models, respectively. d_r is the threshold distance and it is equal to $\sqrt{\alpha_{fs}/\alpha_{mp}}$.

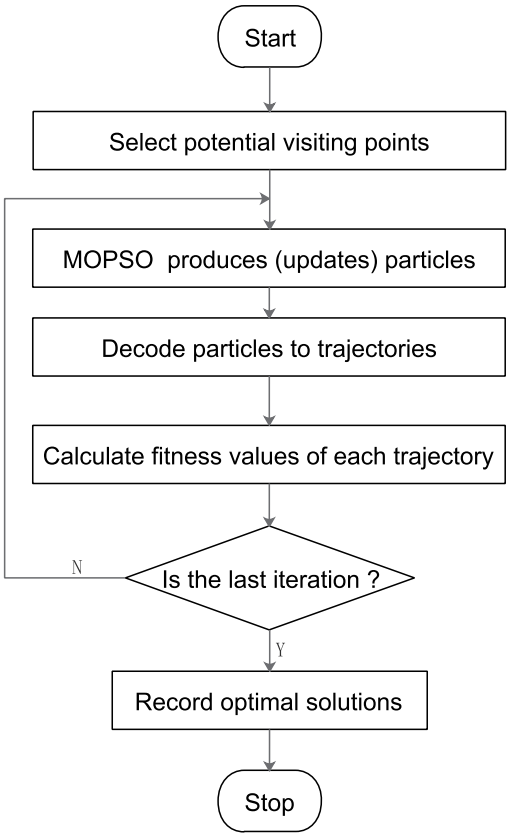

FIGURE 2. Flowchart of the proposed EETP.

IV. THE PROPOSED ALGORITHM

This section describes the proposed work in detail. The proposed EETP can be divided into two phases: selection of potential visiting points and trajectory planning. In the selection of potential visiting points, potential locations of visiting points are selected within communication overlapping ranges of sensor nodes. In the trajectory planning phase, MOPSO is used to select the optimal visiting points and plan the trajectory of the mobile sink. The flowchart of EETP is shown in Figure 2.

A. SELECTION OF POTENTIAL VISITING POINTS

The purpose of this phase is to find potential visiting points within communication overlapping ranges of sensor nodes.

The higher coverage level of the area means that the mobile sink can directly communicate with more nodes in this area (coverage level refers to the number of times covered by communication ranges of sensor nodes). As is shown in Figure 3(a), the area with the highest coverage level is z_1. If the mobile sink visits a point in the area z_1, such as point p_1 in Figure 3(b), it can communicate directly with s_1, s_2 and s_3. However, we need to know the exact location of the visiting point so that we can calculate the trajectory

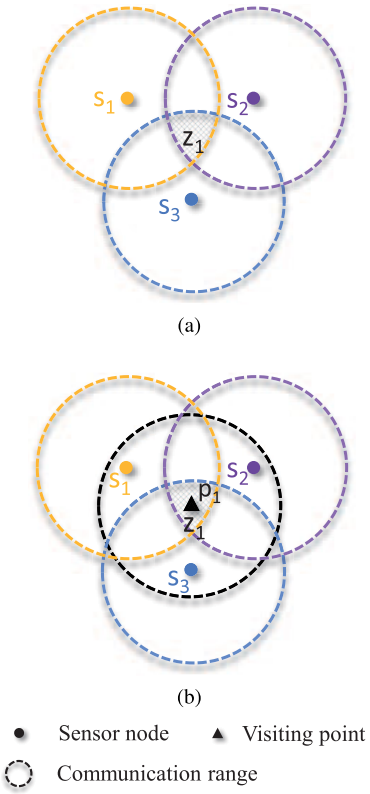

FIGURE 3. An example of communication overlapping ranges of sensor nodes.

length of the mobile sink. To solve this problem, we adopt the following method to find a representative visiting point within communication overlapping ranges of sensor nodes.

Algorithm 1 shows the entire process. P denotes the set of potential visiting points and P=$\{p_1, p_2, p_3, \cdots, p_N\}$, where N is the total number of potential visiting points. Initially, all intersection points between communication ranges of sensor nodes are calculated (introduced in the next paragraph). Then, intersection points with the highest coverage level can be found in each communication range of sensor nodes and their center is calculated as a potential visiting point by Equation (10). Finally, same potential visiting points are deleted. By doing so, the set of potential visiting points can be obtained.

$$p_i\left(\frac{1}{k}\sum_{j=1}^{k}x_j, \frac{1}{k}\sum_{j=1}^{k}y_j\right), \tag{10}$$

where $(x_1, y_1), (x_2, y_2), \cdots, (x_k, y_k)$ are locations of intersection points with highest coverage level in a communication range of sensor nodes.

To quickly get intersection points of communication ranges, we adopt the following measure. Figure 4 is used as an example to illustrate how to calculate intersection points of communication ranges. We assume that the coordinate positions of s_a and s_b are $s_a(x_a, y_a)$ and $s_b(x_b, y_b)$ respectively. d_{ab} denotes the distance between s_a and s_b. If d_{ab} is less than $2R$, there are two intersection points between communication

Algorithm 1 Potential Visiting Points Selection

Input: Communication range R and locations of sensor nodes
Output: Potential visiting points set P

for i=1, 2, 3, . . . n **do**
  for j=$i+1, i+2, i+3, \ldots n$ **do**
    if $0<d_{ij}<2R$ **then** //*two intersection points*//
      Find intersection points by Equation (19) and Equation (20)
      Save intersection points: s_i: $K \leftarrow \{K, k_1, k_2\}$
    end if
    if d_{ij}=$2R$ **then** //*one intersection point*//
      Find the intersection point:
      $k_1(\frac{x_a+x_b}{2}, \frac{y_a+y_b}{2})$
      Save the intersection point: s_i: $K \leftarrow \{K, k_1\}$
    end if
  end for
end for
for i=1, 2, 3, . . . n **do**
  Calculate the center of intersection points with highest coverage level in each communication range by Equation (10)
  The obtained center is selected as a potential visiting point: $P \leftarrow \{P, p_i\}$
end for
The same p_i is deleted
return Set of potential visiting points P

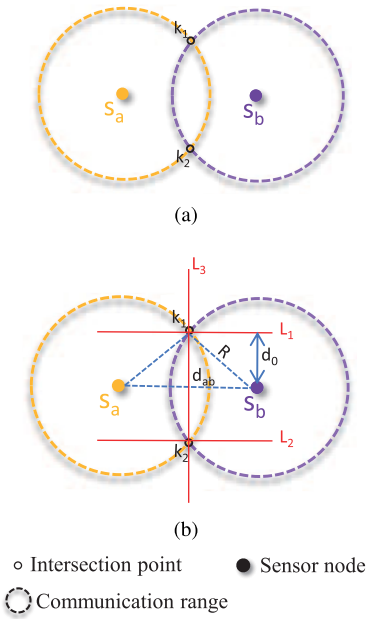

FIGURE 4. An example of calculating intersection points of communication ranges.

ranges of s_a and s_b. As is shown in Figure 4(a), k_1 and k_2 are intersection points between communication ranges of nodes s_a and s_b. L_{ab} is a straight line passing through nodes s_a and s_b. L_1 and L_2 are straight lines that are parallel to the straight line L_{ab} and pass through k_1 and k_2 respectively.

