End-to-end data collection strategy using mobile sink in wireless sensor networks

Xiaofeng Wu¹, Zhuangqi Chen², Yi Zhong³, Hui Zhu⁴ and Pingjian Zhang²

Abstract
Several data collection algorithms, which are based on the combination of using mobile sinks and multiple-hop forwarding, have been proposed to prolong the network lifetime of wireless sensor networks. However, most approaches treat the collection point selection and touring path planning as two independent problems, which leads to a sub-optimal solution for data collection. This article proposed an ant colony optimization based end-to-end data collection strategy to perform the collection point selection and the touring path planning simultaneously. The proposed algorithm first constructs a data-forwarding tree, and then heuristically selects collection points and plans a touring path at the same time. The performance evaluation shows that the end-to-end strategy can improve the network lifetime of wireless sensor network compared to other approaches, especially in the unbalanced distribution scenario of sensors. The end-to-end strategy is also capable of being integrated with other methods.

Keywords
Data collection, wireless sensor network, mobile sink, network lifetime, end-to-end

Date received: 31 July 2021; accepted: 12 January 2022
Handling Editor: Miguel Acevedo

Introduction
Wireless sensor network (WSN) is a distributed network for sensing the environment. Because of the flexibility, WSNs have been widely used in several applications, such as smart homes,¹ medical care,² environmental monitoring,³ disaster prediction,⁴ and military field⁵. A WSN consists of a set of sensor nodes deployed in a certain area. Each sensor has limited battery capacity and data buffer of a fixed size, which leads to the issue of energy efficiency. Thus, several protocols are used to achieve the energy efficiency in WSN. Those protocols are mainly categorized into four classes: ratio module, data reduction, sleep/awake schemes, and battery charging.⁶ The ratio module considers four states for each sensor node: idle, sleep, transmit, and receive. Most of the time, sensors are in idle state which causes depletion of the sensor node. The ratio module minimizes the energy consumption by modulation of optimization parameters. However, the ratio module has to manage four states which introduce additional energy consumption. The next category, data reduction, minimizes the quantity of transmitted data packets to achieve energy efficiency.

¹Public Courses Department, Guangzhou Polytechnic of Sports, Guangzhou, China
²School of Software Engineering, South China University of Technology, Guangzhou, China
³Guangzhou Center for Educational Technology, Guangzhou, China
⁴School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China

Corresponding author:
Pingjian Zhang, School of Software Engineering, South China University of Technology, Guangzhou 510641, China.
Email: pjzhang@scut.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
and reduce the rate of traffic and latency, too. Nevertheless, it inevitably losses the accuracy and precision of original data. The sleep/awake scheme employs the duty cycling mechanism to diminish the idle state and support the sleep mode. However, the duty cycling leads to sleepy latency and wastes energy for redundant wakeups. The battery charging harvests energy from the wind, solar, and other new energy. It takes nodes with the uneven residual energy into consideration for designing the protocol.

In this article, we focus on the energy efficiency based on the sleep/awake scheme. The sensor nodes periodically generate sensing data and transfer data to the base station through multiple-hop forwarding. Due to the nature of multiple-hop forwarding, the nodes near the base station will receive and transfer more data packets so that those nodes will consume more energy. In this case, the nodes closest to the base station are prone to deplete the energy first. When those nodes are dead, the data of other nodes cannot arrive at the base station, and the network cannot work until the dead nodes are charged. The problem of unbalanced energy consumption will significantly shorten the lifetime of the network.

Recently, to alleviate the problem of non-uniform energy consumption, several studies have considered a novel data collection strategy with a mobile sink. The mobile sink plays the role of a data transporter that moves around the WSN and collects the data packets from sensor nodes. There are two major categories of those studies: no-data-forwarding and hybrid approaches. The no-data-forwarding methods apply the mobile sink to visit all sensor nodes to gather data packets, where no communication between sensor nodes is needed and the energy consumption of whole network will significantly reduce. But the touring path of mobile sink is too long, which leads to longer delay of network. The hybrid approaches use some data collection points (CPs) to communicate with some part of sensor nodes and gather their data packets. The mobile sink only visits the CPs to gather data packets. The key is to tradeoff the delay of data collection and energy consumption of communication between nodes. Most of those methods select CPs and plan a touring path for the mobile sink to visit all selected CPs separately.

However, the two problems of CPs selection and path planning interrelate with each other. The selected CPs determine a corresponding touring path while maximum touring length influences the process of selecting CPs. Independently solving the two problems will result in the low utilization of the given maximum touring length, and the full use of maximum length can effectively prolong the network lifetime. As shown in Figure 1, due to the separation of CPs selection and path planning and the local optimization of each stage, it is not uncommon to obtain a result with lower path utility.

To further prolong the network lifetime, we investigate whether we can solve both CPs selection and path planning simultaneously. The following summarize the contributions of this article:

- We proposed an end-to-end data collection strategy, which selects CPs and plans the touring path heuristically and simultaneously. We first construct a data-forwarding tree (DF-Tree) and then apply an ant colony optimal (ACO)-based algorithm to select CPs in order.
- We introduced a path estimation method modified from the path construction of EAPC to improve the ACO-based algorithm. Instead of the path found by the best ant, a re-estimated shortest path is used to update the pheromone of ACO.

Figure 1. The relation between CPs selection and path planning: (a) independently solving the CPs selection and path planning and (b) fully use of maximum length.
The remaining parts of this article are structured as follows. We review the related works in section “Related works,” formulate the problem in section “Problem formulation,” detail the proposed end-to-end method in section “End-to-end data collection strategy,” show the performance of proposed method in section “Performance evaluation,” and conclude this article in section “Conclusion and future works.”

