

Fig. 1. Spatial and temporal distances will impact the data accuracy in WSNs.

[25], [26], [27], [28], [29], [30], [31], [32]. The offline recharging approaches [13], [14], [15], [16], [17], [18], [19], [20], [21], recharge the sensors in a periodical manner. These approaches ignored the sudden changes in the network. That is to say, the predetermined charging schedule of MC is not optimal since it cannot adapt to applications with the high energy demand of sensors. To solve the issues faced by offline recharging approaches, the online recharging approaches [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32] were proposed. They allowed the sensors to send the recharging requests. Then, the MC will make the decision based on the real-time information of the requested sensors. The online approaches can adapt to networks with different energy requirements. The most important issue in online recharging approaches is to select the best charging candidate among the sensors that have sent the charging requests. There have been several online charging approaches proposed in recent years. The GSA-based mechanism [30] adopted the gravitational search algorithm and fitness function to select the charging candidate, while MERSH [31] considered the charging latency to recharge the sensors. Another study, ETLBO [32] proposed an algorithm based on the teaching-learning-based optimization method. Different from the related studies, this article aims to propose the recharging approach by considering the spatial quality contribution and adjusting the sensing time sequence of the requested sensors, aiming to improve the data accuracy of the network.

WSN consists of tiny sensors, which sense data periodically according to their schedule and forward it to the sink. In WSN, if any user sent a data query, the sink would check its database and report the data according to the data query of the user. The data query consists of data-centric operations, such as information about the location and temperature. As shown in Fig. 1, assume that the user sent a data query 1 “what is the temperature in location X at time 10:45 A.M.?” to the sink. The sink will respond to the temperature according to its database. As shown in Fig. 1, there is no sensor deployed in location X. Therefore, the sensing data of the nearest

sensor are used to represent the data of location X. As shown in Fig. 1, sensors s_j and s_k are closer to the location X. Compared to the sensor s_k , we have $(X, s_j) < d(X, s_k)$. Therefore, the sink will report the sensing data collected by the sensor s_j at 10:45 A.M. to the data query 1, resulting in better data accuracy. This implies that the data accuracy reported to the user is decreased with the spatial distance between the query location and the closest sensor location.

On the other hand, assume there is another data query 2 “what is the temperature in location Y at time 10:35 A.M.?” to the sink. As shown in Fig. 1, location Y is in the sensing range of the sensor s_i . Therefore, the sensing data of the sensor s_i are used to respond to the data query 2. Assume sensor s_i scheduled sensing time slots at 10:30 A.M., 10:45 A.M., and 11:00 A.M., as shown in the bottom part of Fig. 1. Since the sensor s_i is not sensing at 10:35 A.M., the sink should report the closest sensing time information to the sink. As shown in Fig. 1, the sink will report the sensing data at 10:30 A.M. because it is the closest sensing time compared to 10:45 A.M. If the sink reports the sensing data of 10:45 A.M., the reported data accuracy is very low. According to the above two examples, if the spatial and temporal distances are larger, the reported data accuracy will be lower, which degrades the data accuracy of the network.

To improve the data accuracy of the network, this article proposed a *Recharge Scheduling Algorithm for Maximizing Spatial and Temporal Data Accuracy* (RS-STQ). The proposed RS-STQ can better adapt to the dynamic energy consumption of sensors in WRSNs. All the recharging requests are stored in the service pool of the MC. This article proposed a recharging schedule for MC based on the spatial quality contribution of each requested sensor, aiming to improve the data accuracy of the network. During the network initialization, the MC constructs a Hamiltonian path by considering the set of recharging candidates. When each sensor on the path is recharged, the MC checks its service pool to determine whether any sensor closer to its path has requested a recharging request. The MC will include the newly requested sensor if the spatial quality contribution is higher than the sensors on the path. In addition, the energy management strategy for each sensor is proposed, aiming to adjust the sensing time sequence of each sensor for maximizing the data accuracy. The contributions of the proposed RS-STQ are itemized in the following.

A. Recharging Request Threshold Value of Each Sensor Is Adaptively Determined

In the proposed RS-STQ algorithm, each sensor determines its recharging request threshold value by considering its discharging rate and the total time required by the MC to recharge it. Therefore, each sensor can better utilize the energy of MC for recharging.

B. Dynamic Recharging Schedule

The recharging schedule of the RS-STQ algorithm is dynamically adjusted. This study considered the data accuracy benefit of the recharging requested sensors. Therefore, all

the requested sensors whose data accuracy is higher will be recharged with priority. This strategy improves the data accuracy of the network.

C. Maximizing the Data Accuracy of the Network Based on the Spatial and Temporal Qualities

In the proposed RS-STQ algorithm, the data accuracy is measured from the aspects of space and time, which is different from the existing studies [35], [36]. The spatial quality of the sensors is considered to construct the recharging schedule, while the sensing time sequence of the sensors is adjusted to improve the temporal quality, aiming to maximize the data accuracy.

The remainder of this article is organized as follows. The offline and online recharging approaches are reviewed in Section II. Section III details the considered environment and objective function. Section IV details the RS-STQ algorithm. Section V compares the experimental results, while the conclusion is presented in Section VI.

II. RELATED WORKS

The existing studies related to recharging approaches are reviewed in this section. These studies were partitioned into two types: offline [13], [14], [15], [16], [17], [18], [19], [20], [21] and online [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32] recharging approaches.

A. Offline Recharging Approaches

In these types of approaches, the MC traverses in the predefined trajectory to finish its recharging tasks. These approaches mainly focused on planning the optimal path for the MC. The study [13] proposed an f -approximate scheduling scheme by considering the energy consumption of nodes. Another study [14] developed a periodic charging schedule by considering the relationship between the charging cycle of the MC and the lifetime of sensor nodes. Chen et al. [15] focused on designing the path of the MC and collaboration between them to reduce the number of MCs.

Different from the abovementioned studies, another study [16] developed a partial energy charging model. In addition, the shortest charging path for MC is constructed by ensuring that the sensor lifetime is maximized. To minimize the path length of the MC, another study [17] assumed that all the sensors that fall in the transmission range of the MC could be recharged simultaneously. The study [18] constructed the path of the MC by assuming that the energy consumption rate of sensors is constant, which is not suitable for many real applications.

Another study [19] proposed a periodic charging scheme based on a genetic algorithm to minimize the dead nodes. This study assumed that sensors are partially recharged in each charging round. However, partial recharging schemes lead to longer path length, which consumes a lot of energy for the movement of the MC. Another study [20] partitioned the grids and considered the weight of each grid to construct the recharging path. In addition, cost-effective and fairness

recharging approaches were proposed. The study [21] proposed an intelligent charging scheme to optimize the system utility, aimed to improve data accuracy. This study considered the impact of sensing nodes with different data accuracy. If multiple sensing nodes cover the same area, the system will only select the data with the highest quality value to avoid repeated calculations for the utility. However, in this study, the spatial and temporal qualities of the sensors were ignored. In studies [13], [14], [15], [16], [17], [18], [19], [20], and [21], the schedule of the MC is static, which ignored the new recharging requests from the sensors. Although the new requested sensors were closer to the MC location, it will not consider those sensor requests. They will be visited in the next cycle, which leads to a longer path.

B. Online Recharging Approaches

In these approaches, the MC timely considers the recharging requests of the sensors. The study [22] developed an algorithm based on the remaining energy and energy consumption rate of sensors. The node insertion and deletion algorithms were developed to remove the low-efficient nodes, respectively.

Another study [23] applied the game theory in WRSNs. In this study, the MC decided the charging sequence based on the location and charging deadlines of the charging requested nodes. However, data accuracy is overlooked in this study. In the study [24], the sensor nodes were prioritized and served based on their contribution to the monitoring tasks and deleted the sensor nodes, which degrades the charging performance. However, the partial charging schemes can lead to the ping-pong movement of the MC.

The study [25] designed an adaptive threshold value based on the maximum traveling time of the MC and the consumption rate of sensors. Tomar et al. [26] proposed an architecture for the sensor nodes based on fuzzy logic. This study considered the remaining energy of sensor nodes, node density, and distance to the MC to make the recharge scheduling decision. However, it predefined the threshold value of each sensor. To overcome this issue, another study [27] designed the threshold value by considering the energy consumption rate, average Euclidean distance between any two nodes to calculate the moving distance of the MC, and the average charging time. This study focused on multinode energy transfer by considering the multiple MCs. Initially, the network is partitioned and distributed to the MCs to balance their workloads. When the MC received the charging requests from the sensors of its corresponding region, fuzzy logic is adopted to determine the schedule of the MC.