L_3 is a straight line passing through k_1 and k_2. Positions of k_1 and k_2 can be calculated according to straight lines L_1, L_2 and L_3. The specific process is as follows. Initially, according to $s_a(x_a, y_a)$ and $s_b(x_b, y_b)$, the line L_{ab} passing through them can be obtained, as is shown in Equation (11). In Figure 4(b), according to the Pythagorean theorem and the distance formula of the parallel line as is shown in Equation (12), we can get constant terms of general equations of lines L_1 and L_2, as is shown in Equation (13). Thus, general equations of lines L_1 and L_2 are shown in Equation (14) and Equation (15), respectively. Then, Equation (18) (i.e., the straight line L_3) is obtained by subtraction of Equation (16) and Equation (17). Through L_1 and L_3, the location of k_1 can be obtained, as is shown in Equation (19). Through L_2 and L_3, the location of k_2 can be also obtained, as is shown in Equation (20). Therefore, according to R, $s_a(x_a, y_a)$ and $s_b(x_b, y_b)$, the positions of k_1 and k_2 can be calculated by Equation (19) and Equation (20). If d_{ab} is equal to $2R$, there is only one intersection point between communication ranges and its location is $(\frac{x_a+x_b}{2}, \frac{y_a+y_b}{2})$.

$$L_{ab}: \quad A_1x + B_1y + C_{ab} = 0, \tag{11}$$

where $A_1{=}(y_b - y_a)$, $B_1{=}(x_a - x_b)$, $C_{ab}{=}y_ax_b - x_ay_b$.

$$d_0 = \frac{|C_x - C_{ab}|}{\sqrt{A_1^2 + B_1^2}} = \sqrt{R^2 - \left(\frac{d_{ab}}{2}\right)^2}, \tag{12}$$

where d_0 represents the distance between the straight line L_1 and the straight line L_{ab}.

$$C_x: \begin{cases} C_1 = C_{ab} + \sqrt{\left[R^2 - \left(\frac{d_{ab}}{2}\right)^2\right](A_1^2 + B_1^2)}, \\ C_2 = C_{ab} - \sqrt{\left[R^2 - \left(\frac{d_{ab}}{2}\right)^2\right](A_1^2 + B_1^2)}, \end{cases} \tag{13}$$

$$L_1: \quad A_1x + B_1y + C_1 = 0, \tag{14}$$

$$L_2: \quad A_1x + B_1y + C_2 = 0, \tag{15}$$

$$(x - x_a)^2 + (y - y_a)^2 = R^2, \tag{16}$$

$$(x - x_b)^2 + (y - y_b)^2 = R^2, \tag{17}$$

$$L_3: \quad A_3x + B_3y + C_3 = 0, \tag{18}$$

where $A_3{=}2\,(x_b - x_a)$, $B_3{=}2\,(y_b - y_a)$, $C_3{=}y_a^2 - y_b^2 + x_a^2 - x_b^2$.

$$L_1, L_3 \rightarrow \quad k_1\left(\frac{B_1C_3 - B_3C_1}{A_1B_3 - A_3B_1}, \frac{A_3C_1 - A_1C_3}{A_1B_3 - A_3B_1}\right), \tag{19}$$

$$L_2, L_3 \rightarrow \quad k_2\left(\frac{B_1C_3 - B_3C_2}{A_1B_3 - A_3B_1}, \frac{A_3C_2 - A_1C_3}{A_1B_3 - A_3B_1}\right), \tag{20}$$

Next, an example of selecting potential visiting points is given. The distribution of node s_1, s_2, s_3 and s_4 is shown in Figure 5(a). Firstly, all intersection points ($k_1{\sim}k_{12}$) are calculated by Equation (19) and Equation (20). In the communication range of node s_1, intersection points with the highest coverage level are k_2, k_3, k_4 and k_5, as is shown in Figure 5(b). Then, their center acts as a potential visiting point, which is calculated by Equation (10). In communication ranges of s_2, s_3 and s_4, selected potential visiting points are identical to that selected in the communication range of s_1, so only one potential visiting point is reserved.

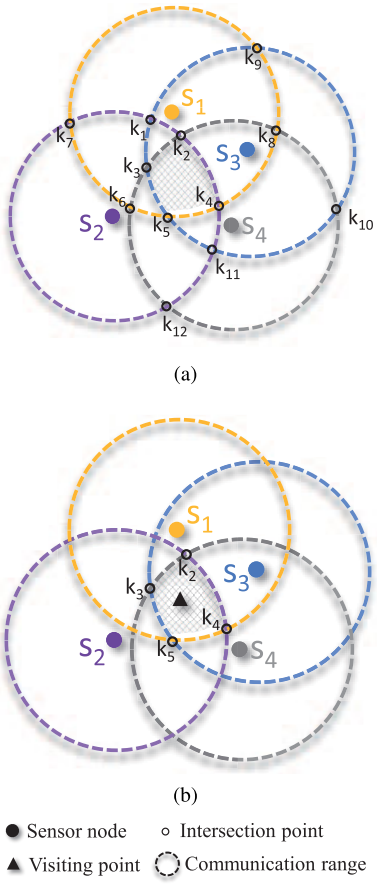

FIGURE 5. An example of selecting the potential visiting point.

B. TRAJECTORY PLANNING

This section first reviews the multi-objective optimization. Then, the trajectory planning of the mobile sink based on MOPSO is presented. The trajectory encoding method used in trajectory planning is discussed in detail at the end of this section.

1) MULTI-OBJECTIVE OPTIMIZATION

A typical multi-objective problem contains k ($\geqslant$2) objective functions and they are usually conflicting [25]. To obtain the optimal solution, they usually need to meet multiple inequality or equality constraints, as is shown in Equation (21).

$$\begin{aligned} &\text{Minimize: } F(x) = \{L(x), W_s(x) \cdots, f_k(x)\} \\ &\text{Subject to: } \begin{cases} h_i(x) = 0, & i = 1, 2, \cdots, q \\ g_j(x) \leqslant 0, & j = 1, 2, \cdots, p \end{cases} \end{aligned} \tag{21}$$

Compared with the single objective optimization, it is difficult to evaluate the advantage of acceptable solutions in multi-objective optimization [26]–[28]. Thus, there is no unique optimal solution, but a set of optimal solutions.

Traditional multi-objective methods transform multi-objective into a single objective, which is inefficient. For example, the weighted summation method needs to adjust parameters to obtain approximate Pareto optimal solutions [20]. However, some heuristic algorithms such as PSO can solve this problem. This group-based search can achieve diversity and globality of solutions, providing efficient algorithms for solving multi-objective problems [29]–[31].