Related works
In the past few years, several energy-efficient routing protocols for WSN have been proposed. In Nabavi et al.,9 the multi-objective greedy method is designed to find the optimal routing path, where each node selects another node as next hop based on several factors for transmitting information. To balance the energy consumption among all sensor nodes and further prolong the network lifetime, most approaches10–14 employ a hierarchical routing protocol, where all nodes are first divided into several clusters and then a cluster head is elected to maintain other nodes in each cluster. Although those routing protocols help to reduce the energy consumption, the performance still remains inadequate. In those methods, the nodes closed to the base station will deplete energy more quickly and become the bottleneck of the whole network, which lowers the upper bound of the performance. To alleviate this issue, several mobile sink-based data collection protocols have been proposed, which adopt the mobile sink to gather data packets.

In those methods, a number of data CPs are selected and responsible for receiving data packets from other sensor nodes. The mobile sink moves periodically within the WSN to visit CPs and gather all data packets. According to whether or not the mobile sink passes through the center of the CPs, those methods can be classified into two mainly categories: stop and collect (SC) and collect while moving (CWM).

SC
The SC policy requires that the mobile sink should stop at the location of the sensor for communication. The mobile sink will pass through the center of sensors to collect data. Those methods can be further divided into two classes: no-data-forwarding and hybrid methods.

No-data-forwarding. In the class of no-data-forwarding, most methods have adopted one or more mobile sinks to visit all sensors, which can avoid the multiple-hop communication between sensor nodes. In Ma and colleagues,15,16 the problem of data collection is formulated as the single-hop data gathering problem (SHDGP), which aims to find the shortest touring path for the mobile sink to visits all sensors. A heuristic tour-planning algorithm for a single mobile sink is proposed to solve the SHDGP. However, the speed of moving the mobile sink is much lower than that of data-forwarding through the network, which will result in a higher latency of data update. In most applications of the WSN, the delay is an important performance metric. A WSN with low latency should be designed. One possible solution is to apply multiple mobile sinks to collect data together. Some studies17–19 have proposed several data collection strategies with multiple mobile sinks. Although those studies reduce the latency of data collection, the high cost of mobile sink makes them impractical. To alleviate the problem of the high latency and the high cost caused by mobile sink, some hybrid methods are proposed.

Hybrid. Those hybrid methods20–26 combines multiple-hop forwarding with the use of mobile sink, where some nodes are selected as CPs. All sensor nodes send their data to the nearest CP through multiple-hop. And the mobile sink periodically visits all CPs to collect data packets through a determined touring path constrained by delay. Since the mobile sink has a fixed moving speed, the time delay constraint can be transformed into a maximum length constraint of the touring path of the mobile sink. Most hybrid methods transform the problem of data collection into the problem of CPs selection and path planning, whose objective is to minimize the energy consumption under a maximum touring length constraint. They first cluster all sensor nodes into several classes, and select a cluster head as the CP. Then, the path planning is solved as the travel salesman problem (TSP) using some heuristic algorithms. In Krishnan and Lim,27 the cluster heads are chosen dynamically based on the remaining energy and then a model free Q-learning approach is introduced to find the shortest routing path. In those methods, selecting CPs and planning the tour are treated as two independent problems, which lead to the sub-optimal solution for the data collection strategy.

CWM
More recently, a new data collection protocol has been proposed, where the mobile can communicate with a sensor when it passes through the communication range of the sensor. The study by Wen et al.8 selects CPs accounting for the path cost from current CP to next point and the forwarding load of each sensor, and then constructs a data collection path. The data can be collected by mobile sink if the path passes through the communication range of these sensors. The study by Chang et al.28 dynamically adopts the proper transmission rate based on the data amount of each sensor to
construct the shortest and energy-efficient touring path but guarantees the completeness of data collection. In Dona et al.,29 an extended ant colony optimization (ACO)-based method is first designed to select the best set of rendezvous points and the efficient touring path of mobile sink and then a virtual rendezvous points selection is performed to increase the data gathering speed. Although the protocol can reduce the constructed path length, the design of the protocol is more complicated. Sensors should have the ability to sense the location of the mobile sink and the duration that the mobile sink locates in the communication range should be enough to transfer all data packets.

In this article, we focus on the SC policy and propose a heuristic end-to-end data collection strategy based on ACO, where CPs selection and tour planning perform at the same time. The proposed ACO-based algorithm can be extended to combining other methods which solve the CP selection and path planning independently, to fully use the maximum path length.

Problem formulation

In this article, we focus on a WSN in which sensor nodes periodically generate data packets. All the data packets must be transmitted to the base station within a given delay. There are a set of CPs that responsible for collecting and storing the data packets generated by a certain group of sensor nodes. And a mobile sink can move around the WSN to collect data from the CPs. The objective is to select the set of CPs and plan a tour that passes through all CPs for mobile sink within the given delay. We first detail the problem description, outline some assumptions, and then formulate the problem definition.

Problem description

A WSN consists of a set of sensor nodes, a communication topology network, and a base station. Each node is responsible for sensing environmental status and generating data packets periodically. Those data packets must be transmitted to the base station within a delay $T$. In order to decrease the energy consumption and prolong the lifetime of WSN, a mobile sink is used to collect data packets. Because the mobile sink has a fixed moving speed, the traveling time is proportional to the touring length. Given the delay $T$, there is a length $L_{\text{max}}$ corresponding to it where $L_{\text{max}}$ denotes the maximum touring length. To satisfy the length constrain, all sensor nodes are divided into several groups. Each group selects a node as the CP, which is used to gather data packets from other nodes and upload those packets to the mobile sink. Other nodes will transmit their data packets to the CP through multiple-hop. The mobile sink visits all CPs and collects all data packets in a round.