The study [28] adopted a reinforcement learning technique to construct the path of the MC. In addition, this study focused on charging the sensors, aiming to maximize the sum of charging rewards collected by the MC. Kumar and Mukherjee [29] considered the energy depletion of MCs during their charging tours and proposed a vehicle-assisted framework for charging the sensors in WRSNs. The study [30] proposed a linear programming formulation for scheduling the MC. Another study [31] proposed an energy replenishment scheme, aiming to improve the charging performance in WRSNs. This study

TABLE I
COMPARISON OF THE PROPOSED AND RELATED WORKS

Related works	Adaptive threshold value	Spatial and temporal qualities of the sensors	Adjust the sensing time sequence	Considering the data accuracy of the network
[22]	×	○	×	×
[23]	×	×	×	×
[24]	×	×	×	×
[25]	○	○	×	×
[26]	×	×	×	×
[27]	○	×	×	×
[28]	×	×	×	×
[29]	×	×	×	×
[30]	×	○	×	×
[31]	×	×	×	×
[32]	×	○	×	×
[34]	×	×	○	×
The proposed algorithm	○	○	○	○

adjusted the charging duration of sensors. Zhao et al. [32] proposed charging scheduling by considering the service cost and energy replenishment utility, aiming to improve the charging efficiency. Although the studies [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], and [32] dynamically constructed the recharging path of the MC, none of them considered the data accuracy of the network.

Table I gives the comparison between proposed and related works. In the proposed RS-STQ algorithm, by considering the spatial quality contribution of the requested sensors, the insertion and deletion conditions are proposed for reconstructing the recharging path dynamically. In addition, most of the studies [22], [23], [24], [26], [28], [29], [30], [31], [32] assumed that the threshold value is fixed. A recharging request threshold, which is too loose or tight, may have negative effects on the WRSNs [33]. In the proposed RS-STQ algorithm, each sensor determines an appropriate threshold value locally by considering its discharging rate and the total time required for the MC to move and recharge all the requested sensors in the service pool.

III. SYSTEM MODEL AND PROBLEM STATEMENT

This section first introduces the environment of the considered WRSNs. Then, the problem statement of this study is discussed.

A. Network Environment

Let R denote the considered monitoring region. This article assumes that there are n static sensors, represented by $S = \{s_1, s_2, \dots, s_n\}$, which have been randomly deployed in R . The sensor is responsible for collecting data and transmitting its readings to the static sink via multihop forwarding. The sensing and communication operations executed by each sensor consume energy from the battery. When the battery power is lower than a certain level, the sensor needs to be recharged. There is one MC in the network to recharge the sensor nodes.

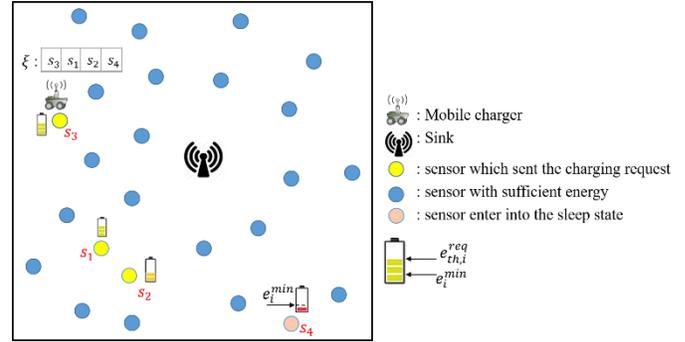


Fig. 2. Considered network environment.

The stationary sink will support the energy of MC, which is initially located at the sink location. When the MC receives the charging requests from sensors, it will move from the sink to the requested sensors for providing the charging service and, finally, return to the sink. Since the MC might receive many requests from the sensors and the recharging task is time-consuming, some requested sensors might not be timely recharged. Therefore, some sensors may enter into a sleeping state due to energy exhaustion, resulting in data accuracy loss (DAL). To obtain the maximal data accuracy, this article aims to develop an efficient energy recharging schedule for MC and an energy management strategy for each sensor. The developed recharging schedule for the MC aims to maximize the working sensors, while the energy management strategy aims to adjust the sensing rate of each sensor for maximizing the data accuracy.

The lifetime of each sensor depends on various parameters, such as the time, when the sensor sends the requests and the amount of recharged energy. These parameters related to the recharging model can determine how well the sensors are recharged by the MC. To completely describe the recharging-related parameters and exhibit their relations, the sensor recharging model is presented in Section II-B.

B. Sensor Recharging Model

The battery capacity of each sensor is denoted by E_S^{\max} . Since the forwarding load and sensing rate of each sensor are different, each sensor s_i has its discharging rate e_i^{disch} . In this article, there are two threshold values for each sensor, including the *recharging request threshold* and *sleep threshold*, which are denoted by $e_{\text{th},i}^{\text{req}}$ and e^{min} , respectively. If the remaining energy of the sensor s_i reaches the threshold value $e_{\text{th},i}^{\text{req}}$, the sensor s_i sends a *recharging request* to the sink node. Whenever the MC finishes recharging one sensor, it will update its service pool by querying the sink node. The other threshold value e^{min} is predefined for all the sensors, and it is the minimum energy required for basic operations. Therefore, if the remaining energy of the sensor s_i reaches the threshold value e^{min} , the sensor s_i will switch to a sleep state. The MC maintains the charging service pool denoted by ζ , and all the requests are stored in ζ . The considered network scenario is depicted in Fig. 2.

Let $T = \{t_1, t_2, \dots, t_j, \dots, t_m\}$ denote the total time for measuring the data accuracy of region R . Let r_{crg} denote the recharging radius of each sensor, which is the same for all sensors. Let l_M and l_i denote the current locations of MC and sensor s_i , respectively. Let $d(l_i, l_M)$ denote the distance between s_i and MC. Let $\zeta_{i,j}^{\text{crg}}$ be a Boolean variable indicating whether MC is located in the recharging range of the sensor s_i at time slot t_j . The value of $\zeta_{i,j}^{\text{crg}}$ can be calculated as shown in the following equation:

$$\zeta_{i,j}^{\text{crg}} = \begin{cases} 1, & d(l_i, l_M) \leq r_{\text{crg}} \\ 0, & d(l_i, l_M) > r_{\text{crg}}. \end{cases} \quad (1)$$

Let E_i^{crg} denote the energy of the sensor s_i charged by MC at a one-time slot. The energy of sensors is evaluated in voltage. Let $E_i^{\text{tot_crg}}$ be the total charged energy of the sensor s_i . The value of $E_i^{\text{tot_crg}}$ can be derived by the following equation:

$$E_i^{\text{tot_crg}} = \sum_{t_1}^{t_m} \zeta_{i,j}^{\text{crg}} \times E_i^{\text{crg}} \quad \forall t_j \in T, \quad s_i \in S. \quad (2)$$

Let $\mu_{i,t}^{\text{time}}$ denote the Boolean variable representing whether or not the sensor s_i is sensing at the current time slot t . That is,

$$\mu_{i,t}^{\text{time}} = \begin{cases} 1, & \text{if sensor } s_i \text{ is sensing at time } t \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Let $E_{i,j}^{\text{con}}$ represent the energy consumed by the sensor s_i at time slot t_j for executing both the sensing and communication tasks. Let $E_{i,j}^{\text{rem}}$ denote the remaining energy of the sensor s_i at time slot t_j . The value of $E_{i,j}^{\text{rem}}$ can be evaluated, as shown in the following equation:

$$E_{i,j}^{\text{rem}} = E_{i,j-1}^{\text{rem}} + E_{i,j}^{\text{crg}} * \zeta_{i,j}^{\text{crg}} - E_{i,j}^{\text{con}} * \mu_{i,t}^{\text{time}}. \quad (4)$$

After the calculation of recharged energy and the remaining energy of each sensor, the data accuracy provided by each sensor should be determined. The data accuracy of the monitoring region consists of both space and time qualities. To determine the data accuracy of each space point at a certain time slot, the data accuracy model is presented in Section II-C.