2) TRAJECTORY PLANNING BASED ON MOPSO

This section aims to design a trajectory of the mobile sink and the process is shown in Algorithm 2. Initially, we apply Floyd algorithm to establish a minimum hop distance table for sensor nodes so that the non-RN node can quickly find the nearest RNs (according to the hop distance). In other words, once the RNs are selected, the child nodes of the RNs can be determined by querying this table. Then, a certain number of particles are randomly generated and each particle is encoded to a trajectory (discussed in subsequent trajectory encoding), which contains an unfixed number of visiting points. According to visiting points included in the trajectory, the set of RNs can be also determined. Therefore, the trajectory length of the mobile sink, the number of RNs and child nodes of each RNs can be further obtained (i.e., the fitness value of the trajectory can be obtained). Additionally, the archive A_r is adopted to preserve the obtained non-inferior solution and supply the global leader of particles [32]. The next steps are basically to repeat the above process, but the search direction of particles depends on local leaders and global leaders. The local leader of the particle is the optimal position that the particle has searched for in the current iteration. The global leader of the particle is chosen according to the crowding level of particles in [33]. In this paper, we use the location update of the particle in [34], as is shown in Equation (22) and Equation (23):

$$\upsilon_{t+1} = a\left[\omega\upsilon_k + \varphi_1\xi_1\left(\theta_p - \theta_t\right) + \varphi_2\xi_2\left(\theta_g - \theta_t\right)\right], \quad (22)$$

$$\theta_{t+1} = \theta_t + \upsilon_{t+1}, \quad (23)$$

where υ_{t+1} is the velocity of the particle in the next iteration and υ_t is the current velocity of the particle. θ_{t+1} is the location of the particle in the next iteration and θ_t is the current location of the particle. φ_1 and φ_2 are local learning factors and global learning factors, respectively. ξ_1 and ξ_2 are random numbers between 0 and 1. a is a regulatory factor. θ_p and θ_g are the local leader and the global leader, respectively. ω is the inertia weight used to improve the quality of the particles and its value varies from large to small with the number of iterations.

3) TRAJECTORY ENCODING

In the iteration process of trajectory, vectors in particles of MOPSO need to be encoded into trajectories. The authors in [35] proposed a trajectory encoding algorithm for PSO, which is used to get the shortest trajectory between the source node and the destination node. However, the model in this work is different from that in [35]. On the one hand, the trajectory we need is a closed loop (i.e., the first point of the trajectory has the same location as the last point of the trajectory). On the other hand, the maximum allowed trajectory length for the mobile sink is L_{max}. It means that the mobile sink can reach an area not more than $L_{max}/2$ away from its initial position, as is shown in Figure 6. Otherwise, its total trajectory length will exceed L_{max}. Therefore, the potential visiting point that the mobile sink cannot visit will not be included in the trajectory. In addition, there is a direct link between any two potential visiting points.

According to these characteristics of the model in this work, we improve the trajectory encoding algorithm. Algorithm 3 shows how the algorithm works. It takes a particle and the L_{max} as inputs. Then, it outputs a trajectory sequence (numbers in the trajectory sequence are the serial numbers of potential visiting points). To obtain a closed loop, we add two virtual points (p_1 and p_{N+2}) with the same location as the initial position of the mobile sink. The total number of potential visiting points is N, so the dimension of

Algorithm 2 Trajectory Planning Based on MOPSO

Input: Set of sensor nodes S, set of potential visiting points P, maximum allowed trajectory length L_{max}
Output: Pareto optimal set, trajectories of the mobile sink and set of RNs

Establish a minimum hop distance table H_d by Floyd
Initialize the particle $X_i(X_i \in X)$
for i=1, 2, 3, $\cdots$, N_p **do** //* population size N_p *//
    Decode the particle to a trajectory by Algorithm 3:
    $X_i \rightarrow Path_i$
    Get the set of RNs
    Get the child nodes of RNs by querying table H_d
    Calculate fitness values of the trajectory:
    $Path_i \rightarrow L^i, W_s^i, G^i$
end for
Get archive of non-inferior solutions: Ar
for k=1, 2, 3, $\cdots$, $iter_{max}$ **do**
    Select personal leaders and global leaders
    Update position of particles by Equation (22) and Equation (23)
    for i=1, 2, 3, $\cdots$, N_p **do**
        Decode the particle to a trajectory:
        $X_i \rightarrow Path_i$
        Get the set of RNs
        Get the child nodes of RNs by querying table H_d
        Calculate fitness values of the trajectory:
        $Path_i \rightarrow L^i, W_s^i, G^i$
    end for
    Update the Ar
end for
Get the Pareto optimal set
Select a suitable solution from Pareto optimal set
return Pareto optimal set, trajectories of the mobile sink and set of RNs

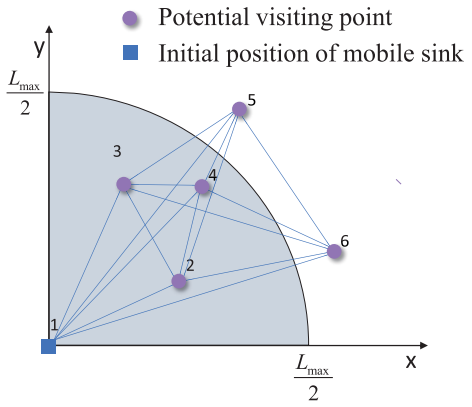

FIGURE 6. An example of the area mobile sink can visit.

Algorithm 3 Improved Trajectory Encoding for Closed Loop

Input: A particle $X=\{x_1, x_2, \cdots, x_{N+1}, x_{N+2}\}$ and L_{max}

Output: A trajectory containing an unfixed number of visiting points.

1: The initial position of the mobile sink is the first point of the trajectory.
2: $k=0$, $X^k=X$
3: $k=k+1$, $M^k_{add}=p_1$, $X^k(1)=-W_\infty$
4: **while** $M^k_{add} \neq p_{N+2}$ and $k \leqslant N+2$ **do**
5: the point p_i with highest priority is considered as next point of the trajectory:
6: $M^k_{add}=p_i$, $X^k(i)=-W_\infty$
7: **if** the visiting point p_i is within the area the mobile sink can visit **then**
8: $M=\{M, M^k_{add}\}$
9: **end if**
10: $k=k+1$
11: **end while**
12: **if** the obtained trajectory is closed loop **then**
13: **return** a valid trajectory
14: **else**
15: **return** an invalid trajectory
16: **end if**

the particle is $N+2$. The position vector of the particle acts as the priority of potential visiting points. Initially, p_1 is selected as the first point of the trajectory (i.e., the initial position of the mobile sink) and the first position in the particle is given a large negative value $-W_\infty$. Then, the point p_i with the highest priority is considered as the next point of the trajectory and the corresponding position in the particle is given a large negative value $-W_\infty$. If the p_i is within the area the mobile sink can visit, it is selected as the next visiting point. The iteration process continues until the selected visiting point is p_{N+2}. If the number of the iteration exceeds $N+2$, the iteration process also stops. If the obtained trajectory is a closed loop, it is valid. Otherwise, it is invalid and the corresponding objective function evaluation returns a very large value as a penalty.

