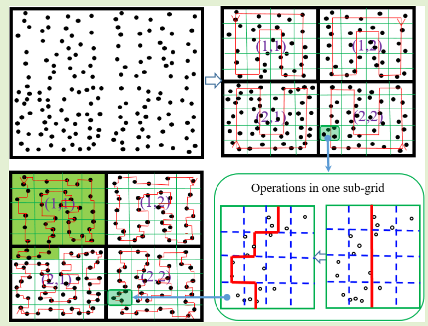
CORL^2: Exploring Cooperative Opportunities, Reducing Latency and Prolonging Network Lifetime for Data Collection Using Mobile Sinks in Wireless Sensor Networks

Weimin Wen, Chih-Yung Chang, *Member, IEEE*, Shih-Jung Wu, and Diptendu Sinha Roy



***Abstract*—Data collection is one of the most important issues in Wireless Sensor Networks (WSNs). Many data collection algorithms have been proposed for collecting data using multiple mobile sinks. However, the anchor selection strategy, which significantly impacts the network lifetime, still can be improved. In addition, the collaboration between multiple mobile sinks is still rare, resulting in the long latency of each round of data collection. This paper proposes an Exploring Cooperative Opportunities, Reducing Latency and Prolong Network Lifetime algorithm for Data Collection, called *CORL*^{2}, aiming to improve the network lifetime and reduce the latency of data collection. Initially, the proposed *CORL*^{2} partitions the monitoring region into several equal-sized subregions and assigns each mobile sink to each subregion. Then *CORL*^{2} further partitions each subregion into several equal-sized grids and constructs a path passing through all grids in that subregion. To reduce the energy consumption for packet forwarding, the proposed *CORL*^{2} further partitions each grid into several small grids and adjusts the path such that it can pass through high-density small grids, aiming to maximize the number of anchors and prolong the network lifetime. Then the cooperative strategy is applied to cooperatively reduce the maximal latency of data collection in all subregions and balance the data collection workload between neighboring mobile sinks. Performance evaluations reveal that the proposed *CORL*^{2} outperforms existing data collection algorithms in terms of latency, energy consumptions, the workload balancing improving index(*WBII*), and network lifetime for data collection in the given WSNs.**

***Index Terms*—Network lifetime, mobile sink, WSN, latency, cooperative opportunities, data collection.**

I. Introduction

WIRELESS sensor networks (WSNs) which consist of a large number of low-cost sensors have been used in many applications. A typical example is the healthcare systems where varieties of medical sensors are attached to the patients to collect their vital signs [1], [2]. The WSNs were also widely applied to smart living or home monitoring applications. A number of sensor-based intelligent devices [3], [4] were connected with each other to detect human behavior, aiming to improve living quality. In addition to the indoor applications, the sensors were also deployed in the outdoor for environment monitoring [5], traffic monitoring [6] or smart city [7]. In these applications, collecting data from sensors is one of the most important tasks in WSNs. In literature, many mechanisms have been proposed for data collection. The simplest way [8], [9] is accomplished by multi-hop forwarding from each sensor to the sink node. However, multi-hop forwarding led to the problem of network disconnection which impacted the network lifetime. This occurs because that the sensors closer to the sink consume much more energy for data forwarding than those sensors far away from the sink. How to collect data from sensors for prolonging the network lifetime is still a big challenge.

In recent years, many studies have proposed data collection schemes using the mobile sink. These studies are typically divided into two classes: single mobile sink and multiple

Manuscript received April 13, 2021; revised June 12, 2021; accepted June 14, 2021. Date of publication June 22, 2021; date of current version August 31, 2021. This work was supported by the Engineering Technology Research Center of Intelligent Perception and Health Endowment Anhui Education Department, Natural Science Foundation of Anhui Higher Education under Grant KJ2019A0648. The associate editor coordinating the review of this article and approving it for publication was Prof. Stefan Knauth. *(Corresponding author: Chih-Yung Chang.)*

Weimin Wen is with the School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China (e-mail: wenweim@chzu.edu.cn).

Chih-Yung Chang and Shih-Jung Wu are with the Department of Computer Science and Information Engineering, Tamkang University, Taipei 25137, Taiwan (e-mail: cychang@mail.tku.edu.tw; wushihjung@mail.tku.edu.tw).

Diptendu Sinha Roy is with the Department of Computer Science and Engineering, National Institute of Technology, Shillong 793003, India (e-mail: diptendu.sr@nitm.ac.in).

Digital Object Identifier 10.1109/JSEN.2021.3091545

mobile sinks. The class of single mobile sink can be further divided into no-data-forwarding and data-forwarding.

In the class of single mobile sink, some studies [10], [11] adopted a mobile sink to patrol all sensor nodes and directly collect data from them without data-forwarding. In study [12], each sensor transmitted data to the bus adopting different power levels based on its remaining energy. Since there is no need for data forwarding, the energy consumptions of sensor nodes were balanced. However, the length of the constructed path for the mobile sink is too long, which raises the time-consuming problem for the mobile sink to visit each sensor. Moreover, it also leads to the problem of buffer overflow of some sensor nodes or energy depletion of the mobile sink.

To cope with the long path problem, some other studies [13]–[15] adopted the partial data-forwarding scheme which selected a set of sensors as anchors and constructed a path for the mobile sink to pass through anchors and collect data from them. All other sensors sent their data to the closest anchors in a multi-hop manner. Study [16] partitioned the monitoring region into several grids, constructed a regular path in balanced grids and adjusted the path segments for these unbalanced grids. Study [17] proposed a trajectory planning scheme (*DAAP*) for the mobile sink to collect data from sensors in delay-harsh applications. However, the selected anchors might not be appropriate. Besides, the sensors closer to the anchors will consume more energy for data forwarding.

There were some other studies [18], [19] that fall in the second class using multiple mobile sinks, aiming to reduce both the path length and time required for each round of data collection. To achieve this, the whole network region was divided into several subregions. A mobile sink was assigned to each subregion. Then the mobile sink constructed a path that passed through all sensor nodes in its subregion. The mobile sink then traversed the path and collected data from these nodes and then sent those data to the base station. Study [20] adopted multiple mobile sinks to collect and relay data in a cooperative manner. However, most of them did not take into account the cooperation for balancing workloads between the mobile sinks. As a result, the latency of data collection was long. The established route still can be improved.

Given multiple mobile sinks and a set of deployed sensors, this paper proposes a cooperative data collection algorithm, called *CORL*^{2}, which selects as many as possible anchors with higher remaining energy. Then the proposed *CORL*^{2} explores the cooperation opportunities between neighboring mobile sinks, aiming to reduce the latency of data collection. Initially, the *CORL*^{2} partitions the monitoring region into several equal-sized subregions according to the number of mobile sinks and then assign mobile sinks to each subregion. Then the *CORL*^{2} further partitions each subregion into several equal-sized grids and the designed path patterns to construct a path passing through all grids in each subregion. The designed path patterns aim to explore more cooperative opportunities from neighboring subregions. To reduce the energy consumption for packet forwarding, the proposed *CORL*^{2} further selects more anchors in each grid according to the density of sensor node distribution and the residual energy and then adjusts the constructed path in each subregion. Then the cooperative policy is applied for neighboring mobile sinks to reduce the maximal latency time of data collection in all subregions and balance the data collection workload between neighboring mobile sinks. The following highlights the contributions and key features of this paper.

1) ***Balancing the energy consumptions of anchors.*** The proposed *CORL*^{2} selects more anchors in each grid according to the density and the residual energies of sensors and then adjusted the constructed path in each grid. Compared to the existing study [16], the proposed *CORL*^{2} can better balance the energy consumptions of anchors and hence improve the network lifetime.
2) ***Exploring the cooperative opportunities.*** This paper proposed the designed path patterns for exploring the cooperative opportunities between neighboring subregions. In addition, before executing the data collection task, the neighboring mobile sink calculates the number of sensors and the path length for its subregion and then dynamically adjusts the size of the subregion. Compared with the existing studies [16], [19], the proposed *CORL*^{2} can balance the workloads of neighboring mobile sinks as well as reduce the latency time of data collection.
3) ***Improve the freshness of collected data.*** The freshness of collected data highly depends on the time of each round for data collection. This paper proposes the cooperative policy and explores the cooperative opportunities between neighboring mobile sinks, which reduce the maximal latency of each round of data collection. Compared with the existing work [16], the freshness of collected data can be significantly improved.
4) ***Hierarchically partitioning the region for better mobile sink dispatching, path construction, and anchor selection.*** This paper partitions the region into three layers, including subregions, grids, and small grids. The region partitioning aims for work distribution for the multiple mobile sinks. Then the proposed *CORL*^{2} further partitions each subregion into several grids for path construction. Then each grid is further partitioned into small grids such that the sensors in a high density small grids and with high remaining energy can be selected as the anchors.

The remaining parts of this paper are organized as follows. Section II reviews the related works and compares them with our work. Section III presents the network environment, assumptions, and problem formulation of this paper. Section IV presents the designing concepts and details of the proposed *CORL*^{2}. Section V presents the performance improvement of the proposed mechanism against the existing studies. Finally, Section VI presents the conclusions.

II. Related Works

In literature, many studies adopted mobile sink for collecting data from sensors. These studies can be further classified into two categories: single mobile sink and multiple mobile sinks. The following briefly reviews these related works.

A. Single Mobile Sink

The class of single mobile sink can be further divided into no-data-forwarding and data-forwarding. Some studies adopted a mobile sink to visit all sensor nodes and collect data from them. Study [10] proposed an approach to cope with the problem of data mule scheduling (*DMS*). The heuristic algorithm aimed to minimize the latency of data collection and improve the reliable transmission of data by controlling the speed. The heuristic algorithm provided the functions of the path selection, speed control, and job scheduling for the mobile sink to visit and collect data from each sensor.

Somasundara *et al*. [11] proposed an approach for scheduling the mobile sink, aiming to construct a path for collecting data from all sensors before the buffer of each sensor overflows. Each sensor generated data at a different rate in different areas. The mobile sink collected data by visiting each sensor at a constant speed. However, visiting each sensor is time consuming, especially for a large scale wireless sensor network.

Study [12] proposed a distributed bus-based Data collection (*DBDC*) algorithm, aiming to maximize the amount of collected data and the network lifetime of wireless sensor networks. Each sensor transmitted data to the bus adopting different power levels based on its remaining energy. The algorithm allowed each sensor to locally negotiate with neighbors and schedule its time slots based on the size of buffered data, remaining energy, and the number of time slots contacted with the bus. However, sensors were required to synchronize their clocks, increasing the complexity of the MAC protocol.

To cope with the long path problem, some other studies [13]–[17] adopted the data-forwarding scheme when developing the data collection mechanisms. These studies selected a set of sensors as anchors and constructed a path for the mobile sink to pass through anchors, aiming to collect data from them. Studies [13] and [14] proposed heuristic algorithms that selected sensors to be visited based on the number of forwarding data packets and the hop distance of the tour. However, *WRP* did not consider the length of the path from the current anchor position to the next one. The path length can be further improved.