C. Data Accuracy Model

The data accuracy of the network depends on the quality of sensing data of all the sensors in the network. Every data have its quality. The data accuracy is measured with accuracy. The accuracy is the difference between the reporting data to the actual data. The time of reporting data may not be the time of sensing data. Therefore, the word “how well” means the accuracy of the reporting data to the user query. That is, the data accuracy is measured by how accurately the sensing data can be represented at each point of space and time.

The data accuracy will be measured from the aspects of space and time. The following explains the spatial quality calculations. Assume that there is a user who wants to query the temperature of location “y” at time t . Assume that the closest sensor to location “y” is deployed at location “x.” Assume that the sensor performs a sensing operation and

detects the temperature of 37 °C at the time t . It is appreciated to use temperature = 37 °C to estimate the temperature of the location “y.” Since the distance of “x” and “y” is small, the reporting temperature of location “x” is much close to the actual temperature of location “y.” Therefore, the data accuracy of the location “y” is high. However, if location “x” is far from location “y,” the spatial quality is low since the difference between the reporting temperature and the actual temperature is big. In case there is no sensor deployed at a space point “y,” the data accuracy of “y” depends on the distance between “y” and “x,” where “x” is the sensor location closest to “y.” From the explanation of the abovementioned two cases, it indicates that the data accuracy is impacted by the spatial quality.

In addition, the data accuracy is also impacted by the temporal quality. The WSN will report the data to the user according to the user query. Assume that there is a user query about the temperature on Tuesday at 9:00 A.M. However, the sensor did not sense at 9:00 A.M., and its last sensing time was 8:00 A.M. The WSN will report that the temperature is 33 °C, but the actual temperature is 35 °C at that time (8:00 A.M.). Therefore, the data loss is 2 °C. The data accuracy is defined as the reciprocal of data loss. That is to say, if the data loss increases, the data accuracy will be decreased. Data accuracy depends on the accuracy of the reporting data (temperature) to the user query at a given time t . If the WSN reports the temperature at time t_1 to the user query, there is data loss. The data accuracy is related to data loss. Let $d(x, y)$ denote the distance between the locations “x” and “y.” Let ρ^{space} and ρ^{time} denote the loss rate of spatial quality per unit distance and temporal quality per unit of time, respectively. Let $d(t, t_1)$ denote the time difference between time slots t and t_1 . The following two conditions define the data accuracy in the aspect of spatial and temporal qualities:

$$\text{Spatial Quality} = \begin{cases} \rho^{\text{space}} \text{ is low,} & \text{if } d(x, y) \text{ is closer} \\ \rho^{\text{space}} \text{ is high,} & \text{if } d(x, y) \text{ is far} \end{cases}$$

$$\text{Temporal Quality} = \begin{cases} \rho^{\text{time}} \text{ is low,} & \text{if } d(t, t_1) \text{ is smaller} \\ \rho^{\text{time}} \text{ is high,} & \text{if } d(t, t_1) \text{ is larger.} \end{cases}$$

According to the above-discussed conditions, if the spatial and temporal qualities are higher, the data accuracy is also higher. Let $p \in R$ denote any point in the monitoring region. Let $\mu_{p,i}^{\text{space}}$ denote a Boolean variable indicating whether or not point p falls in the sensing range of the sensor s_i . The value of $\mu_{p,i}^{\text{space}}$ is shown in the following equation:

$$\mu_{p,i}^{\text{space}} = \begin{cases} 1, & \text{if } p \text{ falls in the sensing range of } s_i \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

If point p falls in the sensing range of the sensor s_i , which is sensing at the current time slot t , the spatial and temporal qualities of that point p are one. Recall that $\mu_{i,t}^{\text{time}}$ denotes the Boolean variable representing whether or not the sensor s_i is sensing at the current time slot t . On the other hand, assume that point p falls in the sensing range of the sensor s_i , but s_i is not sensing at the current time slot t . Assume that the most

recent sensing of the sensor s_i is at the time slot t_1 . Therefore, the data accuracy at the time slot t_1 is used to represent the data accuracy of point p .

Recall that $d(t, t_1)$ denotes the time difference between time slots t and t_1 . The temporal quality of point p is $(1 - \rho^{\text{time}})^{d(t, t_1)}$. Let $\lambda_{p, t}$ denote the spatial-temporal quality of point p at time slot t . The value of $\lambda_{p, t}$ is calculated, as shown in the following equation:

$$\lambda_{p, t} = \begin{cases} 1, & \text{if } \mu_{p, i}^{\text{space}} = 1, \mu_{i, t}^{\text{time}} = 1 \\ (1 - \rho^{\text{time}})^{d(t, t_1)}, & \text{if } \mu_{p, i}^{\text{space}} = 1, \mu_{i, t_1}^{\text{time}} = 1 \end{cases} \quad (6)$$

where ρ^{time} denotes the loss rate of temporal quality per unit of time.

In the following, the spatial and temporal qualities of point p are further defined for $\mu_{p, i}^{\text{space}} \neq 1$. Let $d(p, s_i)$ denote the distance between point p and sensor s_i . If point p does not fall in the sensing range of the sensor s_i , which is sensing at the current time slot t , the spatial quality of that point p is $(1 - \rho^{\text{space}})^{d(p, s_i)}$, where ρ^{space} denotes the loss rate of spatial quality per unit distance. The spatial quality is calculated based on the distance between point p and the sensor s_i . Besides, the temporal quality is calculated based on the time difference between the current time slot t and the most recent sensing of the sensor s_i . Therefore, the spatial-temporal quality of point p if $\mu_{p, i}^{\text{space}} \neq 1$ is shown in the following equation:

$$\lambda_{p, t} = \begin{cases} (1 - \rho^{\text{space}})^{d(p, s_i)}, & \text{if } \mu_{p, i}^{\text{space}} \neq 1, \mu_{i, t}^{\text{time}} = 1 \\ (1 - \rho^{\text{space}})^{d(p, s_i)} \times (1 - \rho^{\text{time}})^{d(t, t_1)}, & \text{if } \mu_{p, i}^{\text{space}} \neq 1, \mu_{i, t_1}^{\text{time}} = 1. \end{cases} \quad (7)$$

Since the environmental data are changed with time and distance, the data accuracy is measured according to the spatial and temporal distances between the reported data and the actual data. Let the user query be “what is the temperature of location X at time t ? Assume sensor s_i that is located at location Y , and it performs the sensing operation at the time t_1 , and the query system reports the data collected by the sensor s_i to answer this query.

Let $\mu_{p, i}^{\text{space}}$ denote a Boolean variable indicating whether or not point p falls in the sensing range of the sensor s_i . Let $\mu_{i, t}^{\text{time}}$ denote the Boolean variable representing whether or not the sensor s_i is sensing at the current time slot t . Since the data accuracy decreased with the spatial distance and time distance between the actual data and reported data, the accuracy can be derived by (6) and (7). In a special case, if the sensor s_i performs the sensing operation at time t and the sensor s_i is located at location X , the accuracy formula will be simplified as 1. On the other hand, in case the sensing time t_1 is far away from time point t and the sensor is located far away from the query location X , the formula will be zero. Therefore, the results of (6) and (7) are between $[0, 1]$. The data reporting error can be calculated by $1 - \lambda_{p, t}$.

Fig. 3(a) depicts the deployment of the sensor set S in the monitoring region. Let notation $s_i.loc$ denote the location of the sensor s_i . In Fig. 3(a), the region R has been partitioned into several Voronoi cells according to the location $s_i.loc$ of

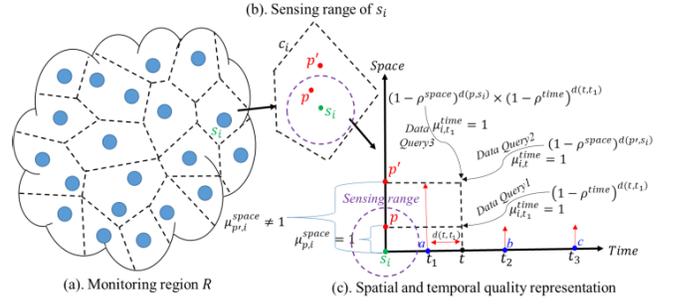


Fig. 3. Example to represent the spatial and temporal qualities. (a) Monitoring region R . (b) Sensing range of s_i . (c) Spatial and temporal quality representation.

each sensor $s_i \in S$. Let c_i denote the Voronoi cell of the sensor s_i . As shown in Fig. 3(b), the polygon which surrounds the sensor s_i is the Voronoi cell c_i . Therefore, the data collected by the sensor s_i will be used to represent the data of any point $p \in c_i$. As shown in Fig. 3(b), the violet dotted lines denote the sensing range of the sensor s_i . There are two points denoted as points p and p' , where p falls, but p' does not fall in the sensing range of the sensor s_i , respectively. The spatial and temporal qualities representations of sensor s_i to points p and p' are shown in Fig. 3(c).