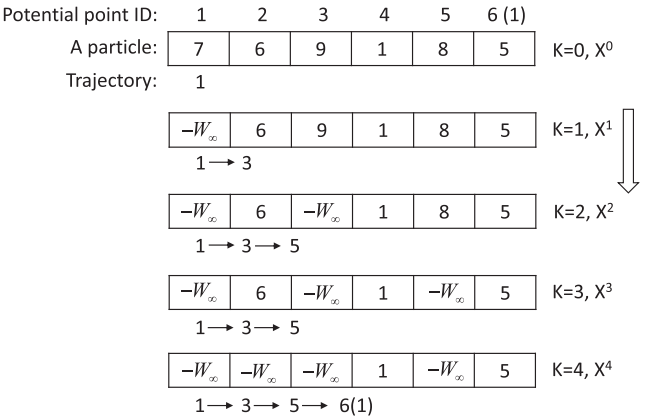

FIGURE 7. An example of trajectory encoding.

Next, an example of trajectory encoding is given in Figure 7. We assume that potential visiting points are p_2, p_3, p_4, p_5 and only point p_3, p_5 are within the area that the mobile sink can visit. p_1 and p_6 are virtual visiting points and their location is the same as the initial position of the mobile sink. Therefore, p_1, p_3, p_5 and p_6 are within the area that the mobile sink can visit and the dimension of the particle $N+2$ is 6. Initially, p_1 is selected as the first point of the trajectory (i.e., trajectory sequence: {1}) and the first position in the particle is given a large negative value $-W_\infty$. Then, the point p_i with the highest priority is p_3 and the corresponding position in the particle is given a large negative value $-W_\infty$. p_3 is within the area that the mobile sink can visit, so it is selected as the next point of the trajectory (i.e., trajectory sequence: {1, 3}). The iteration continues. The point p_i with the highest priority is p_5 and the corresponding position in the particle is given a large negative value $-W_\infty$. p_5 is within the area that the mobile sink can visit, so it is selected as the next point of the trajectory (i.e., trajectory sequence: {1, 3, 5}). The point p_i with the highest priority is p_2 and the corresponding position in the particle is given a large negative value $-W_\infty$. p_2 is out of the area that the mobile sink can visit, so it cannot be selected. The next point with the highest priority is p_6 and it is within the area the mobile sink can visit. In this iteration, the selected p_6 is the p_{N+2} (i.e., the initial position of the mobile sink), so we have got a closed-loop trajectory by Algorithm 3 (i.e., trajectory sequence: {1, 3, 5, 6}).

C. COMPLEXITY ANALYSIS

In the selection phase of potential visiting points, the time cost of finding all intersections is $O(n^2)$ and the time cost of selecting visiting points is $O(n^1)$. Thus, the time complexity of this phase is $O(n^2) + O(n^1) = O(n^2)$. In the trajectory planning phase, the time complexity of the establishment of the minimum hop distance table is $O(n^3)$. In MOPSO, the time cost of updating A_r is the largest and study [36] has proved that the time complexity of updating A_r is $O(3(|A_r| + \varrho)^2)=O((|A_r| + \varrho)^2)$, where $|A_r|$ is the capacity of A_r and ϱ is the number of particles. Therefore, the time complexity of EETP is $O(n^2) + O(n^3) + O((|A_r| + \varrho)^2) = O(n^3)$.

TABLE 3. Parameters of the sensor network.

Parameter	Value
Size of deployment area	300m×300m
Number of sensor nodes (n)	100 to 200
Communication range of sensor nodes (R)	30 to 50m
Length of packet (ι)	240bits
Allowed length of the trajectory (L_{max})	650m
Speed of the mobile sink (v)	1.5 m/s
Initial energy of sensor nodes (E_0)	5.0J
Buffer size of sensor nodes	10KB
Energy required by amplifier in free space radio (α_{fs})	10pJ /bit /m^2
Energy required by amplifier in multipath radio (α_{mp})	0.0013pJ /bit /m^4
Energy consumption per bit for the electronics circuit (E_{elec})	50nJ /bit

TABLE 4. Parameters in MOPSO.

Parameter	Value
Population size	100
Range of position vectors in particles	[-50, 50]
Maximum number of iterations	300
Local learning factors (φ_1)	2
Global learning factors (φ_2)	2
Regulatory factor (a)	0.729

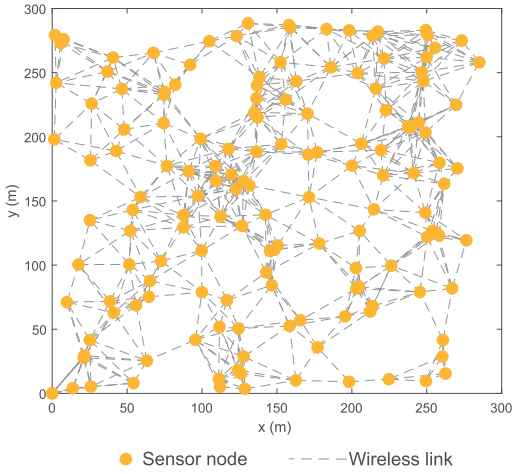

FIGURE 8. Network scenario ($n = 150$, $R = 35$m).

V. PERFORMANCE EVALUATION

In this section, the performance of the proposed EETP is evaluated by the simulation, compared with the existing algorithm EAPC, CB and the MOPSO-based algorithm. The simulation is conducted on MATLAB and simulation parameters are shown in Table 3 and Table 4. The network scenario is shown in Figure 8 and any line between two nodes indicates that they can communicate directly with each other. According to Algorithm 1, potential visiting points can be found, as is shown in Figure 9. When the mobile sink visits the potential visiting point, the mobile sink can communicate directly with sensor nodes in the communication range of the mobile sink. Under constraints (4) to (7), the set of Pareto optimal solutions is obtained by MOPSO, as is shown in Figure 10.

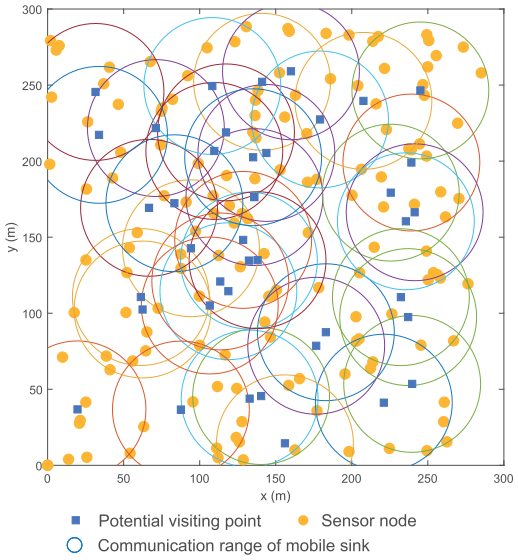

FIGURE 9. Selecting potential visiting points ($n = 150$, $R = 35$m).

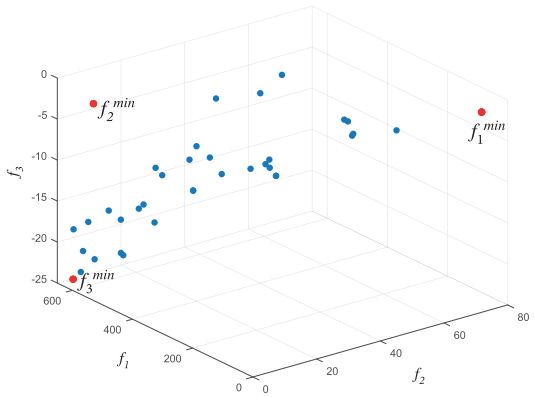

FIGURE 10. Pareto optimal solutions ($n = 150$, $R = 35$m).