Study [15] proposed an energy-aware path construction algorithm (*EAPC*). The *EAPC* evaluated the benefit of each sensor node and selected those sensors with maximal benefits to act as anchors. Then a path was constructed for passing through the anchors. The *EAPC* considered the cost of length between any two continuous anchors. As a result, the constructed path can likely visit the maximal number of anchors, which balanced the forwarding workload and hence improved the network lifetime. However, the path length of the mobile sink was increased with the network size, raising the problem of high latency for data collection.

In [16], a joint density-aware and energy-limited path construction algorithm for data collection (*DEDC*) was proposed. The objective of the paper was to select as more as possible appropriate anchors under the path length constraint for prolonging the network lifetime. The *DEDC* firstly partitioned the monitoring region into several grids and then applied the agglomerative hierarchical clustering (*HC*) algorithm to identify those grids to be balanced or unbalance grids. After that, the *DEDC* constructed a regular path in balanced grids and adjusted the path segments for these unbalanced grids. However, the *DEDC* did not take into account the residual energy of those anchors when adjusted the constructed path in each balance grid. When the size of the network area or the number of sensors grows, the path length of the mobile sink was increased accordingly. The mobile sink spent much more time for each round and hence caused high latency for data collection.

Study [17] proposed a trajectory planning scheme (*DAAP*) for the mobile sink to collect data from sensors in delay-harsh applications. The proposed *DAAP* aimed to construct a short path for the partitioned WSNs. The *DAAP* firstly found the center point of each isolated segment and then constructed a path connecting all center points. After that, all sensors in each isolated segment forwarded their data to the center point of that segment in a multi-hop manner. Finally, the mobile sink collected data from the center point of each isolated segment along the constructed path. However, the multi-hop forwarding might cause network disconnection in each isolated segment because that the sensors closer to the center point consumed more energy for packet forwarding.

B. Multiple Mobile Sinks

Santos *et al*. [18] proposed an approach for solving the problem of building cluster-based topologies for wireless sensor networks with several sinks. The study considered the deployment of a wireless sensor network as a graph. Then the graph is divided into an independent dominating set with connecting requirements (*IDSC*) and each *IDSC* was assigned a mobile sink. After that, the clustered model was used to balance the number of sensors in each *IDSC*, reducing data forwarding times and improving energy consumption in the network. However, the paper did not take into account the cooperation between the mobile sinks. As a result, the latency of data collection was long.

Study [19] proposed a collaborative data collection algorithm (*CDCA*) which aimed to reduce data gathering latency, increase the network throughput and decrease the number of mobile elements. The main idea of the *CDCA* was to divide the network field into several partitions, assign each mobile sinks to each partition, and then cooperatively collect data from sensors to the base station. However, the *CDCA* did not take into account the freshness of the collected data. As a result, the latency of collaborative data collection was long for a large scale sensor network. The collaborative data collection algorithm still can be improved.

Study [20] proposed a cooperative data-gathering scheme (*HiCoDG*) which adopted multiple mobile sinks to collect and relay data in a cooperative manner. Each mobile collector collects data from sensors and forwards them to the mobile relay, which will deliver the received data to the sink. The objective of *HiCoDG* is to minimize the traveling distances of mobile sinks by finding the optimal paths for the local and global mobile sinks. However, it did not consider the distance between the current cluster-head and the next one, resulting in the long latency of data collection for a large scale sensor network.

All of the above algorithms emphasized the improvement of energy unbalanced or coped with the latency problem. However, existing algorithms adopted a single mobile sink spent much time for each round and hence cause high latency for data collection. On the other hand, existing algorithms adopted multiple mobile sinks did not develop the cooperative policy to balance the tasks of data collection in multiple mobile sinks. Different from the previous studies, the proposed data collection algorithm, called *CORL*^{2}, initially partitions the monitoring region into several subregions. Then it assigns each mobile sink with one subregion and constructs the initial path for each mobile sink. To reduce the energy consumption for packet forwarding, the proposed *CORL*^{2} further selects more anchors in each grid according to the density of sensor node distribution and the residual energy of sensors and then adjusts the constructed path in each subregion. Finally, the *CORL*^{2} explores the cooperation opportunities between neighboring mobile sinks, aiming to reduce the maximal latency of each round of data collection in a large-scale sensor network.

Table I summarizes the comparisons between the proposed mechanism and the related studies. The 'Multiple mobile sinks' present whether or not the proposed mechanism is considered multiple mobile sinks. The 'Data forwarding' indicates whether or not the proposed mechanism adopted the data forwarding policy. The 'Adjusting path' indicates whether or not the path is firstly constructed and then further improved aiming to find more anchors or construct a more efficient path. The last matric is 'Cooperation for balancing work loads' which examines the existence of cooperation relation between multiple sinks. In comparison, the proposed *CORL*^{2} selects more anchors by considering the density and the residual energies of sensors and then adjusts the constructed path in each grid to better balance the energy consumptions of anchors. Furthermore, the proposed *CORL*^{2} applies the cooperative policy for reducing the maximal latency of each round when executing the data collection task.

III. Network Environment and Problem Formulation

This section presents the network environment and the assumptions of the given WSNs. The problem formulation is subsequently presented.

TABLE I
COMPARISONS OF PROPOSED CORL^2 AND EXISTING MECHANISMS

Studies	Multiple mobile sinks	Data forwarding	Adjusting path	Cooperation for balancing workloads
[10]	×	×	×	×
[11]	×	×	×	×
[12]	×	×	×	×
[13]	×	○	×	×
[14]	×	○	×	×
[15]	×	○	×	×
[16]	×	○	○	×
[17]	×	○	×	×
[18]	○	○	×	×
[19]	○	○	×	×
[20]	○	○	×	×
The proposed *CORL*^{2}	○	○	○	○

A. Network Environment

This paper assumes that a set of n static sensors $S = \{s_1, s_2 \cdots, s_n\}$ are randomly deployed in a monitoring region R which is divided into equal-sized $h \times z$ subregions. Let $R = \{R_{1,1}, R_{1,2} \cdots, R_{h,z}\}$ denote the set of those subregions. Let l and w denote the length and width of each subregion, respectively. There are m mobile sinks $\{M_{1,1}, M_{1,2}, \cdots M_{h,z}\}$ assigned to the corresponding subregion, where m=h×z. Let $P = \{P_{1,1}, P_{1,2}, \cdots P_{h,z}\}$ denote the set of those paths where the path $P_{i,j}$ denotes the path constructed in the subregion $R_{i,j}$. The mobile sink $M_{i,j}$ moves along the path $P_{i,j}$ to collect data from sensors in $R_{i,j}$ with a constraint of maximal length $L_{i,j}$. The constraint of path length for the mobile sink reflects the fresh requirement of data carried by the mobile sink and the limited energy of each mobile sink. The paper assumes that every mobile sink is aware of its location while moving and the locations of all static sensors in its subregion. The sensors which can directly transmit data to the mobile sink is called an anchor. Let $A_{i,j} = \{a_{i,j,1}, a_{i,j,2} \cdots, a_{i,j,b}\}$ be the set of b anchors in the subregion $R_{i,j}$. It is obvious that each anchor node falls in its communication range of mobile sink and can directly transmit data to the mobile sink when the mobile sink moves along a path $P_{i,j}$.

Assume that the communication ranges of all mobile sinks and sensors are identical. Each mobile sink has rich power or is supported by an energy harvesting system. On receiving data from sensor nodes, each mobile sink can transmit the data to the base station through GSM. The function of the base station is to upload the data to the internet for further processing.

Fig.1 illustrates an example of several anchors and four mobile sinks in a monitoring region. In Fig. 1, the empty circles represent the static sensor nodes and the purple circles represent the anchors whereas the red trilateral denotes the base station. The monitoring region has been partitioned into four subregions and each subregion is assigned to a mobile sink collection data. In each subregion, all static sensors send their data to the nearest anchor, and the anchor stores the data

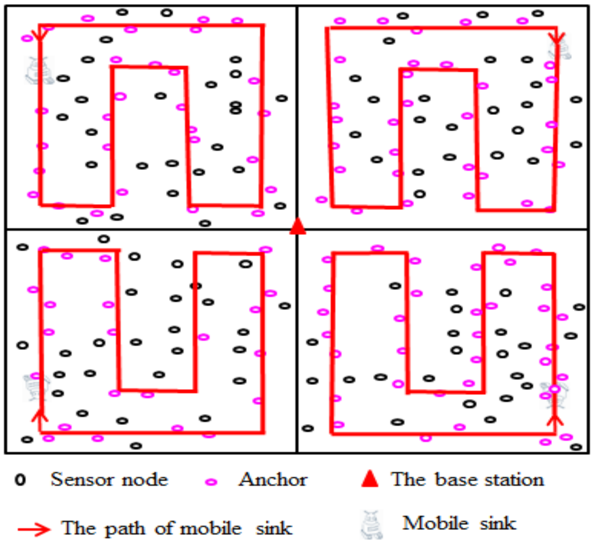


Fig. 1. Network environment.

until the mobile sink visits it. Furthermore, the mobile sink moves along the established path, which passes through all anchors to collect data and then sends those data to the base station through GSM.

B. Problem Formulation

This paper essentially studies how to construct the path of each mobile sink aiming at maximizing the lifetime of the given WSN while balancing the data collection time of each mobile sink. The lifetime of a WSN is defined as the length of time from the point at which the network runs to the point at which the first sensor node runs out of energy. The power consumption of a sensor mainly includes transmitting and receiving data. It is assumed that the size of each packet is k bit. The energy consumption for the sender s_i to transmit one packet (k bits) to its neighbor s_j is measured by:

$$E_t = k\varepsilon_1 + kd_{i,j}^2\varepsilon_2 \tag{1}$$

where ε_1 is the transmitting circuit of energy consumption per bit, ε_2 is the amplifier circuit of energy consumption per bit and d_{ij} denotes the distance between s_i and s_j.

The energy consumption of each node for receiving one packet (k bits) can be measured by (2):

$$E_r = k\beta \tag{2}$$

where β is the receiving circuit energy consumption per a bit.

In each subregion, the sensors which fall in the communication range of the constructed path will play the role of anchors. All sensors in this subregion will organize many trees each rooted by one anchor. Along with the constructed tree topology, sensors will forward their data to the root of the corresponding tree. Let $C(s_j)$ denote the number of children of sensor s_j. Assume that there are x packets generated by each sensor in each round. The total energy cost of the sensor s_j in a round is expressed as:

$$E_j^{con} = (C(s_j)x + x) \times E_t + (C(s_j)x \times E_r) \tag{3}$$

The network lifetime highly depends on the sensor nodes with the lowest residual energy. Since that each anchor node consumes more energy than any sensor in its tree, this study aims to minimize the energy consumption of the bottleneck anchor node, which consumes the largest energy. Exp. (4) reflects the goal of this paper.