As shown in Fig. 3(c), the sensor s_i is sensing at time slots t_1 , t_2 , and t_3 . There are three different cases to determine the spatial and temporal qualities of the sensor s_i to points p and p' . In the first case, assume that *Data Query 1* is related to point p at the current time t , and point p falls in the sensing range of s_i . However, the sensor s_i does not execute the sensing operation at time t . Therefore, the closest sensing time will be used to represent the data accuracy at time t . As shown in Fig. 3(c), the closest sensing time slots to the current time t are t_1 and t_2 . Comparing t_1 and t_2 , the time slot t_1 is closer to the time point t i.e., $\mu_{i, t_1}^{\text{time}} = 1$. Therefore, the sensor data collected by s_i at time t_1 is used to represent the sensing data of point p at time t , that is, the value of $\lambda_{p, t} = (1 - \rho^{\text{time}})^{d(t, t_1)}$. If the WSN reports the sensing data of t_1 (i.e., a), it will result in good data accuracy. On the other hand, if the WSN reports the sensing data of t_2 (i.e., b), it will result in bad accuracy quality. If the difference between a and b is larger, the reported data accuracy for *Data Query 1* at time point t is lower. In the second case, assume that there is *Data Query 2* that concerns the temperature of point p' at current time t , where $\mu_{p', i}^{\text{space}} \neq 1$ and $\mu_{i, t}^{\text{time}} = 1$, and the value of $\lambda_{p', t} = (1 - \rho^{\text{space}})^{d(p', s_i)}$. Finally, in the third case, assume that there is *Data Query 3* about point p' , where $\mu_{p', i}^{\text{space}} \neq 1$ and $\mu_{i, t_1}^{\text{time}} = 1$, and the value of $\lambda_{p', t} = (1 - \rho^{\text{space}})^{d(p', s_i)} \times (1 - \rho^{\text{time}})^{d(t, t_1)}$.

Finally, it should be noticed that all the points $p \in R$ have spatial and temporal qualities, no matter whether or not it belongs to the sensing range of any sensor and whether or not the sensor is sensing at the current time slot.

Let Q denote the total data accuracy of all the points in the monitoring region $p \in R$ over all the time instants T . The value of Q is calculated according to the following equation:

$$Q = \sum_{p \in R} \sum_{t \in T} \lambda_{p, t}. \quad (8)$$

Let \mathbb{S} represent a possible schedule. The set of possible schedules is denoted by Φ . Based on the schedule \mathbb{S} , the proposed algorithm aims to maximize the data accuracy of all the points $p \in R$ over all the time instants T . Let \mathbb{S}^{best} denote the best schedule. The objective function of this study is shown in (9).

Objective Function:

$$\mathbb{S}^{\text{best}} = \arg \max_{\mathbb{S} \in \Phi} \sum_{p \in R} \sum_{t \in T} \lambda_{p,t}. \quad (9)$$

The first constraint is the sensor battery constraint, which is shown in (10). This constraint guarantees that $E_{i,j}^{\text{rem}}$ of each sensor cannot be smaller than e^{min} and greater than E_S^{max} .

1) Sensor Battery Constraint:

$$e^{\text{min}} \leq E_{i,j}^{\text{rem}} \leq E_S^{\text{max}} \quad \forall t_j \in T \quad \forall s_i \in S. \quad (10)$$

Another constraint is the charging time constraint, as shown in (11). Let T_i^{crg} represent the time required to recharge the sensor s_i . If $\zeta_{i,j}^{\text{crg}} = 1$ and it is fully charged, then $E_i^{\text{crg}} = E_S^{\text{max}} - E_{i,j}^{\text{rem}}$. That is to say, the additional energy of the sensor s_i obtained from MC cannot be larger than $\min(E_S^{\text{max}} - e^{\text{min}}, E_S^{\text{max}} - E_{i,j}^{\text{rem}})$. Let e_M^{charge} denote the recharging rate of MC. The charging time constraint is shown in (11).

2) Charging Time Constraint:

$$0 \leq T_i^{\text{crg}} \leq \frac{\min(E_S^{\text{max}} - e^{\text{min}}, E_S^{\text{max}} - E_{i,j}^{\text{rem}})}{e_M^{\text{charge}}} \quad \forall t_j \in T \quad \forall s_i \in S. \quad (11)$$

Let $\zeta_{i,j}^{\text{req}}$ denote a Boolean variable indicating whether or not the sensor s_i requested a charging request at the time slot t_j . That is,

$$\zeta_{i,j}^{\text{req}} = \begin{cases} 1, & E_{i,j}^{\text{rem}} < e_{\text{th},i}^{\text{req}} \text{ (request charging)} \\ 0, & E_{i,j}^{\text{rem}} > e_{\text{th},i}^{\text{req}} \text{ (sufficient energy).} \end{cases} \quad (12)$$

Let $\zeta_{i,j}^{\text{wrk}}$ denote a Boolean variable indicating whether or not the sensor s_i is working at the time slot t_j . We have

$$\zeta_{i,j}^{\text{wrk}} = \begin{cases} 1, & E_{i,j}^{\text{rem}} > e_i^{\text{min}} \text{ (working)} \\ 0, & E_{i,j}^{\text{rem}} < e_i^{\text{min}} \text{ (sleeping).} \end{cases} \quad (13)$$

The final recharging request constraint guarantees that every sensor s_i must send the recharging request to MC before it switches to the sleeping state. This constraint is shown in (14).

3) Recharging Request Constraint:

$$\zeta_{i,j}^{\text{req}} \geq \zeta_{i,j}^{\text{wrk}} \quad \forall s_i \in S \quad \forall t_j \in T. \quad (14)$$

Section IV will detail the proposed algorithm, aiming to achieve the objective function (9) while satisfying the given constraints (10), (11), and (14). Nomenclature summarizes the common notations used in this article.

IV. ALGORITHM DESCRIPTION

This article proposed an algorithm called RS-STQ. The proposed RS-STQ algorithm develops an efficient-energy recharging schedule for MC, aiming to maximize the data accuracy. To increase the temporal quality, an energy management strategy is proposed for each sensor aiming to adjust its sensing time sequences of neighboring sensors. Therefore, sensors can rescue themselves by saving their energy and wait for the MC to recharge them.

The proposed RS-STQ algorithm mainly consists of three phases, including the *Threshold Determination (TD) Phase*, the *Recharging Path Construction (RPC) Phase*, and the *Sensing Rate Calculation (SRC) Phase*. The first phase aims to calculate the threshold value of each sensor for sending the recharge request to the MC. In the second phase, the MC considers some additional requests in ζ and dynamically adjusts its recharging path. In the final phase, the total waiting time of the scheduled sensors on the current path is calculated. Based on the waiting time, each scheduled sensor will calculate its sensing frequency. If the sensing range of any scheduled sensor overlaps with neighboring scheduled sensors, they can adjust their sensing time sequence to manage their energy until the MC recharges them.

A. Threshold Determination Phase

As mentioned in Section III-B, there are two threshold values for each sensor, including the *recharging request threshold* $e_{\text{th},i}^{\text{req}}$ and *sleep threshold* e^{min} , respectively. Recall that e^{min} is predefined for each sensor s_i , while the other threshold value $e_{\text{th},i}^{\text{req}}$ is locally determined by the sensor s_i . The detailed computation of $e_{\text{th},i}^{\text{req}}$ is discussed in the following.

Most of the related studies assumed that the threshold values of each sensor are fixed. Those studies defined that, whenever the remaining energy of the sensor reached the predefined threshold value, a recharge request will be sent from the sensor to the MC. This strategy is not very flexible. One reason is that the threshold value represents the remaining lifetime of that sensor. A big threshold value causes each sensor to send a recharge request to the MC early, which might mislead the MC schedule. This can cause the sensor to still have higher remaining energy when the MC arrives. Since the energy of the MC was major consumed by moving instead of recharging, the recharging efficiency is low. However, a small threshold value can cause the sensor to enter a sleep state due to energy exhaustion. Therefore, the threshold value should be dynamically determined based on several parameters, including the estimated required time for the MC and the energy discharging rate of each sensor.