Transmitting and receiving one data packet will consume several energies. The energy consumption is proportional to the packet size. In this article, we assume the sensor nodes generate one data packet with fixed size in a round. We can formulate the energy consumption of sensor $i$ in each round as following

$$E_i = E_t \times (n_{d_i} + 1) + E_r \times n_{d_i}$$

(1)

where $E_t$ and $E_r$ denote the energy consumption of transmitting and receiving one packet, respectively, and $n_{d_i}$ is the number of packets sent to sensor $i$. The lifetime of WSN is dependent on the node which first runs out of energy. It is obviously that the CPs consume the most energy. To prolong the lifetime of WSN, we should minimize the energy consumption of the CP which has maximum energy consumption. The objective is defined as following

$$\min \max_{1 \leq i \leq m} E_i$$

(2)

where $m$ is the number of CPs. From equation (1), we can find that $E_t \propto n_{d_i}$. For the CP $i$, the $(n_{d_i} + 1)$ is equal to the number of nodes in the group containing CP $i$. Minimizing the energy consumption of a CP is equivalent to minimizing the size of the group containing the CP.

Assumption

We make some assumptions as following:

- The mobile sink moves at a fixed speed.
- The mobile sink collects data packets from a CP only when it arrives at the position of the CP.
- Comparing to the touring time of mobile sink, the communication time between the mobile sink and a CP is small enough to be neglected. Similarly, the time of transmission and reception between sensor nodes is negligible compared to the touring time.
- Each CP has enough capacity to store all sensed data.
- The sensor nodes have a fixed communication range and initial energy.
- There are no isolated nodes, and each sensor node can communicate with other nodes with multiple-hop.
- Each sensor node produces on data packet each round.
- The energy of mobile sink is sufficient to visit all CPs and gather data packets in a round within the given delay.
**Problem definition**

The topology of WSN can be formulated as an undirected graph $G = \langle V, E \rangle$, where $V$ is the set of sensor nodes in WSN and $E$ is the set of edges between two sensor nodes where $(v_i, v_j) \in E$ represents sensor $i$ and sensor $j$ are within each other’s communication range. For ease of description, let $v_0 \in V$ denote the base station. A solution of the problem is a tour $T = \langle v_0, v_1, ..., v_m, v_m + 1 = v_0 \rangle$ that starts and ends at $v_0$, where $v_i$ denotes the selected CP $i$ and the index $i$ denotes the visited order by the mobile sink. The problem can be formulated as an optimal problem

\[
\text{Minimize } \max_{1 \leq i \leq m} nd_i
\]

\[
\text{s.t. } \sum_{i=0}^{m} \text{dist}(v_i, v_{i+1}) \leq L_{\text{max}}
\]

where $\text{dist}(v_i, v_{i+1})$ is the Euclidean distance, and $v_j \in V$.

**End-to-end data collection strategy**

This section details our proposed algorithm for solving the problem described in section “Problem formulation.” The proposed algorithm is based on the key observation that selecting CPs and planning a tour are not independent. The proposed data collection strategy constructs a DF-Tree first and then uses the ACO method and describes how the method can be extended.

**Initial phase**

Given a graph $G = \langle V, E \rangle$ denoting the communication topology of WSN, we first construct a DF-Tree. The DF-Tree is rooted by the base station. All nodes in the tree can only send their data packets to their parent nodes. When a node is selected as the CP, there are several changes to the tree structure. First, the edge between the CP and its parent is broken. That means the CP does not need to send packets to its parent but directly upload to the mobile sink. In addition, a new tree rooted by the CP is generated. In this case, an isolated tree can be viewed as a group. All nodes forward the data to the CP through their parents in their own groups. In this case, the $nd_i$ mentioned above denotes the number of nodes in the sub-tree rooted by $node_i$.

Motivated by Wen et al., we construct a minimum spanning tree as the DF-Tree which has the minimum total forwarding cost. The Prim algorithm is applied to construct the DF-Tree.

In addition, a complete graph $G' = \langle V, E' \rangle$ is constructed based on the original input graph $G$, where $E'$ denotes the possible movements of the mobile sink.

**End-to-end joint optimization framework**

Based on the constructed DF-Tree, we aim to select CPs and plan a path. The essence of path planning is to determine the order of the CPs. In fact, the process of selecting CPs also can be viewed as a kind of access order of CPs. Based on this insight, we propose an end-to-end framework to jointly optimize the two problems using improved ACO methods.

**ACO for DC.** ACO is a heuristic algorithm, which has been proved to be effective for solving combinatorial optimization problem. The basic idea of ACO is motivated by practical ant colony system. When an ant walks on the road, it will release some pheromone. The ants will prefer to move toward the direction which contains more pheromone. While the path with the better solution always contains more pheromone. In the end, most ants will likely choose the best path which has the most pheromone. The key of ACO is how to set up and update the pheromone. The goal of the problem is to minimize $\max nd_i$. A better solution of the problem means a smaller $\max nd_i$. Based on this point, we propose an ACO-based algorithm to solve CPs selection and tour planning simultaneously.

We iteratively update the pheromone using the best result of the elitist ant at each iteration. In each iteration, several ants search the feasible solutions based on a probability function independently. And then, the best solution with minimum $\max nd_i$ is used to update the pheromone. The probability function is defined as following

\[
P_{ij} = \frac{\sum_{u \in J(i)} \tau_{iu}^{2} \eta_{ij}^{2}}{\sum_{u \in J(i)} \tau_{iu}^{2} \eta_{iu}^{2}}
\]

where $P_{ij}$ denotes the probability of moving from $node_i$ to $node_j$ for an ant, $J(i)$ is the set of neighbor nodes of $node_i$ in the graph $G'$ but have not been visited for the ant, $\tau_{ij}$ is the amount of pheromone, and $\eta_{ij}$ is the heuristic value which is defined as following

\[
\eta_{ij} = \frac{w_i}{\text{dist}(i,j)}
\]

where $\text{dist}(i,j)$ is the Euclidean distance, and $w_i$ denotes the gain value of selecting $node_j$ as the CP which is defined as following

\[
w_j = nd_j \times H(j, \text{root}_j)
\]
where \( H(j, i) \) is the hop distance from \( node_j \) to \( node_i \) in the DF-Tree, and \( root_j \) is the root of the sub-tree containing \( node_j \). When \( node_j \) is selected as CP, \( node_j \) does not need to send data packets to its parent. And all nodes in the path from \( node_j \) to the root will save energy consumption.