TABLE 5. Optimal solution of each object function ($n = 150$, $R = 35$m).

Solution	$L(m)$	W_s	G	NL(round)
L^{min}	86.226	79.965	-3	1336
W_s^{min}	529.71	0	-1	1281
G^{min}	648.305	5.123	-25	9242

L is the trajectory length of the mobile sink. W_s is the standard deviation of child nodes among RNs. G is the negative value of the number of RNs. NL means the network lifetime.

Pareto set can provide decision makers with the best trade-off or near-optimal solutions between key objectives. Decision makers need to choose a suitable trade-off solution according to priorities or purposes. If the delay in data delivery needs to be further reduced, decision makers can choose a solution with a shorter trajectory length. However, the shorter trajectory length is not conducive to prolonging the network lifetime.

Table 5 analyzes the optimal solution of each objective function. It can be seen that the optimal solution G^{min} can

TABLE 6. Optimal solutions with different trajectory length ($n = 150$, $R = 35$m).

Ranges (m)	$L(m)$	W_s	G	NL(round)
100~0	86.226	79.965	-3	1336
200~100	178.907	48.079	-4	1855
300~200	240.523	29.997	-8	2394
400~300	350.828	23.369	-9	2524
500~400	436.877	10.405	-15	6500
600~500	511.456	7.615	-20	7263
700~600	633.421	5.823	-24	9242

L is the trajectory length of the mobile sink. W_s is the standard deviation of child nodes among RNs. G is the negative value of the number of RNs. NL means the network lifetime.

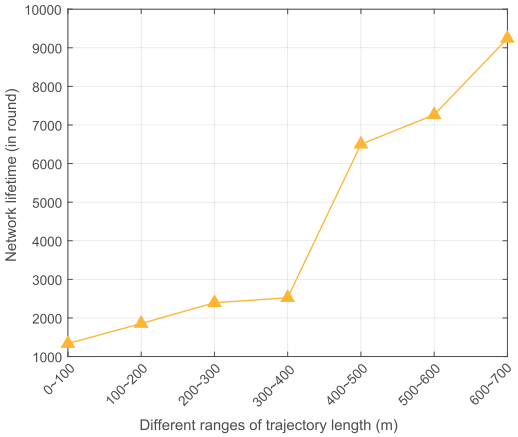

FIGURE 11. Network lifetime in solutions with different ranges of trajectory length.

prolong the network lifetime more effectively. However, the corresponding trajectory length is the longest. Table 6 further analyzes the impact of different trajectory length on network lifetime (a trade-off solution is selected in different ranges of the trajectory length). It is clear that the shorter the trajectory of the mobile sink, the fewer number of RNs, leading to the reduction of network lifetime. This is because fewer RNs reduce the number of multi-hop transmissions. When the trajectory length is more than 400m, the network lifetime increases rapidly, as is shown in Figure 11. For a sufficiently long network lifetime, a trade-off solution with a trajectory length greater than 400m is chosen to compare the performance of EETP with other algorithms. By doing so, we can not only get a long enough lifetime, but also the trajectory length is not too long. When a trade-off solution is selected, the optimal set of visiting points, RNs and trajectory of the mobile sink can be determined. Non-RN nodes that are one-hop distance away from the near visiting point directly transmit their data to the mobile sink. According to the minimum number of hop counts, other non-RN nodes transmit data to RNs for temporary storage. The data is transmitted to the mobile sink when the mobile sink visits the near visiting point, as is shown in Figure 12.

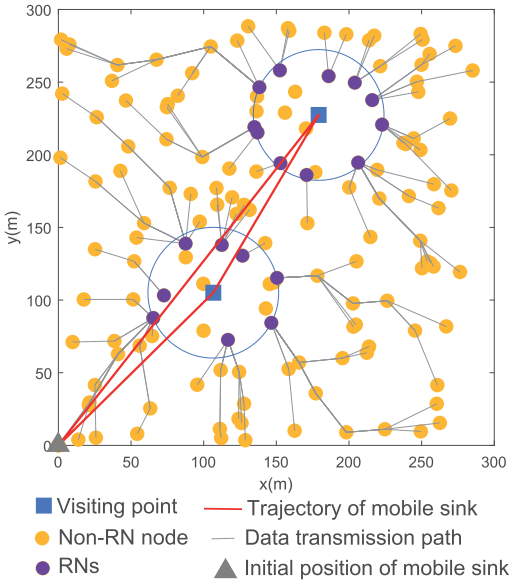

FIGURE 12. Trajectory planning of the mobile sink.

A. NETWORK LIFETIME

Figure 13 compares the network lifetime of the four algorithms under different n. The network lifetime is defined as the time from the start of data collection to the energy exhaustion of the first node. With the increase of n, the network lifetime will be shortened gradually. This is because the average load of RNs and unbalanced energy consumption of RNs also increase as n increases. It is clear that the proposed EETP is superior to EAPC, CB and the MOPSO-based algorithm in terms of network lifetime. Because the mobile sink in EETP is only allowed to collect data within the communication overlapping ranges of sensor nodes, it can communicate with more nodes within one-hop distance, which means that the energy consumption of multi-hop transmissions can be significantly reduced. Therefore, EETP is superior to EAPC, CB and the MOPSO-based algorithm, in terms of the network lifetime. In the MOPSO-based algorithm, a limited number of RNs sufficiently reduce the multi-hop transmission by minimizing the maximum total hop counts of RNs. Therefore, the network lifetime in the MOPSO-based algorithm is longer than that in EAPC. EAPC focuses on the distance between two consecutive RNs, so there are more RNs in EAPC than that in CB and the network lifetime in EAPC is longer than that in CB.

In Figure 14, we further analyze the network lifetime of the four algorithms under different communication ranges R. According to the obtained results, the superiority of EETP is further proved. In EETP, the increase of R means that more nodes can act as RNs and more available transmission paths from non-RN nodes to RNs can be obtained. In EAPC, CB and the MOPSO-based algorithm, this also means that more available transmission paths from non-RN to RN can be also obtained, but it has little effect on the number of RNs. Thus, the network lifetime in EETP is longer than that in other three algorithms under different R.

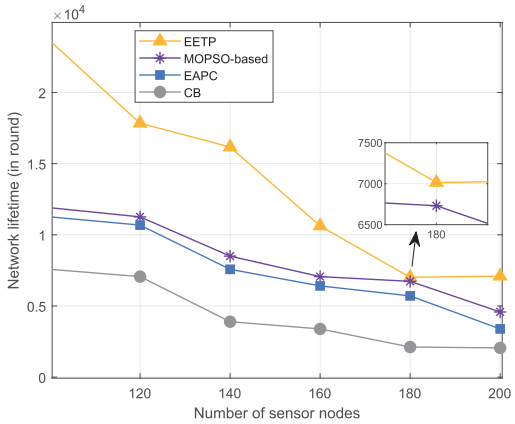

FIGURE 13. Network lifetime with different n ($R = 35$m).