Let sensors in set S can be further partitioned into two subsets \tilde{S} and \hat{S} , called the strong-energy set and the weak-energy set of sensors, respectively. Let notation $\tilde{S} = \{\tilde{s}_1, \dots, \tilde{s}_{|\tilde{S}|}\}$ denote the set of strong-energy sensors that are working but still do not send recharge requests to the MC. On the contrary, let $\hat{S} = \{\hat{s}_1, \dots, \hat{s}_{|\hat{S}|}\}$ denote the set of weak-energy sensors that have sent the recharge request to the MC. The strong-energy sensor \tilde{s}_i should perform the operations designed in this phase, which aims to calculate its threshold value.

Unlike the related studies, the proposed RS-STQ algorithm considered the discharging rate e_i^{disch} of sensor $\tilde{s}_i \in \tilde{S}$, and the length of the service pool ζ is maintained by the MC, aiming to dynamically calculate the *recharging request threshold* $e_{\text{th},i}^{\text{req}}$ of each sensor $\tilde{s}_i \in \tilde{S}$. Therefore, the threshold value $e_{\text{th},i}^{\text{req}}$ is not predefined and is determined by each sensor locally. By satisfying the constraint given in (14), this article ensures every sensor sends the recharge request to the MC at the appropriate time. Hence, the *recharging request* plays important role in the recharging scheduling of the MC.

In this article, the MC uses quorum to broadcast its current schedule periodically to all the sensors. This can help sensors estimate the required time for the MC. Recall that MC maintains all the recharge requests in ζ . Assume that there are y requests stored in ζ represented as $\{\hat{s}_1, \hat{s}_2, \dots, \hat{s}_y\}$. Recall that l_k represents the location of the sensor s_k . The MC will broadcast its current schedule to all the sensors by using a quorum. Assume that the current recharge schedule of MC is represented as $((\hat{s}_1, l_1), (\hat{s}_2, l_2), \dots, (\hat{s}_y, l_y))$. The current schedule broadcasted by MC includes the location of each scheduled sensor. The proposed RS-STQ is a distributed algorithm. Based on the schedule of the MC, all the other sensors determine their threshold value $e_{\text{th},i}^{\text{req}}$ locally.

To calculate the threshold value $e_{\text{th},i}^{\text{req}}$ of each sensor, the required time duration for MC to recharge the sensor \tilde{s}_i should be calculated. The total required time duration includes the time required for the MC to move to each sensor node and recharge those scheduled sensors $\hat{s}_i \in \hat{S}$ in service pool ζ . Let τ_k^{crg} denote the time duration required for MC to fully recharge the sensor \hat{s}_k . The value of τ_k^{crg} can be calculated, as shown in the following equation:

$$\tau_k^{\text{crg}} = \begin{cases} \frac{E_S^{\text{max}} - (E_{k,j}^{\text{rem}} - e_k^{\text{disch}} * T_k^{\text{wait}})}{e_M^{\text{charge}}}, & \text{if } k = i \\ \frac{E_S^{\text{max}}}{e_M^{\text{charge}}}, & \text{if } k \neq i. \end{cases} \quad (15)$$

Let \hat{s}_0 denote the sink node. Let v denote the speed of the MC. Let τ_k^{mv} denote the time duration required for MC to move from sink/ \hat{s}_{k-1} to \hat{s}_k . The value of τ_k^{mv} can be calculated, as shown in the following equation:

$$\tau_k^{\text{mv}} = \frac{d(\hat{s}_{k-1}, \hat{s}_k)}{v}, \quad 1 \leq k \leq y. \quad (16)$$

By combining the charging and moving times of each sensor $\hat{s}_i \in \hat{S}$ in ζ , the total time required for a strong-energy sensor $\tilde{s}_i \in \tilde{S}$ to wait for MC can be calculated. Let τ_M^{need} denote the total time required for MC to recharge and move to all the sensors in ζ . Assume that sensor $\tilde{s}_i \in \tilde{S}$ is the next requesting sensor whose request is stored in the $(y+1)$ th position. The value of τ_M^{need} is evaluated in the following equation:

$$\tau_M^{\text{need}} = \sum_{k=1}^y \tau_k^{\text{crg}} + \sum_{k=1}^y \tau_k^{\text{mv}}. \quad (17)$$

Finally, the threshold value of the sensor $\tilde{s}_i \in \tilde{S}$ is evaluated, as shown in the following equation:

$$e_{\text{th},i}^{\text{req}} = e^{\text{min}} + \tau_M^{\text{need}} * \sigma * e_i^{\text{disch}} \quad (18)$$

where σ is a constant and e_i^{disch} is the discharging rate of the sensor \tilde{s}_i . Therefore, each sensor $\tilde{s}_i \in \tilde{S}$ can determine its threshold value $e_{\text{th},i}^{\text{req}}$. The request packet format can be represented as $\langle id, l_i, E_{i,j}^{\text{rem}}, e_i^{\text{disch}}, t_j \rangle$, where id denotes the unique identity, l_i is the sensor location, $E_{i,j}^{\text{rem}}$ is the remaining energy at the time slot t_j , e_i^{disch} is the discharging rate, and t_j is the current time slot, respectively, of the sensor $\tilde{s}_i \in \tilde{S}$. The RPC of the MC is detailed in the following phase.

B. Recharging Path Construction Phase

To consider the opportunities of promptly responding to the local charging requests, the MC should first check its service pool ζ after finishing the recharging task of each sensor.

In this phase, the MC considers additional requests from ζ and calculates the spatial quality benefit of each requested sensor $\hat{s}_i \in \hat{S}$ in ζ . Then, the sensor with the largest spatial quality benefit in ζ will be included in the current path \wp^{current} , while the sensor with the least spatial quality benefit will be deleted from \wp^{current} and be sent to ζ . Recall that $\xi = \hat{s}_1, \hat{s}_2, \dots, \hat{s}_y$ denotes the set of sensors, which have sent recharging requests to the MC. Recall that sensor $\tilde{s}_i \in \tilde{S}$ denotes the strong-energy sensor, which is working but still does not send a recharge request to the MC. Initially, each sensor $\tilde{s}_i \in \tilde{S}$ will calculate the threshold value according to (18) and send the request to MC. After finishing the recharging task of one sensor, the MC will examine its request queue. For each requested sensor \tilde{s}_i in the queue, the MC will calculate the spatial quality contributed by \tilde{s}_i and decides whether or not to include it to \wp^{current} .

The spatial quality calculation is complicated and will be presented in the following. Let c_j denote the Voronoi cell of the sensor s_j . Let $v_j^{\text{far}} \in c_j$ denote the farthest point to s_j . Let $d_{i,j}$ denote the distance between v_i and s_j . v_j^{far} can be derived, as shown in the following equation:

$$v_j^{\text{far}} = \arg \max_{v_i \in c_j, s_j} d_{i,j}. \quad (19)$$

Recall the definition of the spatial quality of a point p at time t , as shown in (7)

$$\lambda_{p,t} = (1 - \rho^{\text{space}})^{d(p,s_i)}. \quad (20)$$

Herein, only the spatial quality contributed by the sensor \tilde{s}_i is concerned. Therefore, the time dimension in (7) will be ignored. That is, the spatial quality of the vertex v_j^{far} will be calculated by

$$\lambda_{v_j^{\text{far}}} = (1 - \rho^{\text{space}})^{d(v_j^{\text{far}}, s_i)}. \quad (21)$$

Let λ_{c_j} denote the spatial quality of cell c_j . The spatial quality of cell c_j will be represented by the smallest spatial quality of points in c_j . That is,

$$\lambda_{c_j} = \arg \min_{p \in c_j} \lambda_p = \lambda_{v_j^{\text{far}}}. \quad (22)$$