Let \( T = \{ v_0, v_1, v_2, \ldots, v_m \} \) denotes the best solution after \( k \)th iteration. The elements in the \( T \) present the CPs and the order corresponds to the touring order of the mobile sink. The following rule is used to update the pheromone

\[
\tau_{k+1}(v_i, v_j) = \tau_k(v_i, v_j) + Q \times \Delta \tau \tag{8}
\]

\[
\Delta \tau = \frac{1}{M_{nd}} + F \times \sigma \tag{9}
\]

\[
M_{nd} = \max_{0 \leq i < m} nd_i \tag{10}
\]

\[
F = \left( \frac{\sum_{i=0}^{m} nd_i^2}{(m + 1) \sum_{i=0}^{m} nd_i} \right)^2 \tag{11}
\]

where \( \rho \) is the pheromone evaporation rate, and \( Q \) and \( \sigma \) are the tunable controlling scalar. \( \Delta \tau \) denote the increment of the pheromone, \( M_{nd} \) is the maximum of all \( nd_i \), and \( F \) is the fairness index (FI), which describes the degree of load balance for all CPs.

The proposed ACO-based algorithm details in Algorithm 1.

**Path estimation.** Because of the randomness of ACO-based algorithm, it is easy to get a path in which there are several intersecting segments. With the same CPs, a touring path with intersecting segments means a longer touring length. The problem of intersecting segments also leads to the algorithm get a sub-optimal solution. To alleviate the problem, we utilize an efficient path estimation algorithm to compute the touring length \( L_T \) and refine the touring path. In Algorithm 1, a minimum touring length is estimated instead of directly computing the length based on \( T \) at step 24. At step 29, the \( T_{best} \) is converted to a polygon to avoid the segment intersection. The path planning problem of the CPs is a traveler salesman problem that has been proved to be a NP-hard problem. Motivated by Wen et al.,\(^8\) we introduce a convex polygon-based path estimation algorithm. The algorithm is similar to the path construction proposed in Wen et al.\(^8\) The following briefly presents the method.

The algorithm consists of three steps: the construction of the convex polygon, the connection of internal points, and the estimation of touring length.

### Algorithm 1. ACO-based CPs selection and tour planning.

**Input:** DF-Tree, the complete graph \( G = (V, E) \);

**Output:** a sequence of ordered collection points \( CP_{s} = \{p_0, p_1, \ldots, p_m, p_{m+1}\} \);

1: \( r_0 \leftarrow 1 \);
2: \( BEST_{CPs} \leftarrow \{\} \);
3: for \( epoch = 0 \rightarrow MAX\text{Iteration} \) do
4: \hfill initialize \( N \) ants;
5: \hfill \( T_{all} \leftarrow \{\} \);
6: \hfill \( \text{for all ants do} \)
7: \hfill \( i \leftarrow 0 \);
8: \hfill \( T \leftarrow \{v_0\} \);
9: \hfill \( \text{repeat} \)
10: \hfill \( \text{pro} \leftarrow \text{random}(); \)
11: \hfill \( \text{for } j = 0 \rightarrow |V| \) do
12: \hfill \( \text{if } v_j \text{ in } T \text{ then} \)
13: \hfill \( p_j \leftarrow 0 \);
14: \hfill \( \text{else} \)
15: \hfill \( \text{evaluate } p_j \text{ according to equation (5);} \)
16: \hfill \( \text{end if} \)
17: \hfill \( \text{pro} \leftarrow \text{pro} - p_v; \)
18: \hfill \( \text{if } \text{pro} \leq 0 \text{ then} \)
19: \hfill \( i \leftarrow j; \)
20: \hfill \( T \leftarrow T_{all} \append(v_j); \)
21: \hfill \( \text{break;} \)
22: \hfill \( \text{end if} \)
23: \hfill \( \text{end for} \)
24: \hfill \( \text{evaluate the touring length } L_T \text{ of } T; \)
25: \hfill \( \text{until } L_T > L_{\text{max}} \)
26: \hfill \( T_{all} \leftarrow T_{all} \cup \{T\}; \)
27: \hfill \( \text{end for} \)
28: \hfill \( T_{best} \leftarrow \arg \min \max_{nd}(T); \)
29: \hfill \( T_{best} \leftarrow to\text{Polygon}(T_{best}); \)
30: \hfill \( \text{use } T_{best} \text{ to update } \tau_j \text{ according to equation (8);} \)
31: \hfill \( CP_{s_{best}} \leftarrow BEST_{CPs} \cup \{T_{best}\}; \)
32: \hfill \( \text{end for} \)
33: \hfill \( CP_{s} \leftarrow \arg \min \max_{nd}(T); \)
34: \hfill \( \text{return } CP_{s}; \)