First Objective:

$$\underset{s_i \in A}{Min\,Max}\; E_i^{con} \tag{4}$$

Let $D_{i,j}$ denote the time required for mobile sink $M_{i,j}$ moving along the path $P_{i,j}$ to collect all data in the subregion $R_{i,j}$. To keep the data to be fresh, the maximal $D_{i,j}$ of mobile sink $M_{i,j}$ should be further minimized. Exp. (5) reflects this goal.

Second Objective:

$$\underset{M_{i,j} \in M}{Min\,Max}\; D_{i,j} \tag{5}$$

The following presents some constraints which should be satisfied when achieving objectives (4) and (5). Let $q_{i,j}$ denote the number of anchors in the subregion $R_{i,j}$. An important constraint for the path length is the limited energy of each mobile sink $M_{i,j}$. Let $E_{i,j}^{receiving}$ denote the energy required for mobile sink receiving packets from $q_{i,j}$ anchors. The value of $E_{i,j}^{receiving}$ can be obtained by

$$E_{i,j}^{receiving} = \sum_{a_y \in A_{i,j}} (C(a_y)x + x) \times E_r \tag{6}$$

Let $E_{i,j}^{moving}$ denote the energy consumption of a mobile sink for moving one unit distance. Let E_{full} denote the original battery energy of each mobile sink. Let $L_{i,j}^{max}$ denote the longest path that mobile sink can move under the constraint of its battery energy. Let $L_{i,j}$ denote the path length of $P_{i,j}$. The total energy required for every mobile sink moving along the path $P_{i,j}$ is:

$$E_{full} = E_{i,j}^{moving} \times L_{i,j}^{max} + E_{i,j}^{receiving} \tag{7}$$

$$L_{i,j}^{max} = (E_{full} - E_{i,j}^{receiving})/E_{i,j}^{moving} \tag{8}$$

Constraint (9) gives the upper and lower bounds of path length $L_{i,j}$.

1) Path Length Constraint:

$$L_{i,j}^{min} \le L_{i,j} \le L_{i,j}^{max} \tag{9}$$

where $L_{i,j}^{min}$ denotes the shortest Hamiltonian path passing through each anchor node in the subregion $R_{i,j}$.

Another constraint is that the data generated by each sensor only can be received by at most one mobile sink $M_{i,j}$. Let Boolean variable $\lambda_{i,j,k}$ denote whether or not sensor node s_k can transmit data to the mobile sink $M_{i,j}$. That is,

$$\lambda_{i,j,k} = \begin{cases} 1, & s_k \text{ can transmit data to mobile sink } M_{i,j} \\ 0, & otherwise \end{cases}$$

To ensure those data of each sensor only can receive by one mobile sink, the following Data transportation constraint (10) should be satisfied.

2) Data Transportation Constraint:

$$\sum_{i=1}^{h}\sum_{j=1}^{z}\lambda_{i,j,k}=1,\quad \text{for } \forall \ s_k \tag{10}$$

In this paper, another important constraint is that each anchor should receive all the data packets generated by the sensors in its tree. Let ϑ denote the buffer size of each sensor. Let τ be the data generation rate of each sensor node. Let $T_{i,j}$ denote the time length of each round which is defined by the time required for round trip traversal of the path $P_{i,j}$. Recall that $C(a_j)$ denotes the number of children of anchor a_j. The number of packets received from anchor a_j in each round would be $C(a_j)\tau T_{i,j}$. To prevent the buffer overflow,$T_{i,j}$ should satisfy the following constraint.

3) Buffer Constraint:

$$(C(a_j)+x)\ \tau T_{i,j}\leq\vartheta \tag{11}$$

Let $S_{i,j}$ denote the set of sensors in the subregion $R_{i,j}$. Let $f_{i,j}^{out}$denote the total number of packets sent by all anchors in the subregion $R_{i,j}$ in each round and let $f_{i,j}^{in}$ denote the number of packets created by all sensors $s_i\in S_{ij}$ in each round. To guarantee that all packets of S_{ij} can be received by the mobile sink $M_{i,j}$, the following Flow Constraint (12) should be satisfied.

4) Flow Constraint:

$$f_{ij}^{out}=\sum_{\forall s_i\in S_{ij}} f_{i,j}^{in} \tag{12}$$

IV. The Proposed *Corl*^{2} Algorithm

The proposed *CORL*^{2} algorithm is composed of three phases: *Initial Path Construction Phase*, *Path Adjustment Phase,* and *Cooperation Phase*. The Initial Path Construction Phase aims to partition each subregion into multiple small grids and construct an initial path along which each mobile sink can move for data collection in each subregion. Herein, it is noticed that all sensors that fall in the communication range of the constructed path will play the role of anchors. The Path Adjustment Phase aims to adjust the path such that the number of anchors can be increased. The increase of anchor nodes can further reduce the number of hops when each sensor forwards its data to the anchor node. As a result, the network lifetime can be further improved. Finally, the Cooperation Phase aims to balance the workload of data collection among mobile sinks, aiming at reducing the latency for data collection.

A. Initial Path Construction Phase

The Initial Path Construction Phase mainly consists of two tasks. The first task aims to partition the monitoring region R into equal-sized $h\times z$ subregions and further partition the subregion into several equal-sized grids. Then the second task aims to construct a path passing through each grid. Each mobile sink moves along the constructed path for data collection.

The monitoring region has been partitioned into m subregions and each mobile sink is responsible for collecting data of sensors in one subregion, where m is the number of mobile sinks. In the first task, each subregion $R_{i,j}$ is further partitioned into a set of $\varsigma = q_1\times q_2$ square grids $R_{i,j}=\{g_{1,1}, g_{1,2}, \cdots, g_{q_1,q_2}\}$, where the number of columns q_2 is even.

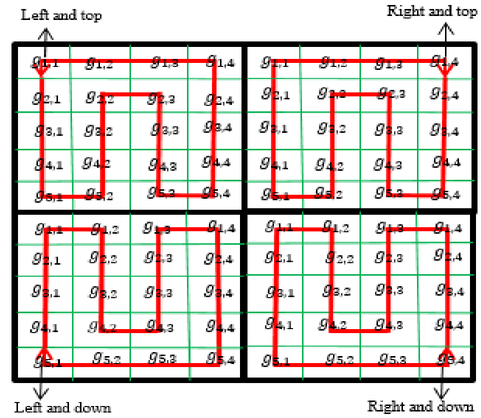


Fig. 2. The initial path and possible initial positions of the mobile sink.

In the second task, a snake-like path is constructed starting from the initial position, passing through each grid once, and finally returning to the initial position. Each mobile sink will move along its path and collect data from sensors in $R_{i,j}$. To explore more collaborative opportunities between neighboring subregions, two path patterns are applied.

The following presents the designed two path patterns. In each subregion $R_{i,j}$, the trajectory $\phi_{i,j}$ of each mobile sink $M_{i,j}$ can be represented by three features $\phi_{i,j}=\left(\varphi,\vec{d},R_{i,j}^{loc}\right)$, where φ denotes the path type, $\vec{d}$ denotes the direction and $R_{i,j}^{loc}$ denote the initial position of the path. Given a subregion $R_{i,j}$, a particular path pattern will be applied to construct the path in $R_{i,j}$.

The first feature of the path pattern is φ, which denotes the type of the constructed path. To maintain the cooperation relation between mobile sinks, Let φ_1 and φ_2 denote the two path types and $\varphi=\{\varphi_1,\varphi_2\}$ be the set of two types. The two path types φ_1 and φ_2 are shown in $R_{1,1}$ and $R_{2,1}$ in Fig.2, respectively. The following path construction rule defines the path type constructed in the subregion $R_{i,j}$.

$$\varphi=\begin{cases}\varphi_1 & \text{when } i \text{ is odd}\\ \varphi_2 & \text{when } i \text{ is even}\end{cases}$$

To maintain the cooperation data collection between neighboring mobile sinks, in each subregion $R_{i,j}$, the mobile sink $M_{i,j}$ should move along the constructed path $P_{i,j}$ in a certain direction. Let $\vec{d}_{i,j}$ denote the moving direction of mobile sink $M_{i,j}$. Mobile sink $M_{i,j}$ should move in the direction $\vec{d}_{i,j}$ according to the following direction rule.

$$\vec{d}_{i,j}=\begin{cases}\vec{d}_{+}\text{when } i+j \text{ is odd}\\ \vec{d}_{-}\text{when } i+j \text{ is even}\end{cases}$$, where $\vec{d}_{+}$ and $\vec{d}_{-}$ denote the clockwise and counterclockwise of the corresponding path.

Similar to the direct rule, the mobile sink $M_{i,j}$ should move to start from the different initial positions. Let 'LT'(Left and

Top), 'LB' (Left and Bottom), 'RT' (Right and Top) and 'RB' (Right and Bottom) denote the four initial positions. The position rule that determines the starting position $R_{i,j}^{loc}$ of mobile sink $M_{i,j}$ is presented below.

$$R_{i,j}^{loc} = \begin{cases} \text{LT}, & \text{if } i \text{ is odd and } j \text{ is odd} \\ \text{RT}, & \text{if } i \text{ is odd and } j \text{ is even} \\ \text{LB}, & \text{if } i \text{ is even and } j \text{ is odd} \\ \text{RB}, & \text{if } i \text{ is even and } j \text{ is even} \end{cases}$$

Base on the above mentioned rules, a path can be constructed for each subregion $R_{i,j}$. Fig.2 depicts the initial path and four possible initial positions of the mobile sink.

Recall that each path length of $M_{i,j}$ is limited by $L_{i,j}$. The total length $L_{i,j}$ of each mobile sink will be divided into three parts, $L_{i,j}^{init}$, $L_{i,j}^{adjust}$ and $L_{i,j}^{help}$, where $L_{i,j}^{init}$ denotes the length of the initial path, $L_{i,j}^{adjust}$ denotes the path length for supporting path adjustment and $L_{i,j}^{help}$ denotes the maximal path length to help the neighboring mobile sink collect data.

$$L_{i,j} = L_{i,j}^{init} + L_{i,j}^{adjust} + L_{i,j}^{help} \tag{13}$$

Next, we will derive the grid size which highly depends on the limited length of the initial path for each mobile sink. Recall that the length and width of each subregion are l and w, respectively. Let notation l_g denote the side length of the square grid. There are w/l_g columns. Ignoring the first and last columns, the length of each vertical red line passing through each column is $l - 2l_g$. The vertical red lines passing through the first and last columns have equal length, which has additional l_g length than the other vertical red lines. The value of $L_{i,j}^{init}$ can be represented by

$$L_{i,j}^{init} = \frac{w}{l_g}\left(l - 2l_g\right) + 2l_g + 2\left(w - l_g\right) \tag{14}$$

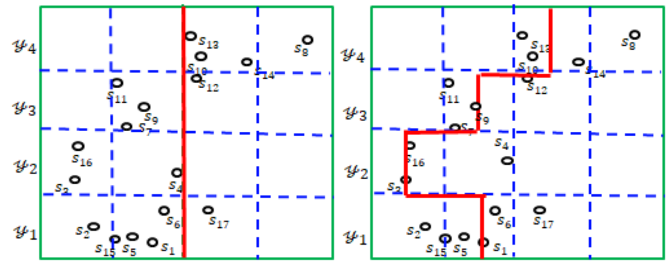
Then the grid length can be obtained by the calculation

$$l_g = \frac{wl}{L_{i,j}^{init}} \tag{15}$$

Till now, the first phase has partitioned the monitoring region into $m = h \times z$ subregions and each subregion is further partitioned into $q_1 \times q_2$ grids with grid length l_g. There are two path patterns, including φ_1 and φ_2, will be applied for constructing the path for each subregion $R_{i,j}$, according to the value of i and j. As a result, this phase has finished the region partitioning and path construction for each subregion. The next step is to perform the operations defined in Phase B which aims to select more anchors to increase the network lifetime and balance the workload of each mobile sink.