According to (22), the spatial quality of each Voronoi cell c_j can be calculated. Next, the spatial quality contributed by the sensor \tilde{s}_i will be calculated by the MC. Let $c_{j,i}$ denote the Voronoi cell, which covers the sensor \tilde{s}_i under the assumption that the sensor \tilde{s}_i stays in a sleep state. Let c_j and c_i denote the

two Voronoi cells in case the sensor \tilde{s}_i stays working. When \tilde{s}_i is sleeping, the Voronoi cell is denoted by $c_{j,\hat{i}}$, and when \tilde{s}_i is working, the Voronoi cells are denoted by c_j and c_i . The difference between the two conditions depends on whether or not \tilde{s}_i is working; the spatial quality contributed by the sensor \tilde{s}_i will be calculated. Consider the first condition that \tilde{s}_i is sleeping. The spatial quality of cell $c_{j,\hat{i}}$ is $\lambda_{c_{j,\hat{i}}}$. On the contrary, consider the second condition that \tilde{s}_i is working. Let λ_{c_j,c_i} denote the spatial quality of cells c_j and c_i . The spatial quality of cells c_j and c_i can be represented by the lower spatial quality of the two cells. That is,

$$\lambda_{c_j,c_i} = \min(\lambda_{c_j}, \lambda_{c_i}). \quad (23)$$

Let \mathcal{A}_i^t denote the spatial quality contributed by the sensor \tilde{s}_i at the current time. The value of \mathcal{A}_i^t can be calculated by the following equation:

$$\mathcal{A}_i^t = \lambda_{c_j,c_i} - \lambda_{c_{j,\hat{i}}}. \quad (24)$$

Let $B_{\tilde{s}_i}$ denote the total data accuracy benefit for including \tilde{s}_i to path \wp^{current} . The total data accuracy benefit is denoted as the ratio of spatial quality contribution of \tilde{s}_i and the increased path length. Recall that the request packet format consists of the location of the sensor. Upon receiving the recharging request from the sensor \tilde{s}_i , the MC knows the location of the sensor. Assume that the MC includes the sensor \tilde{s}_i between the sensors \hat{s}_{j-1} and \hat{s}_j , which have been already scheduled in the recharging path \wp^{current} . Let Δd_i denote the distance difference of MC by including \tilde{s}_i to \wp^{current} . The value of Δd_i can be calculated, as shown in the following equation:

$$\Delta d_i = d_{\hat{s}_{j-1},\tilde{s}_i} + d_{\tilde{s}_i,\hat{s}_j} - d_{\hat{s}_{j-1},\hat{s}_j}. \quad (25)$$

Therefore, the value of $B_{\tilde{s}_i}$ can be evaluated, as shown in the following equation:

$$B_{\tilde{s}_i} = \frac{\mathcal{A}_i^t}{\Delta d_i}. \quad (26)$$

After the MC calculates the data accuracy benefit $B_{\tilde{s}_i}$ of \tilde{s}_i , it further applies two strategies, including *single passer-by update* (SPU) and *multiple passer-by update* (MPU) strategies, to make the recharging decision. In the insertion condition of the SPU strategy, there will be at most one sensor from the service pool ζ to be served and will be included to \wp^{current} . Similarly, in the deletion condition of the SPU strategy, there will be at most one sensor to be deleted from \wp^{current} and to be moved to buffer ζ . On the contrary, the MPU strategy might include more than one sensor from ζ to \wp^{current} , while it also deletes more than one sensor from \wp^{current} . The details of the SPU strategy are presented in the following.

Let $B_{\tilde{s}_i}^\zeta$ and $B_{\hat{s}_j}^\wp$ denote the data accuracy benefit of each sensor $\tilde{s}_i \in \zeta$ and $\hat{s}_j \in \wp^{\text{current}}$, respectively. To obtain $B_{\tilde{s}_i}^\zeta$ and $B_{\hat{s}_j}^\wp$, the data accuracy benefit of each sensor in the path \wp^{current} or in buffer ζ should be calculated by the MC. As shown in (26), the data accuracy benefit of each sensor $\tilde{s}_i \in \zeta$ and $\hat{s}_j \in \wp^{\text{current}}$ should be calculated. Let S_{best}^ζ denote the sensor with the largest data accuracy benefit $B_{\tilde{s}_i}^\zeta$ in ζ . That is,

$$S_{\text{best}}^\zeta = \arg \max_{\tilde{s}_i \in \zeta} B_{\tilde{s}_i}^\zeta.$$

Let S_{worst}^\wp denote the sensor with the least data accuracy benefit $B_{\hat{s}_j}^\wp$ in \wp^{current} . The value of S_{worst}^\wp is calculated as shown in the following:

$$S_{\text{worst}}^\wp = \arg \min_{\hat{s}_j \in \wp^{\text{current}}} B_{\hat{s}_j}^\wp.$$

Let B_{avg}^ζ and B_{avg}^\wp denote the average data accuracy benefit of the sensors in ζ and \wp^{current} , respectively. B_{avg}^ζ can be calculated, as shown in the following equation:

$$B_{\text{avg}}^\zeta = \left(\sum_{\tilde{s}_i=1}^{|\zeta|} B_{\tilde{s}_i}^\zeta \right) / |\zeta|. \quad (27)$$

The notation B_{avg}^\wp can be evaluated, as shown in the following equation:

$$B_{\text{avg}}^\wp = \left(\sum_{\hat{s}_j=1}^{|\wp|} B_{\hat{s}_j}^\wp \right) / |\wp|. \quad (28)$$

The values of B_{avg}^ζ and B_{avg}^\wp are used to determine whether or not sensor \tilde{s}_i will be included to \wp^{current} , and sensors \hat{s}_j will be deleted from \wp^{current} . Let B_{best}^ζ and B_{worst}^\wp denote the data accuracy benefit of S_{best}^ζ and S_{worst}^\wp , respectively. The insertion and deletion conditions of each sensor $\tilde{s}_i \in \zeta$ and $\hat{s}_j \in \wp^{\text{current}}$ of the SPU strategy are presented in the following.

Insert Condition of SPU Strategy:

$$B_{\text{best}}^\zeta > B_{\text{avg}}^\wp.$$

Delete Condition of SPU Strategy:

$$B_{\text{worst}}^\wp < B_{\text{avg}}^\zeta.$$

By applying the insert condition of the SPU strategy, the sensor $\tilde{s}_i \in \zeta$ will be included to \wp^{current} only if it has the largest data accuracy benefit, and it should be greater than B_{avg}^\wp . Similarly, the delete condition of the SPU strategy is that the sensor $\hat{s}_j \in \wp^{\text{current}}$ will be deleted from \wp^{current} if it has the least data accuracy benefit, and it should be smaller than B_{avg}^ζ . The insertion and deletion conditions of the MPU strategy are shown in (29) and (30), respectively.

Insert Condition of MPU Strategy:

$$B_{\tilde{s}_i}^\zeta > B_{\text{avg}}^\wp \quad \forall \tilde{s}_i \in \zeta. \quad (29)$$

Delete Condition of MPU Strategy:

$$B_{\hat{s}_j}^\wp < B_{\text{avg}}^\zeta \quad \forall \hat{s}_j \in \wp^{\text{current}}. \quad (30)$$

In the insert condition of MPU, all the sensors in ζ whose data accuracy benefit is larger than the average benefit B_{avg}^\wp will be included to \wp^{current} . Similarly, in the delete condition of MPU, all the sensors on \wp^{current} whose data accuracy benefit is smaller than B_{avg}^ζ will be deleted from \wp^{current} . Fig. 4 depicts an example to illustrate the insertion and deletion conditions of the SPU strategy.

As shown in Fig. 4, $\wp^{\text{current}} = \hat{s}_{\text{finish}}, \hat{s}_1, \hat{s}_2, \hat{s}_3, \hat{s}_4, \hat{s}_5, \hat{s}_6$. Assume that the MC finishes recharging the sensor \hat{s}_{finish} and wants to recharge the sensor \hat{s}_1 at current time t . During the execution of recharging \hat{s}_{finish} , the MC receives the recharging request from the sensors $\tilde{s}_8, \tilde{s}_{10}$, and \tilde{s}_9 . To include the best

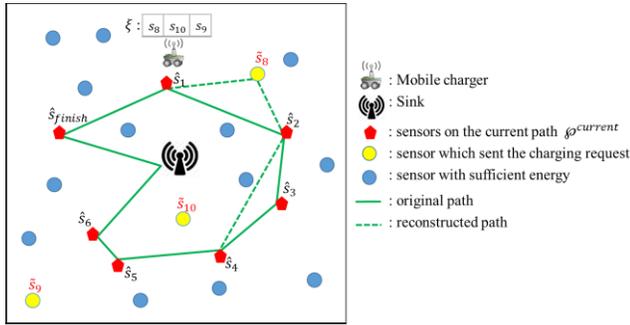


Fig. 4. Example path constructed by MC adopting the SPU strategy.

sensor S_{best}^z to $\varphi^{current}$, the MC executes (19)–(26) for the sensors \tilde{s}_8 , \tilde{s}_{10} , and \tilde{s}_9 . Assume that $S_{best}^z = \tilde{s}_8$ and $S_{worst}^z = \tilde{s}_3$. By applying the SPU strategy, the sensor \tilde{s}_8 will be included only if it satisfies the condition $B_8^z > B_{avg}^z$. Similarly, the sensor \tilde{s}_3 will be deleted from $\varphi^{current}$ if it satisfies $B_3^z < B_{avg}^z$. Let $\varphi^{updated}$ denote the new recharging path constructed by MC. After the execution of the SPU strategy, $\varphi^{updated}$ can be represented as $\varphi^{updated} = \{\hat{s}_{finish}, \hat{s}_1, \hat{s}_8, \hat{s}_2, \hat{s}_4, \hat{s}_5, \hat{s}_6\}$.