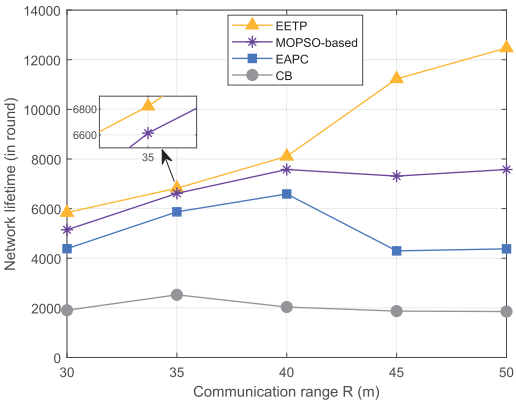

FIGURE 14. Network lifetime with different R ($n = 190$).

B. DELAY IN DATA DELIVERY

The data transmission speed is much faster than that of the mobile sink, so the delay in data delivery is mainly affected by the trajectory length of the mobile sink. As can be seen from Figure 15, among the four algorithms, the trajectory in the proposed EETP is the shortest with different n. That is to say, the delay in data delivery in EETP is the lowest. Figure 16 also proves that the trajectory in EETP is the shortest. This is because the first objective function of EETP is to minimize the trajectory length of the mobile sink. However, both EAPC and CB make full use of the maximum allowed trajectory length to obtain more RNs. The former obtains more RNs by considering the distance between two consecutive nodes. The latter obtains more RNs by maximizing the number of clusters (one RNs per cluster). In the MOPSO-based algorithm, the trajectory length of mobile sink is not restricted, so the trajectory length is the longest. Therefore, in terms of the delay in data delivery, the proposed EETP is superior to EAPC, CB and the MOPSO-based algorithm.

C. RENDEZVOUS NODE (RN)

The number of RNs and the load of RNs affect energy consumption of the network. More RNs are beneficial to better reduce energy consumption caused by multi-hop transmission. The load balance of RNs can promote the

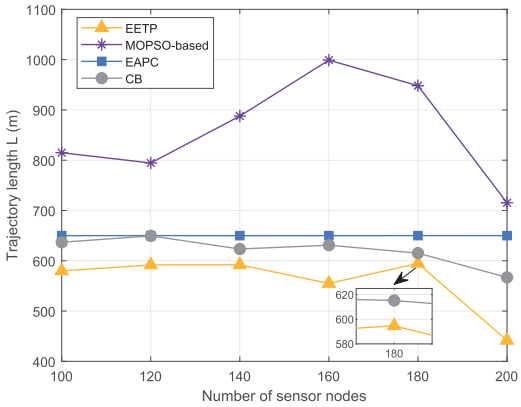

FIGURE 15. Trajectory length of the mobile sink with different n ($R = 35$m).

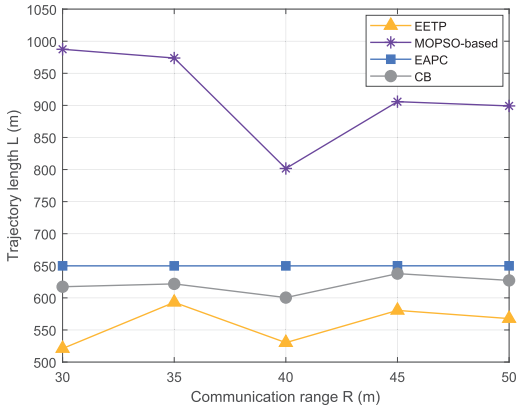

FIGURE 16. Trajectory length of the mobile sink with different R ($n = 190$).

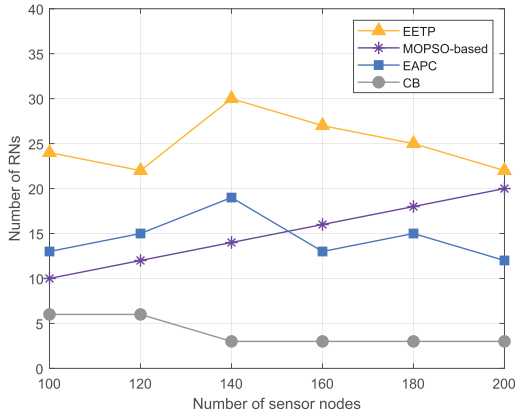

FIGURE 17. Number of RNs with different n ($R = 35$m).

balance of network energy consumption and further prolong the network lifetime. As can be seen from Figure 17 and Figure 18, the number of RNs in CB is less than that in EAPC, EETP and the MOPSO-based algorithm. The number of RNs in EETP is more than that in EAPC. In CB, the number of RNs is determined by the number of clusters, because one RNs (cluster head) is selected in each cluster. In EAPC, the nearest node to RNs is more likely to be selected as RNs

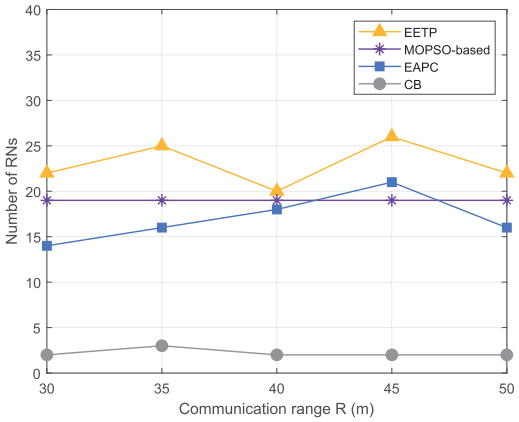

FIGURE 18. Number of RNs with different R ($n = 190$).

TABLE 7. Number of RNs in different network sizes ($R = 35$m, $n = 100 \sim 200$).

n	Number of Visiting points (EETP)	Number of RNs			
		EETP	EAPC	CB	MOPSO-based
100	5	24	13	6	10
120	7	22	15	6	12
140	5	30	19	3	14
160	5	27	13	3	16
180	5	25	15	3	18
200	3	22	12	3	20

because the distance between two RNs is considered in the weight function. Therefore, the average distance between two RNs in EAPC is smaller than that in CB and the number of RNs in EAPC is more than that in CB under the constraint of limited trajectory length. In the MOPSO-based algorithm, the number of RNs has been specified as a definite constant in the algorithm design (i.e., 10% of n). As the number of sensor nodes increases, the number of RNs increases without the considering maximum allowable trajectory length. However, if the maximum allowable trajectory length is taken into account, the number of RNs will also be limited. In EETP, the mobile sink can directly communicate with more RNs at selected visiting points and a few visiting points can contain enough number of RNs, as is shown in Table 7. In addition, as can be seen from Figure 19, the standard deviation of RNs load in EETP is smaller than that in the other two algorithms. This is because the second objective function in EETP is to minimize the standard deviation of child nodes among RNs. Although there are many RNs in EAPC, it does not consider the load balance of RNs. CB focuses on the total hop counts, but also ignores the load balance of RNs. Figure 20 also shows that the load of RNs in EETP is the most balanced, compared with the other two algorithms.