### Construction of the convex polygon.

This step aims to construct a bounding box of all CPs where the box is a convex polygon and has the minimum area. First, the point at the bottom-left is selected as the start point \( p_0 \). Let \( l \) be a horizontal line passing through \( p_0 \). Then, we take the \( p_0 \) as the center and turn \( l \) in an anticlockwise direction until the line touches any other CP, which will be labeled with \( p_1 \). Then the touched point serves as the new center and the turning operation is repeatedly executed. The touched points will be labeled orderly with \( \{p_1, p_2, p_3, \ldots, p_m\} \). The operation will be repeated until the touched point is the start point \( p_0 \). At the end, a convex polygon \( \{p_0, p_1, p_2, \ldots, p_m, p_0\} \) is constructed where \( (k + 1) \) denotes the number of vertexes of the convex polygon. Note that there is something different about the selection of the start point between our method and Wen et al.\(^8\) The start point is the base station in Wen et al.\(^8\) while it is the point at the bottom-left in our method.
**Connection of the internal points.** After the convex polygon is constructed, there may remain some unconnected points. Obviously, those points are all located at the inside of the convex polygon. We iteratively joint the remaining points into the polygon. Let $P_{CP}$ and $P_R$ denote the set of vertexes in the convex polygon and the set of the remaining points, respectively. At each iteration, a point $V \in P_R$ with the least selection cost will be selected and inserted into the polygon. The selection cost of a point $V$ is defined as following

$$\text{cost}_V = \min_{P_i \in P_{CP}} \text{dist}_{\text{incre}}(V, P_i, P_i + 1)$$

(12)

$$\text{dist}_{\text{incre}}(V, P_i, P_i + 1) = \text{dist}(V, P_i) + \text{dist}(P_i, P_i + 1)$$

(13)

The cost$_V$ evaluates the increment of length after inserting point $V$ into the polygon which is proposed in Wen et al.\(^8\) The position to insert the selected point $V$ is at pos, where

$$\text{pos} = \left( \arg \min_i \text{dist}_{\text{incre}}(V, P_i, P_i + 1) \right) + 1$$

(14)

where $P_i \in P_{CP}$. After the point $V$ is inserted into the polygon, there are several changes in the polygon. The $V'$ will be labeled with $P_{\text{pos}}$, and the index of the label of the original points $\{P_{\text{pos}}, P_{\text{pos} + 1}, \ldots\}$ will plus one, that is, their new labels are $\{P_{\text{pos} + 1}, P_{\text{pos} + 2}, \ldots\}$ successively. In addition, the point $V'$ will be removed from $P_R$. Then, the above process will be repeatedly executed until the $P_R$ is empty.

**Estimation of touring length.** When $P_R$ is empty, the $P_{CP}$ presents a touring path which is a polygon. We use the total length of the polygon to estimate the minimum length of the touring path. Note that the estimated length and the polygon are only applied at Step 24 and Step 29, respectively, in Algorithm 1.

**Complexity analysis**

The proposed data collection strategy consists of an initial phase and an ACO-based CPs selection and tour planning. In the initial phase, the Prim algorithm is employed to construct the DF-Tree, whose time complexity is $O(n^2)$ where $n$ denotes the number of sensors. In the ACO-based CPs selection and tour-planning stage, which is described in Algorithm 1, the time complexity of Steps 12–23 is $O(n^2)$ and the complexity of path estimation at Step 24 is $O(m^2)$ where $m$ denotes the number of CPs and is smaller than $n$. As a result, the time complexity of each ant is $O(m \times (O(n^2) + O(m^2))) = O(mn^2)$. Given $t_1$ iterations and $t_2$ ants in each iteration, the overall time complexity of the proposed method is $O(n^2) + O(mn^2) \times t_1 \times t_2 = O(t_1t_2mn^2)$.

**Combining with other methods**

Due to the flexibility of the parameters setting of ACO, the proposed end-to-end strategy can be easily extended to other methods. Given a data collection algorithm $\Gamma$ that independently solves the CP selection and path planning, we first construct a heuristic value table $\eta$ to describe the process of selecting CPs. The $\eta_j$ denotes the probability of selecting node$_j$ as next CP when current CP is node$_i$ in $\Gamma$. Then, the $\eta_j$ is applied in equation (5). In this case, the Algorithm 1 can be viewed as a combination of ACO and $\Gamma$.

**Performance evaluation**

To validate the effectiveness of the proposed method, we have conducted extensive experiments. The Python simulator is used to simulate the network environment. We consider a network area that measures 300 m $\times$ 300 m. Several sensor nodes are randomly deployed over the network and can communicate with each other through multiple-hop. The communication range is 30 m. In general, we assume that each sensor node generates one packet per round of data collection, where the packet has a fixed size. And the consumed energy for transmitting and receiving one packet is 0.021 and 0.015 J, respectively. The initial battery energy of each sensor is 20 J. In addition, the moving speed of the mobile sink is 1 m/s. All the simulation settings are described in Table 1. The deployment of sensor nodes is dependent on its certain application. Different deployment characteristics will result in different performance evaluations. We conduct extensive experiment on two scenarios:

- Unbalanced deployment (UD) scenario: all sensor nodes are uniformly deployed in the network area.

| Table 1. The simulation settings. |
|-----------------------------------|
| **Description**                  | **Value**       |
| Node deployment                   | Random          |
| Network area                      | 300 $\times$ 300m |
| Communication range               | 30 m            |
| Transmission power                | 0.021 J         |
| Receiving power                   | 0.015 J         |
| Battery power                     | 20 J            |
| Moving speed of mobile sink       | 1 m/s           |
| The number of sensor nodes        | 10–300          |
| Simulator                         | Python          |
Balanced deployment (BD) scenario: the network area is divided into a $3 \times 3$ grid of squares, where each square measures 100 m$^2$. Then, we choose five grids and deploy the same number of sensors in each grid. Grids do not share any edges with each other.