C. Sensing Rate Calculation Phase

An energy management strategy is designed in this phase, aiming to adjust the sensing time sequences of each scheduled sensor to increase the temporal data accuracy. There are two tasks in this phase, which are detailed in the following.

1) *Task I: Waiting Time Calculation*: Initially, this task aims to calculate the waiting time of each sensor on the current path. Based on their waiting time, all the sensors can calculate their sensing rate frequency. If the sensing range of any scheduled sensor overlaps with the neighboring scheduled sensors, they can adjust their sensing time sequences according to the following designed operation. Let $\varphi^{current} = \{\hat{s}_{finish}, \hat{s}_1, \hat{s}_2, \dots, \hat{s}_n\}$ denote the current scheduled path of the MC. Let \hat{s}_{finish} denote the most recent recharged sensor on the path $\varphi^{current}$. During the network initialization, the MC considers all the sensors' requests from ζ as the set of recharging candidates and constructs the shortest Hamiltonian path.

The MC calculates the waiting time of each sensor $\hat{s}_i \in \varphi^{current}$. Recall that the value of τ_k^{crg} can be calculated according to (15). Let T_i^{wait} denote the waiting time of each sensor \hat{s}_i . T_i^{wait} can be calculated, as shown in the following equation:

$$T_i^{wait} = \begin{cases} \tau_i^{mv}, & i = 1 \\ T_{i-1}^{wait} + \tau_k^{crg} + \tau_i^{mv}, & 2 \leq i \leq n. \end{cases} \quad (31)$$

Based on their waiting time, each sensor \hat{s}_i can calculate their sensing rate frequency. Let S_i^{freq} denote the sensing rate frequency of the sensor \hat{s}_i . Recall that E_i^{con} denotes the energy consumed by the sensor \hat{s}_i for executing both the sensing and communication tasks. The value of S_i^{freq} is evaluated in the following equation:

$$S_i^{freq} = \frac{E_i^{rem}}{E_i^{con} * T_i^{wait}}. \quad (32)$$

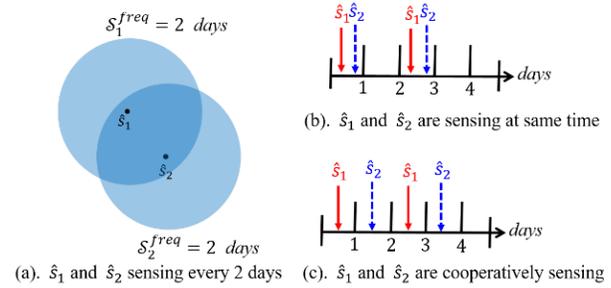


Fig. 5. Example to present the cooperative sensing between neighbors. (a) \hat{s}_1 and \hat{s}_2 sensing every two days. (b) \hat{s}_1 and \hat{s}_2 are sensing at the same time. (c) \hat{s}_1 and \hat{s}_2 are cooperatively sensing.

After calculating the sensing rate frequency, the cooperation between the neighboring sensors is discussed in the following task.

2) *Task II: Cooperative Sensing Between Neighbors*: This task aims to explore the opportunities for cooperative sensing between neighbors to further improve the temporal quality of a local region. The following gives an example to explain the motivation of cooperative sensing between two neighbors. According to the example shown in Fig. 5, it is obvious that the temporal quality can be significantly improved. The following formally presents how each sensor can achieve cooperative sensing by adjusting its sensing time sequences. After calculating S_i^{freq} , each sensor $\hat{s}_i \in \varphi^{current}$ whose sensing range overlaps with their neighboring scheduled sensors will execute the following operations, which adjust its sensing time sequence according to the neighbor's sensing time sequence.

Assume that sensors \hat{s}_u and \hat{s}_v are neighbors. Let α_u and α_v denote the sensing ranges of the sensors \hat{s}_u and \hat{s}_v , respectively.

Let $\eta_{u,v}$ denote a Boolean variable to indicate whether or not α_u and α_v overlaps with each other. The value of $\eta_{u,v}$ is calculated in the following equation:

$$\eta_{u,v} = \begin{cases} 1, & \text{if } \alpha_u \cap \alpha_v \neq \emptyset \\ 0, & \text{otherwise.} \end{cases} \quad (33)$$

Let $T_u = [t_u^1, t_u^2, t_u^3, \dots, t_u^n]$ denote the sensing time sequence of the sensor \hat{s}_u . Any two neighboring sensors should adjust their sensing rate only if it satisfies the following two conditions.

Overlapping Criterion:

The spatial quality of \hat{s}_u and \hat{s}_v should intersect with each other. That is,

$$\eta_{u,v} = 1.$$

Identical-Frequency Criterion:

The two sensors \hat{s}_u and \hat{s}_v should satisfy

$$t_u^i - t_u^{i-1} = t_v^i - t_v^{i-1}.$$

The identical-frequency criteria check if the sensing time sequence of one sensor is identical to the other sensor's time sequences and if they regularly overlap each other.

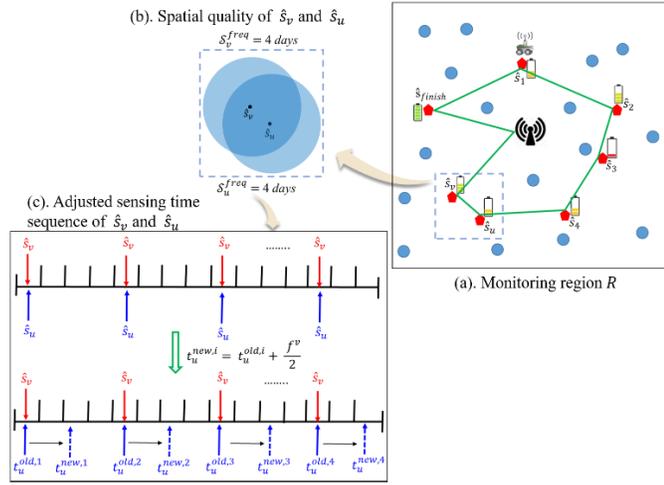


Fig. 6. Example to illustrate the sensing time sequence adjustment of sensors \hat{s}_v and \hat{s}_u . (a) Monitoring region R . (b) Spatial quality of \hat{s}_v and \hat{s}_u . (c) Adjusted sensing time sequence of \hat{s}_v and \hat{s}_u .

Let $N_{\hat{s}_u}$ denote the neighbor nodes of the sensor \hat{s}_u . Each sensor \hat{s}_u should check each of its neighbors $\hat{s}_v \in N_{\hat{s}_u}$ whether or not the above two criteria are satisfied. If it is the case, any sensor \hat{s}_u or \hat{s}_v is scheduled first without any change. Without the loss of generality, the sensor with a larger ID is assumed to schedule first. Assume that \hat{s}_v is scheduled first, and \hat{s}_u adjusts its sensing time sequences to enhance the temporal quality. Let $t_{u,i}^{\text{old}}$ and $t_{u,i}^{\text{new}}$ denote the original and adjusted sensing time sequence of \hat{s}_u . Let f^v denote the sensing time frequency of \hat{s}_v . The sensor \hat{s}_u shifts its sensing time sequence to the right according to (34), aiming to improve the temporal quality of the network

$$t_{u,i}^{\text{new}} = t_{u,i}^{\text{old}} + \frac{f^v}{2}. \quad (34)$$

Fig. 6 illustrates an example to adjust the sensing time sequence of the neighboring sensors. As shown in Fig. 6(a), the red pentagon shape denotes the sensors scheduled on the current path $\wp^{\text{current}} = \{\hat{s}_{\text{finish}}, \hat{s}_1, \hat{s}_2, \hat{s}_3, \hat{s}_4, \dots, \hat{s}_u, \hat{s}_v\}$. The spatial quality of sensors \hat{s}_v and \hat{s}_u is shown in Fig. 6(b). Since the spatial quality of sensors \hat{s}_v and \hat{s}_u overlaps, they should adjust their sensing time sequence, as shown in Fig. 6(c).