D. HOP COUNTS

Table 8 analyses total hop counts of the four algorithms under different n. The total number of hop counts can reflect the

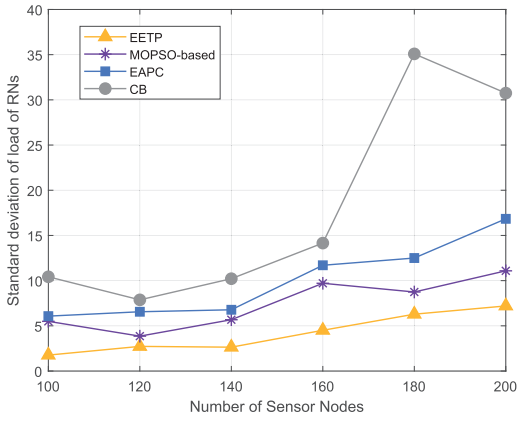

FIGURE 19. Standard deviation of load of RNs with different n ($R = 35$m).

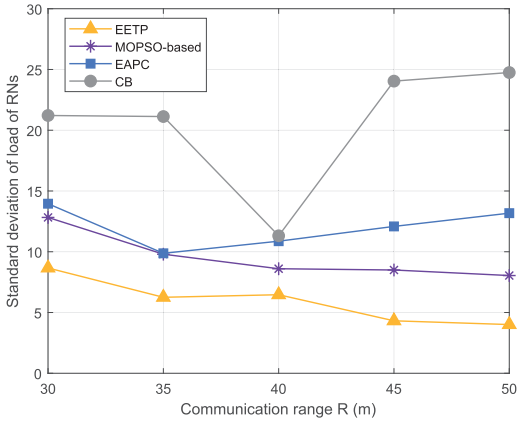

FIGURE 20. Standard deviation of load of RNs with different R ($n = 190$).

TABLE 8. Hop counts with different n ($R = 35$m, $n = 100 \sim 200$).

n	Total hop counts			
	EETP	MOPSO-based	EAPC	CB
100	72	130	396	167
120	100	194	428	199
140	114	198	548	296
160	265	295	919	455
180	292	339	1155	514
200	292	300	1484	456

reduction effect of multi-hop transmission. Obviously, total hop counts in EETP are the smallest and total hop counts in EAPC is the largest in the four algorithms. In EETP, the second objective function is to minimize the standard deviation of child nodes among RNs, which makes the RNs distribution more uniform than that in EAPC and CB. In CB, its objective function is to minimize the total hop counts, but the number of RNs is fewer. In the MOPSO-based algorithm, the second objective function is to minimize the maximum total hop counts of RNs, so the reduction of multi-hop transmission is satisfactory. However, the RNs load in EETP is

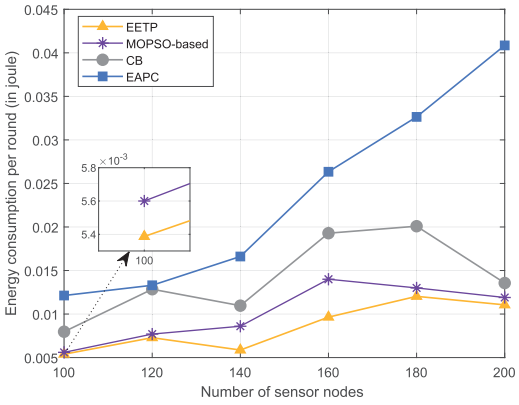

FIGURE 21. Network Energy Consumption ($R = 35$m).

more balanced and its total hop counts is less than that in the MOPSO-based algorithm.

E. ENERGY CONSUMPTION

Figure 21 compares network energy consumption per round of the four algorithms under different n. It is clear that energy consumption in EETP is the lowest between the four algorithms. This is because there are more load-balanced RNs reducing energy consumption caused by multi-hop transmission. In addition, the objection function in CB is to minimum hop counts, so energy consumption in CB is less than that in EAPC. The MOPSO-based algorithm also effectively reduces the energy consumption of sensor nodes, but the load of RNs is less balanced than that in EETP, resulting in its energy consumption more than that in EETP.

VI. CONCLUSION AND FUTURE WORK

This paper presents an energy-efficient trajectory planning algorithm (EETP) based on MOPSO. EETP consists of two phases: selection of potential visiting points and trajectory planning. In the selection of potential visiting points, representative visiting points can be selected within communication overlapping ranges of sensor nodes. In the trajectory planning, set of the optimal visiting points is found and the trajectory of the mobile sink is planned by MOPSO. Through EETP, we can obtain an optimal trade-off between the trajectory length of the mobile sink, the load balance of RNs and the number of RNs, which can reduce the delay in data delivery and prolong the network lifetime. Performance evaluation shows that the proposed EETP outperforms EAPC, CB and the MOPSO-based algorithm in terms of network lifetime, delay in data delivery and energy consumption.

In future work, we will focus on the following two aspects: 1) in low-buffered networks, how to hold the data is a meaningful problem. We will apply multiple sinks to low-buffered networks to increase the frequency of RNs being visited by mobile sinks. However, this will arise some problems, such as the cooperative mechanism of multiple sinks, the selection of RNs. These problems will be carefully explored in future work; 2) to increase the adaptability of the algorithm, we will study the proposed algorithm in different topologies, such as grid topology.