To benchmark our method, we compare it to the version of “stop and collect” of Energy-Aware Path Construction (EAPC-SC) algorithm\textsuperscript{8} in terms of network lifetime, total energy consumption, FI, efficient index (EI), and running time. The EAPC-SC algorithm has more efficient performance than existing data collection mechanisms, which is applied to solve the same problem as the considered in this article. The EAPC-SC first constructs a minimal spanning tree (MST), selects a set of CPs, and then constructs a data collection path. The selection of CPs is based on a heuristic function, which is the same as the $h_{ij}$ in our proposed algorithm. In addition, the path estimation is a modified version of the path construction of the EAPC-SC. In somewhat, our proposed algorithm is an evolution of the EAPC-SC. Our proposed method handles the problem of sub-optimal solution caused by the independence between CPs selection and path planning. We compare with the EAPC-SC algorithm to verify the feasibility of the proposed method. Besides, we also compare our method with the dynamic directional routing (DDR) protocol,\textsuperscript{30} which adapt to the mobility of sensor nodes to optimize the routes toward the static sink. The DDR protocol is a state-of-the-art routing algorithm without mobile sink, which achieve a reliable and energy-efficient routing.

In the experiments, the distributions of sensors are randomly initialized and the proposed algorithm is based on the ACO algorithm which is stochastic. Therefore, we conducted 10 repeated experiments to alleviate the effect from randomness.

### Network lifetime

The network lifetime describes the maximum working time of the whole WSN, which is an important metric of a well-designed WSN. And it uses round as a unit of measure. The higher value denotes better performance. In each round, a mobile sink visits all CPs based on the arranged routes and collects all data packets from CPs. The lifetime is defined as the time until one node runs out of energy. And very simplistically, we assume the sensors consuming energy only for transmitting and receiving data packets. The lifetime of network depends on the number of total sensor nodes and the touring path of mobile sink. We compare the lifetime under various numbers of sensor nodes for two scenarios, where the number of sensors ranges from 10 to 300. The results are shown in Tables 2 and 3, where the

| #sensors | 50   | 75   | 100  | 125  | 150  | 175  | 200  | 225  | 250  | 275  | 300  |
|----------|------|------|------|------|------|------|------|------|------|------|------|
|          |      |      |      |      |      |      |      |      |      |      |      |
| L$_{max}$ = 600 |      |      |      |      |      |      |      |      |      |      |      |
| ACO      | 296.0 | 129.0 | 97.4 | 51.9 | 34.3 | 31.3 | 19.5 | 17.1 | 11.9 | 10.4 | 7.8  |
| EAPC-SC  | 214.6 | 91.5  | 65.8 | 40.9 | 24.5 | 19.7 | 15.0 | 11.2 | 8.6  | 7.6  | 5.3  |
| DDR      | 10.0  | 7.0   | 5.0  | 4.0  | 3.0  | 3.0  | 2.0  | 2.0  | 2.0  | 2.0  | 2.0  |
| L$_{max}$ = 800 |      |      |      |      |      |      |      |      |      |      |      |
| ACO      | 590.8 | 209.0 | 141.4 | 82.8 | 55.1 | 49.2 | 33.4 | 28.3 | 21.3 | 18.5 | 15.3 |
| EAPC-SC  | 577.3 | 141.4 | 109.7 | 64.5 | 41.8 | 40.3 | 25.2 | 21.7 | 14.9 | 13.1 | 11.3 |
| DDR      | 10.0  | 7.0   | 5.0  | 4.0  | 3.0  | 3.0  | 2.0  | 2.0  | 2.0  | 2.0  | 1.0  |

BD: balanced deployment; ACO: ant colony optimal; EAPC-SC: Energy-Aware Path Construction; DDR: dynamic directional routing.

| #sensors | 50   | 75   | 100  | 125  | 150  | 175  | 200  | 225  | 250  | 275  | 300  |
|----------|------|------|------|------|------|------|------|------|------|------|------|
|          |      |      |      |      |      |      |      |      |      |      |      |
| L$_{max}$ = 600 |      |      |      |      |      |      |      |      |      |      |      |
| ACO      | 228.5 | 109.4 | 71.3 | 45.6 | 29.4 | 23.6 | 20.4 | 16.8 | 11.2 | 10.4 | 8.5  |
| EAPC-SC  | 162.2 | 77.0  | 59.0 | 34.7 | 24.0 | 20.6 | 16.9 | 13.1 | 8.6  | 9.0  | 6.0  |
| DDR      | 10.0  | 7.0   | 5.0  | 4.0  | 3.5  | 3.1  | 2.5  | 2.4  | 2.1  | 2.1  | 2.1  |
| L$_{max}$ = 800 |      |      |      |      |      |      |      |      |      |      |      |
| ACO      | 350.0 | 185.0 | 134.6 | 97.1 | 51.2 | 46.6 | 27.4 | 26.5 | 16.3 | 14.9 | 12.9 |
| EAPC-SC  | 323.0 | 120.9 | 98.5 | 77.5 | 40.5 | 33.5 | 20.0 | 21.3 | 12.0 | 12.0 | 9.3  |
| DDR      | 10.0  | 7.0   | 5.0  | 4.0  | 3.4  | 3.0  | 2.4  | 2.3  | 2.1  | 2.0  | 2.0  |

UD: unbalanced deployment; ACO: ant colony optimal; EAPC-SC: Energy-Aware Path Construction; DDR: dynamic directional routing.

- Balanced deployment (BD) scenario: the network area is divided into a $3 \times 3$ grid of squares, where each square measures 100 m$^2$. Then, we choose five grids and deploy the same number of sensors in each grid. Grids do not share any edges with each other.