B. Path Adjustment Phase

This task aims to adjust the path to increase the number of anchors. The increase of anchor nodes can further reduce the number of hops when each sensor forwards its data to the anchor node. To achieve this, each grid $g_{i,j}$ will be further divided into a set of $\delta = d \times f$ small grids $\mathfrak{N} = \{\mathfrak{y}_{1,1}, \mathfrak{y}_{1,2}, \cdots, \mathfrak{y}_{d,f}\}$. Let notations r_s and r_c denote the sensing radius and communication radius of each sensor, respectively. Let $l_{\mathfrak{y}}$ denote the length of the grid, where $r_s \leq l_{\mathfrak{y}} < r_c$.



(a) The first segment of the path in the grid $g_{1,1}$ (b) Adjust the new segment in the grid $g_{1,1}$

Fig. 3. Example of path adjustment.

Fig. 3 gives an example of the initial path and adjustment path. The grid $g_{1,1}$ is divided into 16 small grids $\mathfrak{y}_{a,b}$, for $1 \leq a < 4$, for $1 \leq b < 4$, where seventeen sensors are deployed. Figs. 3(a) and 3(b) depict the initial path and the adjusted path for the grid $g_{1,1}$, respectively. As shown in Fig. 3(a), there are few sensors or even no sensors deployed around the initial path in each small grid. This can increase the number of hops when each sensor forwards its data to the anchor node. In Fig. 3(b), the path adjustment of the grid algorithm selects $\mathfrak{y}_{1,2}$, $\mathfrak{y}_{2,1}$, $\mathfrak{y}_{3,2}$, and $\mathfrak{y}_{4,3}$ to construct the path of the mobile sink since sensors in these grids have higher remaining energy and can play the role of anchor. Consequently, the number of anchors is increased from 7 to 12, reducing the number of hops when each sensor forwarding its data to the anchor. Therefore, the lifetime of WSN can be prolonged.

To achieve this, the *rich sensor* that has high remaining energy will be first identified. Then the *richest small grid* which has maximal remaining energy in each row will be selected to be passed by the adjusted path. Let E_i^{init} and E_i^{rem} denote the initial energy and remaining energy of the sensor s_i, respectively. Let $E_{threshold}$ denote the threshold value for the sensor to play the role of anchor. Let ρ_i denote whether or not the sensor s_i can serve as a *rich sensor*, That is,

$$\rho_i = \begin{cases} 1, & if\ E_i^{rem} \geq E_{threshold} \\ 0, & otherwise \end{cases}$$

The following presents the path adjustment operations for the grid $g_{i,j}$. The adjustment operations are executed round by round. In each round, one row of the grid $g_{i,j}$ will be processed. Without the loss of generality, the following presents how to find the *richest small grid* in the $a-$th row which can be passed in the adjusted path. Let ψ_a denote the $a-$th row of the grid $g_{i,j}$. Let set $A_{a,b}$ denote the set of *rich sensors* in a small grid $\mathfrak{y}_{a,b}$. The basic idea behind this task is to include sensor s_i in set $A_{a,b}$ if sensor s_i satisfying $\rho_i = 1$ and it falls in a small grid $\mathfrak{y}_{a,b}$. Let $E_{a,b}^{\mathfrak{y}}$ denote the energy of a rich small grid, which is measured by the sum of all remaining energy of rich sensors in the small grid $\mathfrak{y}_{a,b}$. That is,

$$E^{\mathfrak{y}}_{a,b} = \sum_{s_i \in A_{a,b}} E^{rem}_i \tag{16}$$

Let $\mathfrak{y}^{max}_a$ denote rich small grid which is the small grid with maximal *energy* in the $j-$th row. That is,

$$\mathfrak{y}^{max}_a = arg \max_{\mathfrak{y}_{a,b},\ 1 \le b \le f} E^{\mathfrak{y}}_{a,b} \tag{17}$$

In the $j-$th row, the small grid $\mathfrak{y}^{max}_a$ will be passed through by the adjusted path. Therefore, the next step aims to connect the $\mathfrak{y}^{max}_a$ in the $a-$th row to form the adjusted path.

Let $g^{max}_{i,j}$ denote the set of all small grids $\mathfrak{y}^{max}_a$ for all $1 \le a \le d$. That is

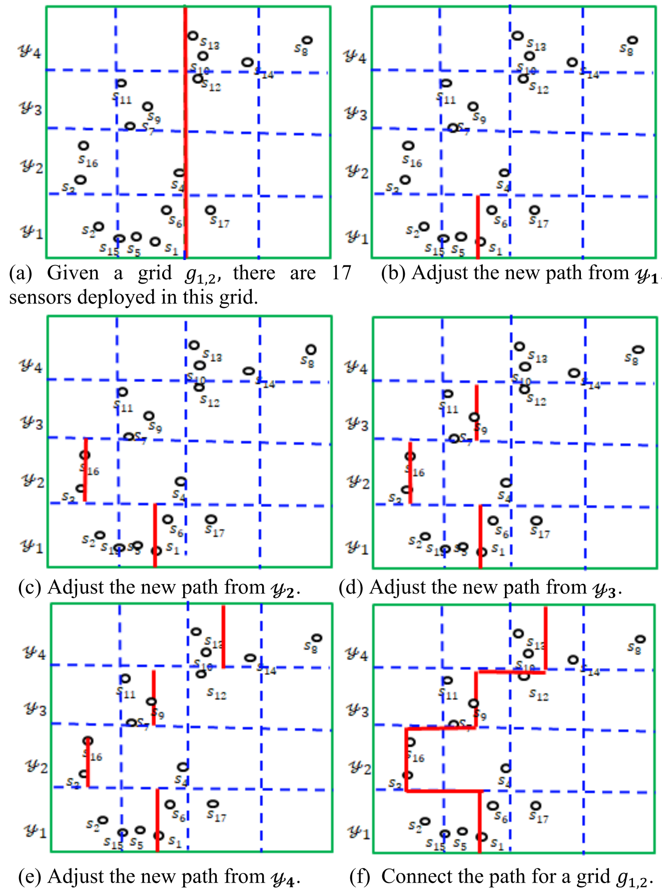
$$g^{max}_{i,j} = \left\{\mathfrak{y}^{max}_a | 1 \le a \le d\right\} \tag{18}$$

The path passing through the grid $g_{i,j}$ will be reconstructed by connecting all small grids $\mathfrak{y}^{max}_a \in g^{max}_{i,j}$. Let $p_{i,j}$ denote the adjusted path which passes through $g_{i,j}$. Let ξ_a denote the central line of small grid $\mathfrak{y}^{max}_a$. The path $p_{i,j}$ can be constructed by operation:

$$p_{i,j} = \oplus_{1 \le a \le d}\ (\xi_a) \tag{19}$$

where $\oplus$ denotes the connection operation performed on all segments ξ_a, for $1 \le a \le d$.

The following gives an example to illustrate the path adjustment algorithm. Fig. 4(a) gives the initial path of a grid $g_{1,2}$. The grid $g_{1,2}$ is divided into sixteen small grid$\{\mathfrak{y}_{1,1}, \mathfrak{y}_{1,2} \cdots, \mathfrak{y}_{4,4}\}$. The path adjustment task firstly identifies the rich small grids $\mathfrak{y}^{max}_1 = \mathfrak{y}_{1,2}$, $\mathfrak{y}^{max}_2 = \mathfrak{y}_{2,1}$, $\mathfrak{y}^{max}_3 = \mathfrak{y}_{3,2}$ and $\mathfrak{y}^{max}_4 = \mathfrak{y}_{4,3}$ as shown in Fig. 4(b). Therefore, we have $A^{most} = \left\{\mathfrak{y}_{1,2},\ \mathfrak{y}_{2,1},\ \mathfrak{y}_{3,2}, \mathfrak{y}_{4,3}\right\}$. Then the path adjustment of the grid task constructs the new path as shown in Fig.4(f) where sensors $s_1, s_5,\ s_6,\ s_3, s_{16},\ s_7, s_9, s_{11}, s_{10}, s_{12}\ , s_{13}$ and s_{14} serve as the anchor nodes.



(a) Given a grid $g_{1,2}$, there are 17 sensors deployed in this grid. (b) Adjust the new path from ψ_1. (c) Adjust the new path from ψ_2. (d) Adjust the new path from ψ_3. (e) Adjust the new path from ψ_4. (f) Connect the path for a grid $g_{1,2}$.

Fig. 4. Example of path adjustment in a given grid.

C. Cooperation Phase

The goal of this phase is to balance the data collection workload between the neighboring mobile sinks so that all data collection tasks can be earlier completed. To achieve this, the mobile sink $M_{i,j}$ of subregion $R_{i,j}$ calculates the number of sensors and the path length in its subregion, and then considers whether or not to share the data collection task of neighboring mobile sinks. Recall that each mobile sink $M_{i,j}$ is responsible to collect data of sensors in each subregion $R_{i,j}$. In case that subregion $R_{i,j}$ contains a large number of sensors, mobile sink $M_{i,j}$ will spend a long time finishing its data collection task.