REFERENCES

[1] S. A. Malek, S. D. Glaser, and R. C. Bales, "Wireless sensor networks for improved snow water equivalent and runoff estimates," *IEEE Access*, vol. 7, pp. 18420–18436, 2019.
[2] J. Zhang, X. Feng, and Z. Liu, "A grid-based clustering algorithm via load analysis for industrial Internet of things," *IEEE Access*, vol. 6, pp. 13117–13128, 2018.
[3] T. Adame, A. Bel, A. Carreras, J. Melià-Seguí, M. Oliver, and R. Pous, "Cuidats: An RFID–WSN hybrid monitoring system for smart health care environments," *Future Gen. Comput. Syst.*, vol. 78, no. 5, pp. 602–615, Jan. 2018.
[4] X. Fu, G. Fortino, W. Li, P. Pace, and Y. Yang, "WSNs-assisted opportunistic network for low-latency message forwarding in sparse settings," *Future Gener. Comput. Syst.*, vol. 91, pp. 223–237, Feb. 2018.
[5] X. He, X. Fu, Y. Yang, and J. Zhang, "Mobile Sink path selection and routing optimization algorithm for network energy balance," in *Proc. Int. Symp. Sens. Instrum. IoT Era (ISSI)*, Sep. 2018, pp. 1–5.
[6] X. Fu, H. Yao, and Y. Yang, "Modeling and analyzing cascading dynamics of the clustered wireless sensor network," *Reliab. Eng. Syst. Saf.*, vol. 186, no. 22, pp. 1–10, Jun. 2019.
[7] V. Kumar, A. Kumar, and M. Singh, "Improving network lifetime & reporting delay in wireless sensor networks using multiple mobile sinks," in *Proc. 3rd Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Mar. 2016, pp. 1675–1678.
[8] X. Fu, G. Fortino, P. Pace, G. Aloi, and W. Li, "Environment-fusion multipath routing protocol for wireless sensor networks," *Inf. Fusion*, to be published.
[9] J. Rao and S. Biswas, "Analyzing multi-hop routing feasibility for sensor data harvesting using mobile sinks," *J. Parallel Distrib. Comput.*, vol. 72, no. 6, pp. 764–777, 2012.
[10] M. Krishnan, S. Yun, and Y. M. Jung, "Enhanced clustering and ACO-based multiple mobile sinks for efficiency improvement of wireless sensor networks," *Comput. Netw.*, to be published.
[11] C. Tunca, S. Isik, M. Y. Donmez, and C. Ersoy, "Distributed mobile sink routing for wireless sensor networks: A survey," *IEEE Commun. Surveys Tutr.*, vol. 16, no. 2, pp. 877–897, 2nd Quart., 2014.
[12] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Netw.*, vol. 1, nos. 2–3, pp. 215–233, 2003.
[13] J. Tang, H. Huang, S. Guo, and Y. Yang, "Dellat: Delivery latency minimization in wireless sensor networks with mobile sink," *J. Parallel Distrib. Comput.*, vol. 83, pp. 133–142, Sep. 2015.
[14] C. Zhu, S. Wu, G. Han, L. Shu, and H. Wu, "A tree-cluster-based data-gathering algorithm for industrial WSNs with a mobile sink," *IEEE Access*, vol. 3, pp. 381–396, 2015.
[15] T. A. Al-Janabi and H. S. Al-Raweshidy, "A centralized routing protocol with a scheduled mobile sink-based AI for large scale I-IoT," *IEEE Sensors J.*, vol. 18, no. 24, pp. 10248–10261, Dec. 2018.
[16] N. Xia, C. Wang, Y. Yu, H. Du, C. Xu, and J. Zheng, "A path forming method for water surface mobile sink using Voronoi diagram and dominating set," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7608–7619, Aug. 2018.
[17] K. Almi'Ani, A. Viglas, and L. Libman, "Energy-efficient data gathering with tour length-constrained mobile elements in wireless sensor networks," in *Proc. 35th IEEE Local Comput. Netw. Conf.*, Oct. 2010, pp. 582–589.
[18] H. Salarian, K.-W. Chin, and F. Naghdy, "An energy-efficient mobile-sink path selection strategy for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2407–2419, Jun. 2014.
[19] W. Wen, S. Zhao, C. Shang, C.-Y. Chang, "EAPC: Energy-aware path construction for data collection using mobile sink in wireless sensor networks," *IEEE Sensors J.*, vol. 18, no. 2, pp. 890–901, Jan. 2018.
[20] H. Mofid, H. Jazayeri-Rad, M. Shahbazian, and A. Fetanat, "Enhancing the performance of a parallel nitrogen expansion liquefaction process (NELP) using the multi-objective particle swarm optimization (MOPSO) algorithm," *Energy*, vol. 172, pp. 286–303, Jan. 2019.
[21] N. Al Moubayed, A. Petrovski, and J. McCall, "D2MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces," *Evol. Comput.*, vol. 22, no. 1, pp. 47–77, Mar. 2014.
[22] R. Zhu, Y. Qin, and J. Wang, "Energy-aware distributed intelligent data gathering algorithm in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 7, no. 1, 2011, Art. no. 235724.

[23] A. Kaswan, V. Singh, and P. K. Jana, "A novel multi-objective particle swarm optimization based energy efficient path design for mobile sink in wireless sensor networks," *Pervasive Mob. Comput.*, vol. 46, pp. 122–136, Jun. 2018.
[24] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.
[25] R. Wang, S. Lai, G. Wu, L. Xing, L. Wang, and H. Ishibuchi, "Multi-clustering via evolutionary multi-objective optimization," *Inf. Sci.*, vol. 450, pp. 128–140, Jun. 2018.
[26] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Struct. Multidisciplinary Optim.*, vol. 26, no. 6, pp. 369–395, Apr. 2004.
[27] S. Sedarous, S. M. El-Gokhy, and E. Sallam, "Multi-swarm multi-objective optimization based on a hybrid strategy," *Alexandria Eng. J.*, vol. 57, no. 3, pp. 1619–1629, 2018.
[28] D. Vázquez, M. J. Fernández-Torres, R. Ruiz-Femenia, L. Jiménez, and J. A. Caballero, "MILP method for objective reduction in multi-objective optimization," *Comput. Chem. Eng.*, vol. 108, pp. 382–394, Jan. 2018.
[29] Y. H. Lin, L. C. Huang, S. Y. Chen, and C. M. Yu, "The optimal route planning for inspection task of autonomous underwater vehicle composed of MOPSO-based dynamic routing algorithm in currents," *Appl. Ocean. Res.*, vol. 75, pp. 178–192, Jun. 2018.
[30] W. Zhao, Z. Luan, and C. Wang, "Parameter optimization design of vehicle E-HHPS system based on an improved MOPSO algorithm," *Adv. Eng. Softw.*, vol. 123, pp. 51–61, Sep. 2018.
[31] M. Z. Bin Mohd Zain, J. Kanesan, J. H. Chuah, S. Dhanapal, and G. Kendall, "A multi-objective particle swarm optimization algorithm based on dynamic boundary search for constrained optimization," *Appl. Soft Comput.*, vol. 70, pp. 680–700, Sep. 2018.
[32] J. Knowles and D. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, Jan. 2000.
[33] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[34] M. Clerc, "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization," in *Proc. Congr. Evol. Comput.*, Washington, DC, USA, vol. 3, Jul. 1999, pp. 1951–1957.
[35] A. W. Mohemmed, N. C. Sahoo, and T. K. Geok, "Solving shortest path problem using particle swarm optimization," *Appl. Soft Comput.*, vol. 8, no. 4, pp. 1643–1653, 2008.
[36] Y. Zhang, D. Gong, Y. Ren, and J. Zhang, "Barebones multi-objective particle swarm optimizer for constrained optimization problems," *ACTA Electronica Sinica*, vol. 39, no. 34, pp. 1436–1440, Dec. 2011.

XIAOLIN HE received the bachelor's degree in automation from the Sichuan University of Science and Engineering, China, in 2017. He is currently pursuing the master's degree in control theory and control engineering with the Shanghai Maritime University. From 2015 to 2016, he was with the East China University of Science and Technology as an Exchange Student. His research interests include wireless sensor networks, the Internet of Things, and machine learning.

XIUWEN FU received the B.S. degree in mechanical engineering from the Henan University of Science and Technology, China, in 2009, and the M.S. and Ph.D. degrees in mechanical engineering from the Wuhan University of Technology, China, in 2012 and 2016, respectively. He is currently serving as a Lecturer with Shanghai Maritime University. His research interests include the Internet of Things, wireless sensor networks, and complex networks.

YONGSHENG YANG received the Ph.D. degree in mechanical engineering from the Nanjing University of Aeronautics and Astronautics, China, in 1998. He is currently a Professor with Shanghai Maritime University. His research interests include modeling of port intelligent scheduling and the Internet of Things.