To benchmark our method, we compare it to the version of “stop and collect” of Energy-Aware Path Construction (EAPC-SC) algorithm\textsuperscript{8} in terms of network lifetime, total energy consumption, FI, efficient index (EI), and running time. The EAPC-SC algorithm has more efficient performance than existing data collection mechanisms, which is applied to solve the same problem as the considered in this article. The EAPC-SC first constructs a minimal spanning tree (MST), selects a set of CPs, and then constructs a data collection path. The selection of CPs is based on a heuristic function, which is the same as the $h_{ij}$ in our proposed algorithm. In addition, the path estimation is a modified version of the path construction of the EAPC-SC. In somewhat, our proposed algorithm is an evolution of the EAPC-SC. Our proposed method handles the problem of sub-optimal solution caused by the independence between CPs selection and path planning. We compare with the EAPC-SC algorithm to verify the feasibility of the proposed method. Besides, we also compare our method with the dynamic directional routing (DDR) protocol,\textsuperscript{30} which adapt to the mobility of sensor nodes to optimize the routes toward the static sink. The DDR protocol is a state-of-the-art routing algorithm without mobile sink, which achieve a reliable and energy-efficient routing.

In the experiments, the distributions of sensors are randomly initialized and the proposed algorithm is based on the ACO algorithm which is stochastic. Therefore, we conducted 10 repeated experiments to alleviate the effect from randomness.

**Network lifetime**

The network lifetime describes the maximum working time of the whole WSN, which is an important metric of a well-designed WSN. And it uses round as a unit of measure. The higher value denotes better performance. In each round, a mobile sink visits all CPs based on the arranged routes and collects all data packets from CPs. The lifetime is defined as the time until one node runs out of energy. And very simplistically, we assume the sensors consuming energy only for transmitting and receiving data packets. The lifetime of network depends on the number of total sensor nodes and the touring path of mobile sink. We compare the lifetime under various numbers of sensor nodes for two scenarios, where the number of sensors ranges from 10 to 300. The results are shown in Tables 2 and 3, where the
ACO denotes our proposed strategy. We conduct experiments in both scenarios under different time delay constraints. It is obvious that our proposed end-to-end strategy can further prolong the network lifetime. To better compare with other methods, we also visualize the results in Figure 2.

**Total energy consumption**

Total energy consumption is defined as the sum of the energy consumption of all sensors in the WSN per round. Figure 3 compares the total energy consumption of WSN for different algorithms under two scenarios. The result shows that our proposed method results in comparable energy consumption per round with other protocols.

**FI**

Since the mobile sink-based data collection strategy is introduced to alleviate the problem of non-uniform energy consumption, the degree of uniformity should be quantified. A higher uniformity of energy consumption will lead to lower total energy consumption and higher network lifetime. The FI reflects the degree of the balance between nodes, which is defined as following

\[
FI = \frac{\left(\sum_{i=1}^{n} E_i\right)^2}{n \times \sum_{i=1}^{n} E_i^2}
\]

where \(n\) is the number of sensor nodes and \(E_i\) is the energy consumption of sensor \(i\) per round. Figure 4 compares the FI of different methods, where reveals the proposed end-to-end strategy can also get a higher up-bound of the FI.

**EI**

The given delay constraint is correspondent to a maximum touring length for mobile sink. The full use of the maximum length can effectively improve the network lifetime and FI. The EI describes the utilization of the given maximum length, which is defined as following

\[
EI = \frac{L_{\text{act}}}{L_{\text{max}}} \times \frac{m}{N}
\]

where \(m\) is the number of CPs, \(N\) is the total number of sensors, \(L_{\text{act}}\) and \(L_{\text{max}}\) denote the actual touring length
of mobile sink and the maximum length, respectively. Figure 5 compares the EI of different methods. The result shows that our method leads to a comparable EI to EAPC. Since the DDR protocol uses the static sink, the actual touring path length is zero.

**Running time**

The running time of different methods with various numbers of nodes is compared in Figure 6. The time complexity of EAPC and DDR methods is of the same magnitude, and the running time of the two methods
drew in the figure is nearly overlapped. To better distinguish, we change the scale of the vertical axis to logarithmic. Our proposed method is more time-consuming and has minor responsiveness, which makes it hard to be applied in online situations. However, there are many applications running algorithms to generate data collection strategies offline, where the structure of WSN is almost invariant and the proposed method is performed only when the structure is changed. Thus, the running time is acceptable in the offline cases.

### Conclusion and future works

Data collection is an important topic in WSNs because it is energy-consumed and affects the network lifetime. The use of mobile sink can significantly reduce the energy consumption and prolong the network lifetime. In this article, we proposed an end-to-end data collection strategy, which selects appropriate sensor nodes as CPs and plans a path of visiting all CPs for mobile sink simultaneously. The proposed strategy is based on ACO and can be extended to combining with other data collection approaches. The experimental results reveal that the proposed end-to-end strategy can achieve a better performance in terms of network lifetime, total energy consumption, and FI, especially in the UD scenario.

In the future, we will extend our end-to-end strategy to combining with other data collection methods to demonstrate the generalization capacity of our method. Currently, we do not pay much attention to the runtime performance issues, it is promising for further performance improvements on other methods.

### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the University Innovation Project of Guangdong Province under grant 2021KTSCX264.

### References

1. Hussain S, Schaffner S and Moseychuck D. Applications of wireless sensor networks and RFID in a smart home environment. In: 2009 seventh annual communication networks and services research conference, Moncton, NB, Canada, 11–13 May 2009, pp.153–157. New York: IEEE.

2. Shnayder V, Chen Br, Lorincz K, et al. Sensor networks for medical care. In: Proceedings of the 3rd international conference on embedded networked sensor systems, pp.314–314, https://dl.acm.org/doi/10.1145/1098918.1098979

3. Mainwaring A, Culler D, Polastre J, et al. Wireless sensor networks for habitat monitoring. In: Proceedings of the 1st ACM international workshop on wireless sensor networks and applications, pp.88–97, https://dl.acm.org/doi/10.1145/570738.570751

4. Ahmad N, Hussain M, Riaz N, et al. Flood prediction and disaster risk analysis using GIS based wireless sensor networks, a review. J Basic Appl Sci Res 2013; 3(8): 632–643.