Recall that $L_{i,j}$ denotes the maximal path length that mobile sink can move while $L^{init}_{i,j}$ and $L^{adjust}_{i,j}$ denote the initial path length and adjusted path length, respectively. Let L^{help}_a denote the additional path length of $\hat{M}_a$ which is associated to $M_{i,j}$. The value of L^{help}_a can be derived by applying Exp. (20).

$$L^{help}_a = L_{i,j} - L^{init}_{i,j} - L^{adjust}_{i,j} \tag{20}$$

Herein, it is observed that only the pair of neighboring mobile sinks whose paths have distinct directions can have cooperative relations. For example, mobile sinks $M_{1,1}$and $M_{2,1}$ are cooperative pair while mobile sinks $M_{3,1}$and $M_{4,1}$are cooperative pairs. Let $\hat{M}_a$ and $\hat{M}_b$ be a cooperative pair. Let $\tau_{a,b}$ be a Boolean variable, representing the cooperative relation of $\hat{M}_a$ and $\hat{M}_{b.}$ More specifically, let $\tau_{a,b}$ be

$$\tau_{a,b} = \begin{cases} 1, & if\ L^{help}_a - L^{help}_b > L_{threshold} \\ -1, & L^{help}_b - L^{help}_a > L_{threshold} \\ 0, & otherwise \end{cases}$$

Let $\pi_{a,b}$ denote the number of cooperative rows of mobile sinks $\hat{M}_a$ and $\hat{M}_{b.}$. Recall that l_g denotes the side length of the square grid $g_{i,j}$. Let α denote the scale parameter. We have

$$\pi_{a,b} = |(L^{help}_a - L^{help}_b)/(\alpha 2 l_g)| \tag{21}$$

In case that $\tau_{a,b} = 1$, it indicates that $\hat{M}_a$ needs to help $\hat{M}_{b.}$ Let

$$\tilde{y} = (2x - 1) mod\ (\frac{q_2}{2})$$

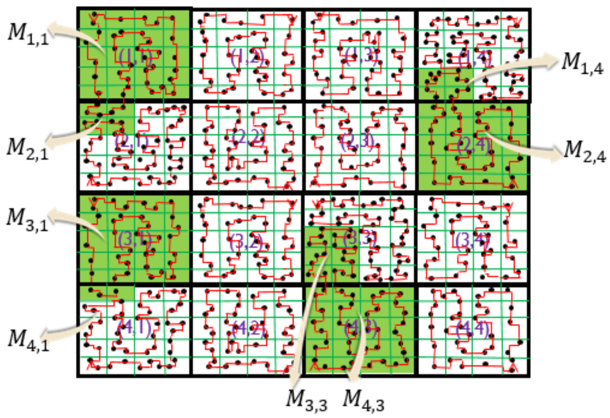


Fig. 5. Example of cooperation algorithm.

The subregion $R_{i,j}$ which is associated with $\hat{M}_a$ should be extended by

$$R_{i,j} = R_{i,j} \bigcup_{x=1}^{\pi_{a,b/2}} \{(g_{x,\tilde{y}}, g_{x,\tilde{y}+1}) | (g_{x,\tilde{y}}, g_{x,\tilde{y}+1}) \in R_{i+1,j}\} \tag{22}$$

$$R_{i+1,j} = R_{i+1,j} \backslash \bigcup_{x=1}^{\pi_{a,b/2}} \{(g_{x,\tilde{y}}, g_{x,\tilde{y}+1}) | (g_{x,\tilde{y}}, g_{x,\tilde{y}+1}) \in R_{i+1,j}\} \tag{23}$$

In another case that $\tau_{x,y} = -1$, it indicates that $\hat{M}_b$ needs to help $\hat{M}_a$. Let $\tilde{x} = (q_1 - x + 1)$ and

$$\tilde{y} = (2x - 1) mod \ (\frac{q_2}{2})$$

The subregion $R_{i+1,j}$ which is associated with $\hat{M}_b$ should be extended by

$$R_{i+1,j} = R_{i+1,j} \bigcup_{x=1}^{\pi_{a,b/2}} \left(\{(g_{\tilde{x},\tilde{y}}, g_{\tilde{x},\tilde{y}+1}) | (g_{\tilde{x},\tilde{y}}, g_{\tilde{x},\tilde{y}+1}) \in R_{i,j}\}\right) \tag{24}$$

$$R_{i,j} = R_{i,j} \backslash \bigcup_{x=1}^{\pi_{a,b/2}} \left(\{(g_{\tilde{x},\tilde{y}}, g_{\tilde{x},\tilde{y}+1}) | (g_{\tilde{x},\tilde{y}}, g_{\tilde{x},\tilde{y}+1}) \in R_{i,j}\}\right) \tag{25}$$

The following gives an example to illustrate the cooperation algorithm. Fig. 5 gives the cooperation between the neighboring mobile sinks. As shown in Fig. 5, $M_{1,1}$ helps $M_{2,1}$, $M_{3,1}$ helps $M_{4,1}$, $M_{2,4}$ helps $M_{1,4}$ and $M_{4,3}$ helps $M_{3,3}$. More specifically, as soon as $M_{1,1}$ finishes its data collecting task in the region $R_{1,1}$, it observes that the neighboring mobile sink $M_{2,1}$ still not finish the data collection task in the region $R_{1,2}$. Therefore, it helps collect data for those sensors in grids $g_{1,1}$, $g_{1,2}$, $g_{2,1}$ and $g_{2,2}$, which are originally belonging to the $M_{2,1}$. With the help from $M_{1,1}$, $M_{2,1}$ can finish its data collection task earlier to achieve the goal of reducing the latency of data collection and improving the data freshness. Recall that only the pair of neighboring mobile sinks whose paths have distinct directions can have cooperative relations. Therefore, the cooperation opportunities of all mobile sinks are identical because the path pattern has been fixed for each subregion. However, the number of cooperative pairs of mobile sinks will be increased with the number of sub-regions.

The following pseudo-algorithm summarizes the proposed $CORL^2$method in this section.

The proposed CORL²algorithm
Input: $S = \{s_1, s_2 \cdots, s_n\}$; $R = \{R_{1,1}, R_{1,2} \cdots, R_{h,z}\}$; $M = \{M_{1,1}, M_{1,2}, \cdots M_{h,z}\}$;
Output: $P'' = \{P''_{1,1}, P''_{1,2} \cdots, P''_{h,z}\}$
/*Initial Path Construction Phase*/
Partition the monitoring region R into equal-sized m=h*z subregions $\{R_{1,1}, R_{1,2} \cdots, R_{h,z}\}$;
Partitions each subregion $R_{i,j}$into a set of $\varsigma = q_1 \times q_2$ square grids $g_{i,j}$, $1 \le i \le q_1, 1 \le j \le q_2$;
Constructs a snake-like path $P_{i,j}$ in each $R_{i,j}$, passing through each grid $g_{i,j} \in R_{i,j}$;
The set of all initial path in the region $P = \{P_{1,1}, P_{1,2} \cdots, P_{h,z}\}$;
/*Path Adjustment Phase */
For each grid $g_{i,j} \in R_{i,j}$ do {
Partition it into a set of $\delta = d \times f$ small grids $\mathfrak{y}_{a,b,}$;}
For each $R_{i,j} \in R$ do{
For each $g_{i,j} \in R_{i,j}${
For each $\mathfrak{y}_{a,b} \in g_{i,j}$ do{
For each sensor $s_i \in \mathfrak{y}_{a,b}${
If ($E_i^{res} \ge E_{threshold}$)
{ $A_{a,b} = A_{a,b} \cup \{s_i\}$;
$E_{a,b}^{\mathfrak{y}} = E_{a,b}^{\mathfrak{y}} + E_i^{res}$;}}}}
For each row $\mathscr{Y}_a \in g_{i,j}$ do{
$\mathfrak{y}_a^{max} = arg \max_{\mathfrak{y}_{a,b} 1 \le a \le f} E_{a,b}^{\mathfrak{y}}$;
$p_{i,j} = \bigoplus_{1 \le a \le d} (\xi_a)$;}}
Constructs an adjusted path $P'_{i,j}$when connecting all $p_{i,j}$ in subregion $R_{i,j}$;
The set of all adjusted path in the region $P' = \{P'_{1,1}, P'_{1,2} \cdots, P'_{h,z}\}$;
/*Cooperation Phase*/
For each $R_{i,j} \in R$ do{
Evaluate the length $L_{i,j}^{help} = L_{i,j} - L_{i,j}^{init} - L_{i,j}^{adjust}$};
Evaluate the number of cooperative pair c in the monitoring region R
For each cooperative pair do{
If ($\tau_{a,b} = 1$)
{ Evaluate $\pi_{a,b} = \lvert (L_a^{help} - L_b^{help})/(\alpha 2 l_g) \rvert$;
$R_{i,j} = R_{i,j} \bigcup_{x=1}^{\pi_{a,b/2}} \{(g_{x,\tilde{y}}, g_{x,\tilde{y}+1}) \| (g_{x,\tilde{y}}, g_{x,\tilde{y}+1}) \in R_{i+1,j}\}$;
$R_{i+1,j} = R_{i+1,j} \backslash \bigcup_{x=1}^{\pi_{a,b/2}} \{(g_{x,\tilde{y}}, g_{x,\tilde{y}+1}) \| (g_{x,\tilde{y}}, g_{x,\tilde{y}+1}) \in R_{i+1,j}\}$;
Else if ($\tau_{a,b} = -1$)
{Evaluate $\pi_{a,b} = \lvert (L_a^{help} - L_b^{help})/(\alpha 2 l_g) \rvert$;
$R_{i+1,j} = R_{i+1,j} \bigcup_{x=1}^{\pi_{a,b/2}} \left(\{(g_{\tilde{x},\tilde{y}}, g_{\tilde{x},\tilde{y}+1}) \| (g_{\tilde{x},\tilde{y}}, g_{\tilde{x},\tilde{y}+1}) \in R_{i,j}\}\right)$;
$R_{i,j} = R_{i,j} \backslash \bigcup_{x=1}^{\pi_{a,b/2}} \left(\{(g_{\tilde{x},\tilde{y}}, g_{\tilde{x},\tilde{y}+1}) \| (g_{\tilde{x},\tilde{y}}, g_{\tilde{x},\tilde{y}+1}) \in R_{i,j}\}\right)$;
Endif}
Endif}
The set of all cooperative paths in the region $P'' = \{P''_{1,1}, P''_{1,2} \cdots, P''_{h,z}\}$
Return the set of cooperative paths $P'' = \{P''_{1,1}, P''_{1,2} \cdots, P''_{h,z}\}$;

D. Complexity Analysis

The proposed *CORL*^{2} mainly consists of Initial Path Construction Phase, Path Adjustment Phase and Cooperation Phase. In the Initialization Phase, the complexities of steps 2, 3 and 4 are O(m).

The following presents the complexity analysis of the Path Adjustment Phase which consists of step 6 to step 20. Steps 7 and 8 partition each grid $g_{i,j}$ into a set of $\delta = d \times f$ small grids $\eta_{a,b}$. This step includes computation for $m = h \times z$ grids. Therefore, the complexities of steps 7 and 8 are O(m). Step 9 performs the *for*-instruction with O(m) loops. Step 10 performs the *for*-instruction with $\varsigma = q_1 \times q_2$ loops, which is a constant value. Step 11 performs the *for*-instruction with $d \times f$ loops, which is also a constant value. Step 12 performs several times depending on the number of sensors in each $\eta_{a,b}$. Combining steps 9 to 12, the total number of loops is n which is the number of sensors in the network. Therefore, the complexities of steps 9-12 are O(n). Steps 16 and 17 perform d loop and f loops, respectively. Since d and f are constants, the complexities of steps 16 and 17 are O(1). The complexity of step 18 is O(1). As a result, the complexity from steps 16 to 18 is O(1). The complexities of steps 19 and 20 are O(m). Therefore, the complexity of Path Adjustment Phase is O($max(m, n)$). In general, the number of sensors is larger than the number of mobile sink. Therefore, the complexity of Path Adjustment Phase will be O(n).

The next task is to analyze the complexity of Cooperation Phase which consists of steps 21-37. Steps 22 and 23 perform the *for*-instruction with m loops. The complexities of steps 22 and 23 are O(m). Step 24 performs the *for*-instruction with c loops. Its complexity is O(c). Inside the loop, the complexities of steps 25-35 are O(1). The complexities of steps 36 and 37 are O(1). As a result, the complexity of Cooperation Phase is O(m). In summary, the overall complexity of the proposed *CORL*^{2} is O(m)+O(n)+ O(m)=O(n). The time complexities of *DEDC* and *CDCA* are O(n^2). In comparison, the proposed *CORL*^{2} requires lower-level of computational complexity, as compared with existing *DEDC* and *CDCA*.

V. Performance Evaluations

This section investigates the performance improvements of the proposed algorithm against the existing algorithms *DEDC* [16] and *CDCA* [19], in terms of the energy consumption, latency time, workload balancing improving index (*WBII*), cooperation index, and network lifetime. In *DEDC*, the monitoring region was divided into several grids. Then the balanced and unbalanced grids were identified. The unbalanced grids were further adjusted and the balanced grids were fine-tuned.

The *CDCA* divided the network field into several partitions and assigned multiple mobile sinks which cooperatively collected data for base station.

The parameter values set in the simulation environment are illustrated. A set of sensors are randomly deployed in a given region with the size 2000 m *2000 m. The initial energy of each sensor node is 200J. The sensing range of each sensor node is set at 40 m, while the communication range of each sensor varies ranging from 20m to 40m. Assume that each sensor knows the information of the mobile sink, including the movement trajectory and arrival time of the mobile sink. The parameter values of the experiments are summarized in Table II. The MATLAB simulator is used as the simulation tool.

TABLE II
Simulation Parameter Setting

Parameters	Values
Node deployment	Random
Region size	2000*2000
The number of sensor nodes in each subregion	2000-6000
The sensing range of each sensor node	40 m
Mobile sink speed	v
Sensor node transmission range	20m-40m
Consumed energy in the transmitter circuit	0.18J
Consumed energy in the receiver circuit	0.1J
The initial energy of each sensor node	200J

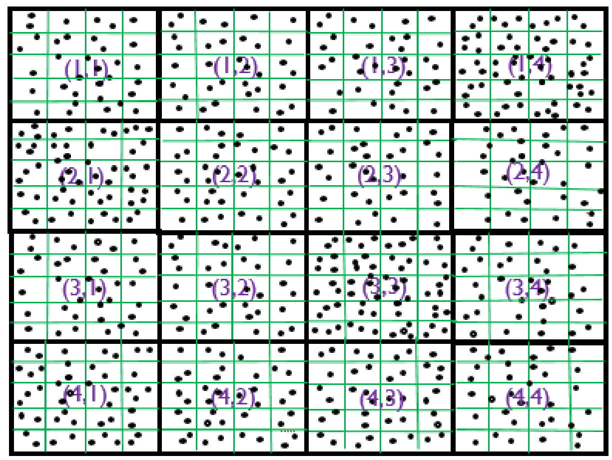
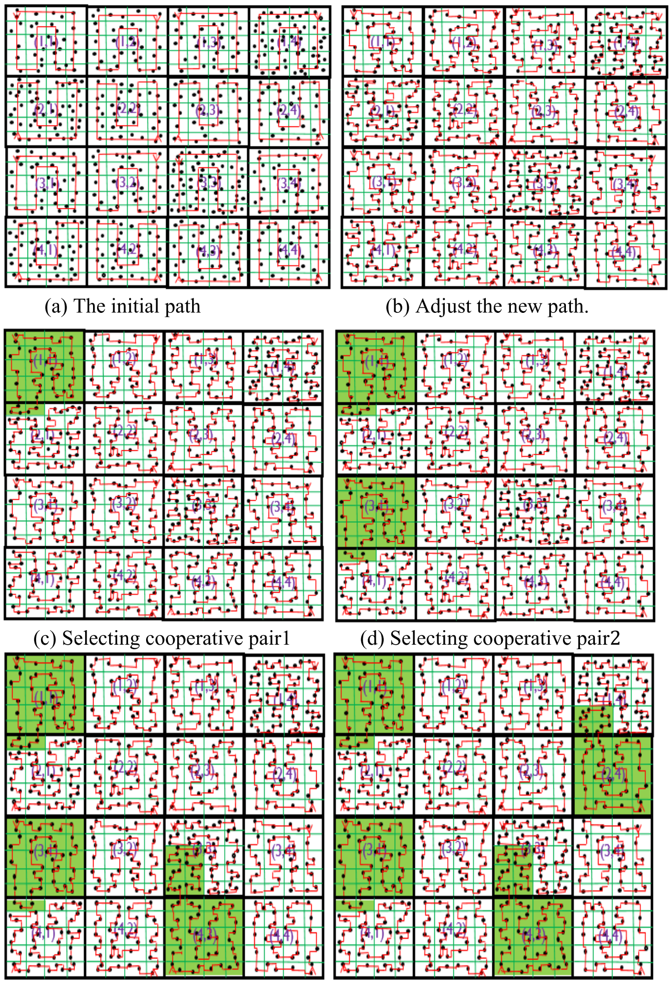


Fig. 6. The scenario considered in the experiments.

Each sensor node generates one data packet in each round. During each round, multiple mobile sinks start from their initial locations and collect data in each grid. Then the mobile sinks go back to the initial locations and then send the collected data to the base station through GSM.

To further investigate the performance of the proposed algorithms, the following illustrates the scenarios considered in the experiments. In the scenario, sensors are randomly deployed over every square grid. Fig. 6 shows one example of the considered scenario.

Recall that the Initial Path Construction Phase of the proposed algorithm aims to partition the subregion into several equal-sized grids and each mobile sink constructs a path passing through every grid in subregions. Then the path adjustment algorithm selects the sensors that have higher remaining energy as anchors, aiming to increase the number of anchors and reduces the number of hops when each sensor forwards data to the anchor. Finally, the mobile sink of each subregion calculates the number of sensors and the path length in its

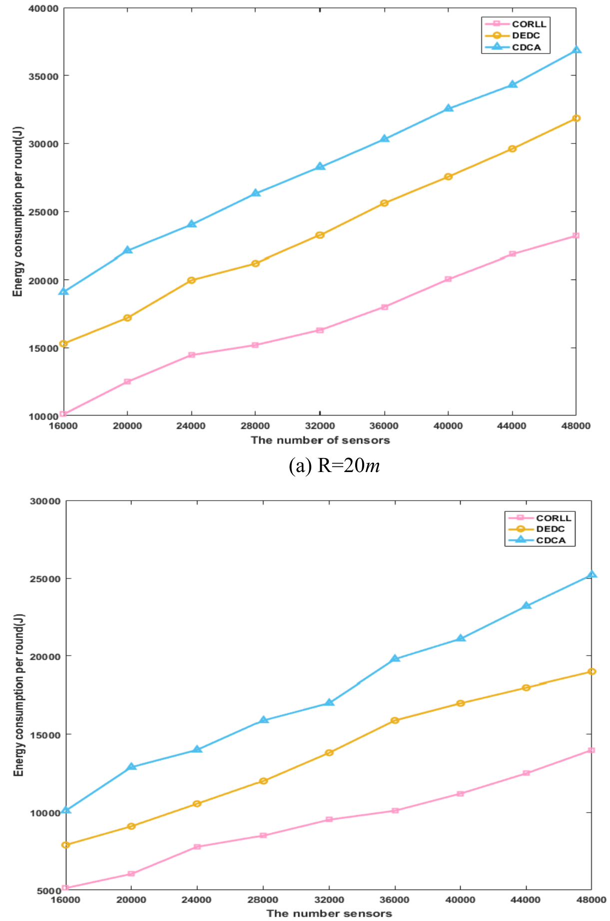


(a) The initial path (b) Adjust the new path.

(c) Selecting cooperative pair1 (d) Selecting cooperative pair2

(e) Selecting cooperative pair3 (f) Selecting cooperative pair4

Fig. 7. Results of operation in the experiments.



(a) R=20m

(b) R=40m

Fig. 8. The comparison of energy consumptions of three approaches.

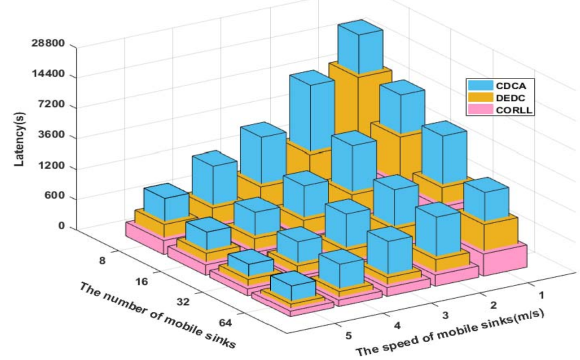


Fig. 9. The comparison of latency of three approaches.

subregion. The Cooperation Phase considers sharing the data collection task of neighboring mobile sinks. Fig.7(a) depicts the initial paths constructed in each subregion. Fig. 7(b) further shows the adjusted path in each subregion. Fig. 7(c), 7(d), 7(e), and 7(f) illustrate the results of cooperation, respectively.

Fig. 8 compares the energy consumption of *CDCA*, *DEDC* and *CORL*^{2}. The number of sensor nodes varies from 16000 to 48000 and the communication range is set at R=20m and R=40m. As shown in Figs. 8(a) and 8(b), the *CORL*^{2} can reduce the energy consumption of sensors. This occurs because the *CORL*^{2} takes into account the density of sensor distribution and the residual energy of sensors. The *CORL*^{2} includes more anchors which help reduce the energy consumption for packet forwarding. The *DEDC* can further adjust the path according to the number of sensors rooted by the anchor. However, it did not consider the remaining energy of the anchors. Some sensors which have low remaining energy but passed by the path are also treated as anchors, leading to network disconnection. The *CDCA* divides the network region into several partitions, assigns each mobile sink to each partition, and then collects data from sensors along with the constructed path. However, the *CDCA* did not take into account the path adjustment according to the sensor density. As result, the *CDCA* includes fewer sensors to play the anchor role and increases the energy consumption of each sensor for packet forwarding. In comparison, the proposed *CORL*^{2} outperforms *CDCA* and *DEDC* schemes in terms of energy consumption.

Fig. 9 compares the latency of the three approaches *CDCA*, *DEDC* and *CORL*^{2}. The number of mobile sinks varies from 8 to 64 and the speed of mobile sinks varies from 1m/s to 5m/s. As shown in Fig. 9, the proposed *CORL*^{2} approach outperforms the *CDCA* and *DEDC* in terms of latency.