5. Hussain MA, Khan P and Kyung Sup K. WSN research activities for military application. In: 2009 11th international conference on advanced communication technology, volume 1, Gangwon, Korea, 15–18 February 2009, pp.271–274. New York: IEEE.

6. Amutha J, Sharma S and Nagar J. WSN strategies based on sensors, deployment, sensing models, coverage and energy efficiency: review, approaches and open issues. Wireless Pers Commun 2020; 111(2): 1089–1115.

7. Perillo M, Cheng Z and Heinzelman W. On the problem of unbalanced load distribution in wireless sensor networks. In: IEEE global telecommunications conference workshops, 2004. GlobeCom workshops 2004, Dallas, TX, 29 November–3 December 2004, pp.74–79. New York: IEEE.
8. Wen W, Zhao S, Shang C, et al. EAPC: energy-aware path construction for data collection using mobile sink in wireless sensor networks. *IEEE Sens J* 2017; 18(2): 890–901.

9. Nabavi SR, Eraghi NO and Torkestani JA. WSN routing protocol using a multiobjective greedy approach. *Wirel Commun Mob Com* 2021; 2021: 6646669.

10. Maheshwari P, Sharma AK and Verma K. Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization. *Ad Hoc Netw* 2021; 110: 102317.

11. Bouakkaz F and Derdour M. Maximizing WSN life using power efficient grid-chain routing protocol (PEGCP). *Wireless Pers Commun* 2021; 117(2): 1007–1023.

12. Soundaram J and Arumugam C. Genetic spider monkey-based routing protocol to increase the lifetime of the network and energy management in WSN. *Int J Commun Syst* 2020; 33(14): e4525.

13. Radhika M and Sivakumar P. Energy optimized micro genetic algorithm based leach protocol for WSN. *Wirel Netw* 2021; 27(1): 27–40.

14. Shafiq M, Ashraf H, Ullah A, et al. Robust cluster-based routing protocol for IoT-assisted smart devices in WSN. *CMC-Comput Mater Con* 2021; 67(3): 3505–3521.

15. Ma M and Yang Y. Data gathering in wireless sensor networks with mobile collectors. In: 2008 *IEEE international symposium on parallel and distributed processing*, Miami, FL, 14–18 April 2008, pp.1–9. New York: IEEE.

16. Ma M, Yang Y and Zhao M. Tour planning for mobile data-gathering mechanisms in wireless sensor networks. *IEEE T Veh Technol* 2012; 62(4): 1472–1483.

17. Chatzigiannakis I, Kinalis A, Nikoletseas S, et al. Fast and energy efficient sensor data collection by multiple mobile sinks. In: *Proceedings of the 5th ACM international workshop on mobility management and wireless access*, pp.25–32, https://dl.acm.org/doi/abs/10.1145/1298091.1298097.

18. Tang S, Yuan J, Li X, et al. Dawn: energy efficient data aggregation in WSN with mobile sinks. In: *2010 IEEE 18th international workshop on quality of service (IWQoS)*, Beijing, China, 16–18 June 2010, pp.1–9. New York: IEEE.

19. Liang W and Luo J. Network lifetime maximization in sensor networks with multiple mobile sinks. In: *2011 IEEE 36th conference on local computer networks*, Bonn, 4–7 October 2011, pp.350–357. New York: IEEE.

20. Donta PK, Rao BSP, Amgoth T, et al. Data collection and path determination strategies for mobile sink in 3D WSNs. *IEEE Sens J* 2019; 20(4): 2224–2233.

21. Zhu C, Wu S, Han G, et al. A tree-cluster-based data-gathering algorithm for industrial WSNs with a mobile sink. *IEEE Access* 2015; 3: 381–396.

22. Fellah K and Kechar B. New approach based on Hilbert curve for energy efficient data collection in WSN with mobile sink. *IET Wirel Sens Syst* 2020; 10(5): 214–220.

23. Yun Y and Xia Y. Maximizing the lifetime of wireless sensor networks with mobile sink in delay-tolerant applications. *IEEE T Mobile Comput* 2010; 9(9): 1308–1318.

24. Xing G, Wang T, Xie Z, et al. Rendezvous planning in wireless sensor networks with mobile elements. *IEEE T Mobile Comput* 2008; 7(12): 1430–1443.

25. Almi’ani K, Viglas A and Libman L. Energy-efficient data gathering with tour length-constrained mobile elements in wireless sensor networks. In: *IEEE local computer network conference*, Denver, CO, 10–14 October 2010, pp.582–589. New York: IEEE.

26. Salarian H, Chin KW and Naghdy F. An energy-efficient mobile-sink path selection strategy for wireless sensor networks. *IEEE T Veh Technol* 2013; 63(5): 2407–2419.

27. Krishnan M and Lim Y. Reinforcement learning-based dynamic routing using mobile sink for data collection in WSNs and IoT applications. *J Netw Comput Appl* 2021; 194: 103223.

28. Chang CY, Chen SY, Chang IH, et al. Multirate data collection using mobile sink in wireless sensor networks. *IEEE Sens J* 2020; 20(14): 8173–8185.

29. Donta PK, Amgoth T and Annavarapu CSR. An extended ACO-based mobile sink path determination in wireless sensor networks. *J Amb Intel Hum Comp* 2021; 12(10): 8991–9006.

30. Almesaeed R and Jedidi A. Dynamic directional routing for mobile wireless sensor networks. *Ad Hoc Netw* 2021; 110: 102301.