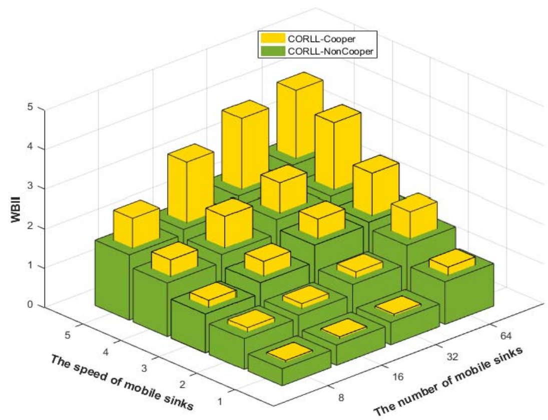


Fig. 10. The comparison of *WBII* of two approaches.

The *CDCA* collects data using mobile sinks in each subregion and then sends those data through neighboring mobile sinks in a multi-hop manner, which increases the overall data collection time. The *DEDC* does not consider the cooperative policy and hence is unable to further reduce the maximal workload, leading to the bottleneck which increases the overall data collection time. On the contrary, the *CORL*^{2} applying cooperative policy can reduce the maximal latency of data collection. This occurs because the mobile sink of each subregion calculates the number of sensors and the path length and then tries to share the data collection task of neighboring mobile sinks for workload balancing. Hence the *CORL*^{2} can further balance the data collection workload between neighboring mobile sinks, reducing the data collection delay.

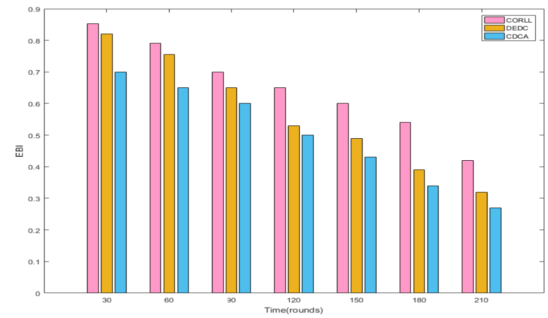
Fig. 10 compares the Workload Balancing Improving Index (*WBII*) of the two approaches, including *not used the cooperative approach* and *used the cooperative approach*. The number of mobile sinks varies from 8 to 64 and the speed of mobile sinks varies from 1m/s to 5m/s. Let n denote the number of subregions. Let $D_{i,j}$ denote the latency time of the subregion $R_{i,j}$ in each round. The *Fairness Index* is defined by (26) as follows:

$$Fairness\ Index = \frac{(\sum_{i=1}^{h}\sum_{j}^{z} D_{i,j})^2}{n \times \sum_{i=1}^{h}\sum_{j}^{z} D_{i,j}^2} \tag{26}$$

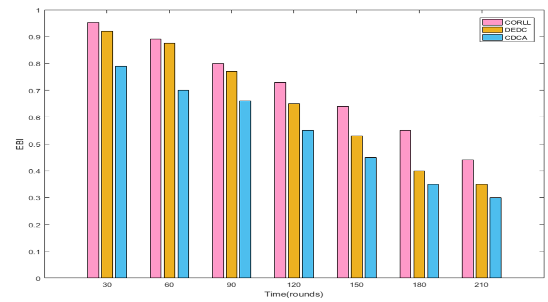
where h and z denote the number of rows and columns, respectively. The *Workload Balancing Improving Index* (*WBII*) which is defined in Equ. (27) measures the improvement of the *Fairness Index* in each time unit.

$$WBII = \frac{Fairness\ Index}{time} \tag{27}$$

Let *CORLL-Cooper* and *CORLL-NonCooper* denote the proposed *CORL*^{2} with and without applying the cooperative policy, respectively. As shown in Fig. 10, the *CORLL-Cooper* approach applying cooperative policy can improve *WBII* and reduce the maximal latency time of data collection. This occurs because the mobile sink of each subregion calculates the number of sensors and the path length in its subregion and then considers sharing the data collection task of neighboring mobile sinks for workload balancing. Hence the *CORLL-Cooper* approach which applies the cooperation policy can further balance the data collection workload between neighboring mobile sinks, reducing the data collection latency and improving the data freshness. On the contrary, the *CORLL-NonCooper* approach without considering the cooperative policy can not further reduce the maximum workload and hence increases the overall data collection time. In comparison, the *CORLL-Cooper* approach outperforms the *CORLL-NonCooper* in terms of $WBII$.



(b) R=20m



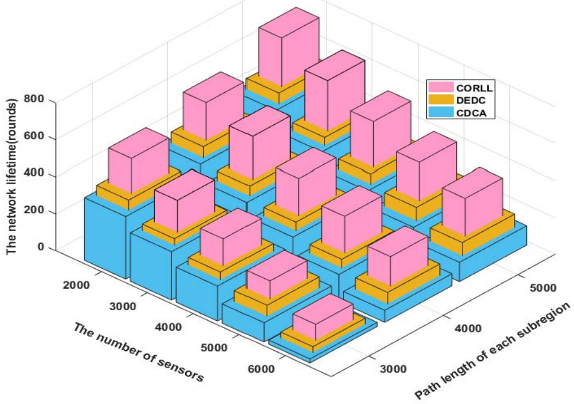
(b) R=40m

Fig. 11. The fairness index of three approaches.

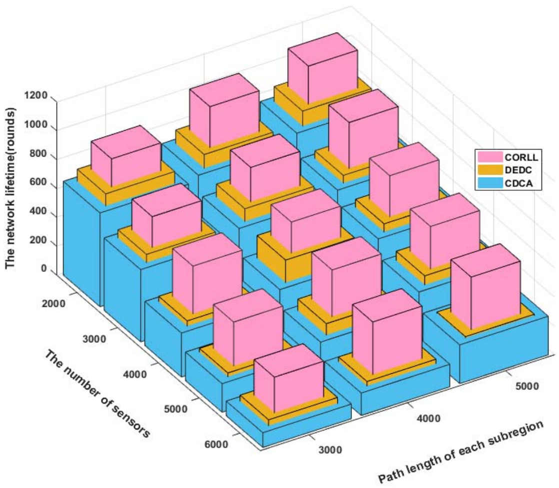
Since anchor node generally consumes more energy than the non-anchor nodes, the lifetime of anchor nodes determines the lifetime of a given WSN. If the forwarding loads of anchor nodes can be balanced, the lifetime can be efficiently prolonged. Fig. 11 investigates the Energy Balance Index (*EBI*) of anchors, which measures the balance degree of energy consumption of anchors. The communication range is set at R=40 and R=20. Let n denote the number of anchors. Let q_i denote the energy consumption of anchor a_i in each round. The Energy Balance Index(*EBI*) is defined by Equ. (28) as follows:

$$EBI = \frac{(\sum_{i=1}^{n} q_i)^2}{n \times \sum_{i=1}^{n} q_i^2} \tag{28}$$

Fig. 11 compares the *EBI* of the three approaches. The existing *CDCA* partitions the whole region into several subregions and assigns each mobile sink to each region. Then the path is constructed for each mobile sink in the subregion. All the nodes that fall in the communication range of the constructed path are treated as anchors. The balancing of workloads among anchors highly depends on the deployment. For an unbalanced deployment, the workload of anchors is not balanced, leading to a small *EBI* value. The existing *DEDC* can further adjust the path according to the number of sensors rooted by the anchor. However, it did not consider



(b) *M=16* mobile sinks



(b) *M=32* mobile sinks

Fig. 12. The comparison of the network lifetimes of three approaches.

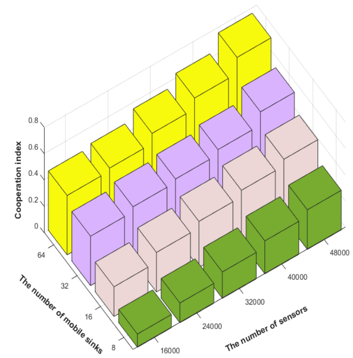


Fig. 13. The cooperation index of the C*ORL*^{2}.

the remaining energy of the anchors. Some sensors which have low remaining energy but passed by the path are also treated as anchors, which leads to low *EBI* value. In the proposed *CORL*^{2}, the mobile sink can visit those anchors which have higher remaining energy. Therefore, the workload of data forwarding can be shared by the anchor nodes which have higher remaining energy. As shown in Fig. 11, the *EBI* of *CORL*^{2} is close to 1. The proposed *CORL*^{2} outperforms *CDCA* and *DEDC* schemes in terms of *EBI* in all cases.

Fig. 12 compares the network lifetimes of the three algorithms by varying the number of sensors ranging from 1000 to 5000 and the path length from 3000m to 5000m in each subregion. Figs. 12(a) and 12(b) measure the network lifetime under 16 mobile sinks and 32 mobile sinks, respectively. The network lifetime is measured from the starting point at which the network runs to the point at which the first sensor node runs out of energy. As shown in Fig. 12, the compared three algorithms have a similar trend that the network lifetime increases with the path length. This occurred because the longer path allows more sensors to be selected as anchors. These anchors share the forwarding workloads of other sensors, hence better balance the forwarding loads of anchors. The *CDCA* divides the network region into several partitions, assigns each mobile sink to each partition, and then collects data from sensors along with the constructed path. However, the *CDCA* did not take into account the path adjustment according to the density of the sensor deployment. Therefore, the path is not efficient, as compared with the other two algorithms. The proposed C*ORL*^{2} outperforms the existing *DEDC* in terms of network lifetime. This occurs because the proposed C*ORL*^{2} adjusts the constructed path and selects more anchors in each grid according to the density and the residual energies of sensors. Compared to existing *DEDC*, the proposed C*ORL*^{2} can better balance the energy consumptions of anchors, reduce the workload of each anchor and hence prolong the network lifetime of the WSNs.

Let cooperation index denote fairness index of cooperation opportunities. The number of mobile sinks, denoted by m, set at 8, 16, 32, and 64 while the number of sensors is ranging from 6000 to 10000. Let n denote the number of experiments. Let ρ_i denote the number of groups that have cooperation in each experiment. The *cooperation index* is defined by Equ. (29):

$$Cooperation\ index = \left(\frac{\sum_i^n \frac{2*\rho_i}{m}}{n} \right) \tag{29}$$

Fig. 13 shows the cooperation index of the proposed C*ORL*^{2} algorithms. In Fig. 13, the cooperation index value increases with the number of sensors and the number of mobile sinks. This occurs because that the increase in the number of sensors might introduce more unbalance opportunities between neighboring subregions. Similarly, the increase of the number of mobile sinks can partition more subregions and hence introduce more cooperative pairs of mobile sinks. As a result, the cooperation index can be increased.

Fig. 14 compares the latency of the three approaches *CDCA*, *DEDC* and *CORL*^{2}. The network size varies from 800m*800m to 2000m*2000m while the speed of mobile sinks varies from 1m/s to 5m/s. As shown in Fig. 14, the compared three algorithms have a similar trend that their latencies are increased with the network size but are decreased with the speed of mobile sink. This occurs because that the path length of mobile sink is increased when the network size grows, resulting in the increase of the latency. On the other hand, when the mobile sink increases its speed, the time required for moving along the constructed path is saved, reducing the latency. In comparison,

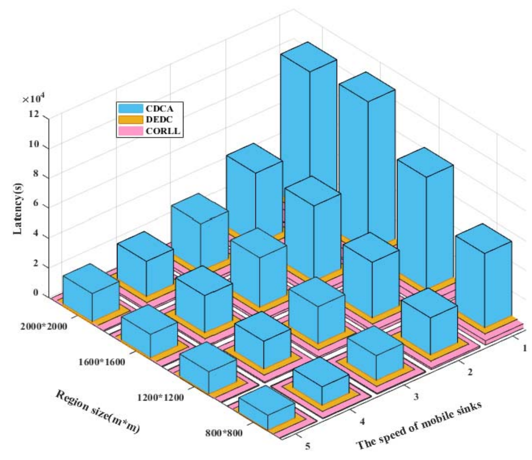


Fig. 14. The comparison of latencies of three approaches.

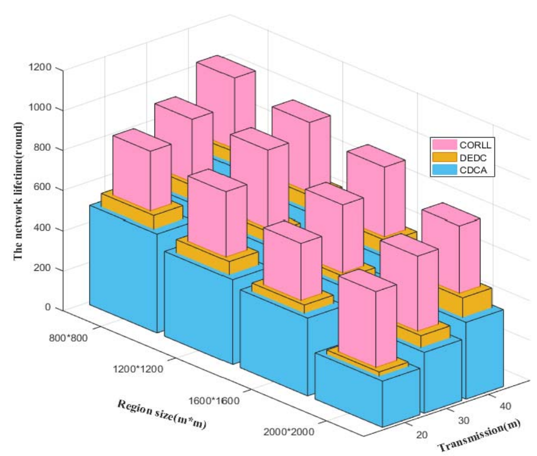


Fig. 15. The comparison of the network lifetimes of three approaches.

the proposed *CORL*^{2} approach outperforms the *CDCA* and *DEDC* in terms of latency. The *CDCA* collects data using mobile sinks in each subregion and then sends those data to the base station through neighboring mobile sinks in a multi-hop manner, which increases the overall data collection time. The *DEDC* does not consider the cooperative policy and hence it is unable to reduce the maximal workload. As a result, the bottleneck mobile sink increases the overall data collection time. On the contrary, the *CORL*^{2} applying cooperative policy can reduce the maximal latency of data collection.

Fig. 15 further compares the network lifetimes of the three algorithms by varying the network size ranging from 800m*800m to 2000m*2000m and the transmission range from 20m to 40m. The number of mobile sinks is set at 16. As shown in Fig. 15, the compared three algorithms have a similar trend that the network lifetime decreases with the network size but increases with the transmission range. This occurred because that a large network can increase the number of sensors and thus increase the energy consumption for packet forwarding, reducing the network lifetime. On the contrary, the increase of transmission range will reduce the number of hop from each sensor to anchor, increasing the network lifetime. In comparison, the proposed *CORL*^{2} outperforms the existing *CDCA* and *DEDC* in terms of network lifetime. This occurs because the proposed C*ORL*^{2} adjusts the constructed path and selects more anchors in each grid according to the density and the residual energies of sensors.

VI. Conclusion

This paper investigates the issue of cooperative data collection using multiple mobile sinks in wireless sensor networks. The paper proposes a cooperative data collection algorithm, called *CORL*^{2}, aiming to prolong the network lifetime and reduce the latency of data collection. The proposed *CORL*^{2} primarily consists of three phases. In the Initial Path Construction Phases, the *CORL*^{2} partitions the monitoring region into several equal-sized subregions according to the number of mobile sinks, and each subregion assigns a mobile sink. Then the *CORL*^{2} further partitions each subregion into several equal-sized grids and construct an initial path along which each mobile sink can move and collect data in each subregion. In the Path Adjustment Phase, the *CORL*^{2} further selects more anchors in each grid according to the density of sensors distribution and the residual energy of sensors and then adjusts the constructed path in each subregion, aiming to better balance the workload of anchors such that the network lifetime can be further prolonged. In the Cooperation Phases, the *CORL*^{2} balances the workload of data collection among mobile sinks such that the data collection tasks can be finished earlier. Performance evaluations show that the proposed *CORL*^{2} outperforms existing schemes in terms of latency, energy consumptions, the workload balancing improving index (*WBII*), energy balance index (*EBI*) as well as network lifetime.

References

[1] N. Dey, A. S. Ashour, F. Shi, S. J. Fong, and R. S. Sherratt, "Developing residential wireless sensor networks for ECG healthcare monitoring," *IEEE Trans. Consum. Electron.*, vol. 63, no. 4, pp. 442–449, Nov. 2017.

[2] P. Chanak and I. Banerjee, "Congestion free routing mechanism for IoT-enabled wireless sensor networks for smart healthcare applications," *IEEE Trans. Consum. Electron.*, vol. 66, no. 3, pp. 223–232, Aug. 2020.

[3] A. L. Bleda, F. J. Fernández-Luque, A. Rosa, J. Zapata, and R. Maestre, "Smart sensory furniture based on WSN for ambient assisted living," *IEEE Sensors J.*, vol. 17, no. 17, pp. 5626–5636, Sep. 2017.

[4] K. Guravaiah and R. L. Velusamy, "Prototype of home monitoring device using Internet of Things and river formation dynamics-based multi-hop routing protocol (RFDHM)," *IEEE Trans. Consum. Electron.*, vol. 65, no. 3, pp. 329–338, Aug. 2019.

[5] T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand, and A. H. Gandomi, "I-SEP: An improved routing protocol for heterogeneous WSN for IoT-based environmental monitoring," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 710–717, Jan. 2020.

[6] M. T. Lazarescu and P. Poolad, "Asynchronous resilient wireless sensor network for train integrity monitoring," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3939–3954, Mar. 2021.

[7] J. Ireneusz, "Graph signal processing in applications to sensor networks, smart grids, and smart cities," *IEEE Sensors J.*, vol. 17, no. 23, pp. 7659–7666, Dec. 2017.

[8] S. Ji, R. Beyah, and Z. Cai, "Snapshot and continuous data collection in probabilistic wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 626–637, Mar. 2014.

[9] R. Velmani and B. Kaarthick, "An efficient cluster-tree based data collection scheme for large mobile wireless sensor networks," *IEEE Sensors J.*, vol. 15, no. 4, pp. 2377–2390, Apr. 2015.

[10] R. Sugihara and R. K. Gupta, "Optimal speed control of mobile node for data collection in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 1, pp. 127–139, Jan. 2010.

[11] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling with dynamic deadlines," *IEEE Trans. Mobile Comput.*, vol. 6, no. 4, pp. 395–410, Apr. 2007.

[12] C.-Y. Chang, C.-C. Lin, C. Shang, I.-H. Chang, and D. S. Roy, "DBDC: A distributed bus-based data collection mechanism for maximizing throughput and lifetime in WSNs," *IEEE Access*, vol. 7, pp. 160506–160522, Oct. 2019.

[13] H. Salarian, K.-W. Chin, and F. Naghdy, "An energy-efficient mobile-sink path selection strategy for wireless sensor network," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2407–2419, Jun. 2014.

[14] Y.-H. Yang, T. Lin, B.-H. Liu, S.-I. Chu, C.-Y. Lien, and V.-T. Pham, "An efficient mobile sink scheduling method for data collection in wireless sensor networks," in *Proc. Int. Conf. Syst. Sci. Eng. (ICSSE)*, Ho Chi Minh City, Vietnam, Jul. 2017, pp. 554–557.

[15] W. Wen, S. Zhao, C. Shang, and 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.

[16] W. Wen, C. Shang, C.-Y. Chang, and D. S. Roy, "DEDC: Joint density-aware and energy-limited path construction for data collection using mobile sink in WSNs," *IEEE Access*, vol. 8, pp. 78942–78955, May 2020.

[17] X. Liu, T. Qiu, X. Zhou, T. Wang, L. Yang, and V. Chang, "Latency-aware path planning for disconnected sensor networks with mobile sinks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 350–361, Jan. 2020.

[18] A. C. Santos, C. Duhamel, and L. S. Belisário, "Heuristics for designing multi-sink clustered WSN topologies," *Eng. Appl. Artif. Intell.*, vol. 50, pp. 20–31, Apr. 2016.

[19] A. Abdullah, A. Khaled, and A. Maamoun, "Data collection algorithm for wireless sensor networks using collaborative mobile elements," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 3, p. 2131, Jun. 2019.

[20] D. Van Le, H. Oh, and S. Yoon, "HiCoDG: A hierarchical data-gathering scheme using cooperative multiple mobile elements," *Sensors*, vol. 14, no. 12, pp. 24278–24304, Dec. 2014.

Weimin Wen received the M.S. degree from the Xi'an University of Science and Technology in 2008. She is currently a Senior Laboratory Teacher with the School of Computer and Information Engineering, University of Chuzhou, China. Her current research interests include wireless sensor networks and healthcare.

Chih-Yung Chang (Member, IEEE) received the Ph.D. degree in computer science and information engineering from National Central University, Taiwan, in 1995. He is currently a Full Professor with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei, Taiwan. His current research interests include the Internet of Things, wireless sensor networks, artificial intelligence, and deep learning. He has served as an associate guest editor for several SCI-indexed journals, including *International Journal of Ad Hoc and Ubiquitous Computing* (IJAHUC) from 2011 to 2019, *International Journal of Distributed Sensor Networks* (IJDSN) from 2012 to 2014, *IET Communications* in 2011, *Telecommunication Systems* (TS) in 2010, *Journal of Information Science and Engineering* (JISE) in 2008, and *Journal of Internet Technology* (JIT) from 2004 to 2008.

Shih-Jung Wu received the M.S. and Ph.D. degrees from the Department of Computer Science and Information Engineering, Tamkang University, Taiwan, in 2001 and 2006, respectively. He is a Professor with the Department of Computer Science and Information Engineering and the Dean of Student Affairs with Tamkang University. He is also the host of several important research projects. He has published many excellent articles in international journals. His current research interests include parallel computing, data mining, wireless networks, and artificial intelligence. He has received many teaching, tutoring, and professional competition awards. Prof. Wu is an Outstanding Academic Researcher in Taiwan.

Diptendu Sinha Roy received the Ph.D. degree in engineering from the Birla Institute of Technology at Mesra, India, in 2010. In 2016, he joined the Department of Computer Science and Engineering, National Institute of Technology (NIT) Meghalaya, India, as an Associate Professor, where he has been serving as the Chair of the Department of Computer Science and Engineering since January 2017. Prior to NIT Meghalaya, he was with the Department of Computer Science and Engineering, National Institute of Science and Technology, Berhampur, India. His current research interests include software reliability, distributed and cloud computing and IoT, specifically applications of artificial intelligence, and machine learning for smart integrated systems. Dr. Roy is a member of the IEEE Computer Society.