Assume that S_v^{freq} and S_u^{freq} are four days, which is identical. If both sensors \hat{s}_v and \hat{s}_u are scheduled at the same time slots, their temporal quality is shown in the upper part of Fig. 6(c). Assume that sensor \hat{s}_v is scheduled first because of its larger ID, and then, the sensor \hat{s}_u will adjust its sensing time sequence, as shown in the bottom part of Fig. 6(c), according to (34). Thus, the temporal quality is improved.

V. PERFORMANCE EVALUATIONS

In this section, the performances of the proposed RS-STQ with SPU and MPU strategies hereafter denoted as RS-STQ(SPU) and RS-STQ(MPU) algorithms are evaluated against the existing ETLBO and MERSH algorithms. The MERSH algorithm [31] considered the tolerable latency to construct the recharging path of the MC. The ETLBO [32]

TABLE II
SIMULATION PARAMETERS

Parameters	Values
Tool	Matlab R2019b
Monitoring area size	600m x 400m
Number of sensor nodes	300 - 700
Deployment type	Random
Sensing range of sensors	5–20m
Sleep threshold e_i^{min}	0.3- 0.7J
Discharging rate e_i^{disch}	0.05J/s
Energy of sensor nodes	3.6kJ
Speed of MC	1 - 8m/s
Moving consumption rate	0.01J/s
Recharging rate	3-15 J/s

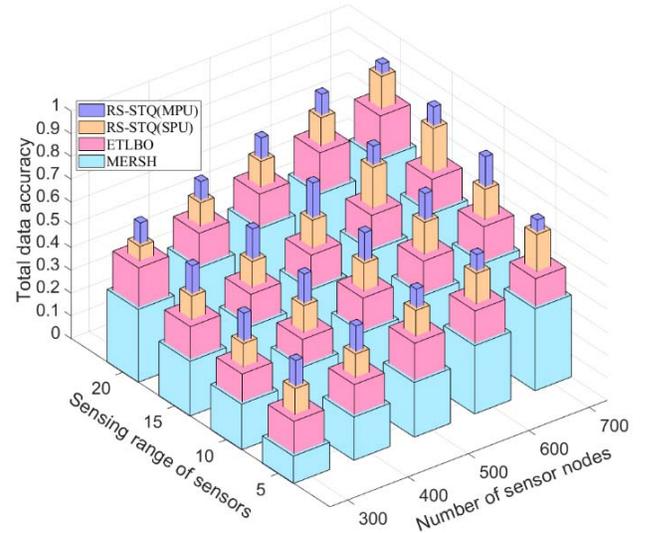


Fig. 7. Performance comparison of total data accuracy for RS-STQ(MPU), RS-STQ(SPU), ETLBO, and MERSH algorithms.

proposed charging scheduling by considering the service cost and energy replenishment utility. This study formulated the best candidate position and time to insert the new charging candidates into the original path aiming to improve the charging efficiency. The simulation environment and results are discussed in the following.

A. Simulation Environment

MATLAB R2019b is used as the simulation tool of this study. The sensor deployment is random in the area of size 600×400 m. The number of sensors is between 300 and 700. The sensing range of each sensor is adjusted between 5 and 20 m. The sleep threshold value e_i^{min} of each sensor is set from 0.3 to 0.7 J. e_i^{disch} is set at 0.05 J/s. The initial energy of each sensor is set at 3.6 kJ. The charging rate of the MC is adjusted between 3 and 15 J/s. The MC speed is varied ranging from 1 to 8 m/s. All the parameters are summarized in Table II.

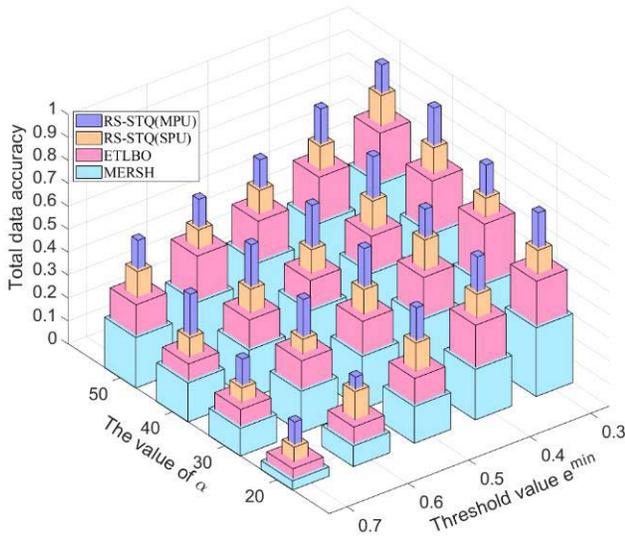


Fig. 8. Performance comparison of total data accuracy by varying the threshold value e^{\min} and value of α .

B. Simulation Results

Fig. 7 compares the total data accuracy of RS-STQ(MPU), RS-STQ(SPU), ETLBO, and MERSH algorithms. The sensors are varied ranging from 300 to 700, while the sensing range is set between 5 and 20 m. As shown in Fig. 7, the total data accuracy of RS-STQ(MPU), RS-STQ(SPU), ETLBO, and MERSH algorithms increases with the sensors. This is due to that deployed sensors can increase the density of sensors in the network. Therefore, the distances between neighboring sensors are generally reduced. This also implies that the MC can recharge more sensors with a smaller movement cost. As a result, the total data accuracy increases with the number of deployed sensors.

On the other hand, the total data accuracy increases with the sensing range. This occurs because a sensor with a larger sensing range can cover a larger area. In comparison, the proposed RS-STQ(MPU) yields the best performance compared to the RS-STQ(SPU), ETLBO, and MERSH algorithms. This occurs because the proposed RS-STQ(MPU) algorithm recharges more sensors that have higher spatial quality compared to RS-STQ(SPU), leading to higher data accuracy. Besides, both the ETLBO and MERSH algorithms improved the charging efficiency and reduced the dead nodes. However, none of the two algorithms considered the spatial and temporal qualities of recharging requested sensors, leading to lower data accuracy.

Fig. 8 depicts the comparison of total data accuracy of RS-STQ(MPU), RS-STQ(SPU), ETLBO, and MERSH algorithms. Let α denote the ratio of recharging and discharging rates. We have

$$\alpha = \frac{e_M^{\text{charge}}}{e_i^{\text{disch}}}. \quad (35)$$

To conduct this experiment, the sleep threshold value e^{\min} is varied ranging from 0.3 to 0.7, while the value of α is adjusted between 20 and 50. As shown in Fig. 8, the total data accuracy decreases with the threshold value e^{\min} . This is due to that the

smaller value e^{\min} leads to longer working time, resulting in higher data accuracy. On the other hand, the total data accuracy is increased with the value of α . This is due to that a large value of α can reduce the recharging time of sensors. Therefore, sensors can be recharged more quickly and then work for improving the data accuracy. In comparison, the proposed RS-STQ(MPU) yields the best performance compared with the RS-STQ(SPU), ETLBO, and MERSH algorithms. This is due to that the RS-STQ(MPU) recharges the sensor, which has higher spatial quality. Therefore, a smaller threshold value e^{\min} and a larger value of α make the sensor work for a longer time, achieving higher data accuracy. The MERSH and ETLBO algorithms considered the tolerable latency and the service cost to recharge the sensors and ignored the spatial and temporal quality. Therefore, the data accuracy of MERSH and ETLBO algorithms is still lower.

The energy usage efficiency (EUE) of the MC and the average waiting time (AWT) of the sensors during the elapsed time are evaluated in Fig. 9(a) and (b), respectively. Let E^{trv} and E^{crg} denote the amount of energy consumed for traveling and charging, respectively. That is,

$$\text{EUE} = \frac{E^{\text{crg}}}{E^{\text{trv}} + E^{\text{crg}}}. \quad (36)$$

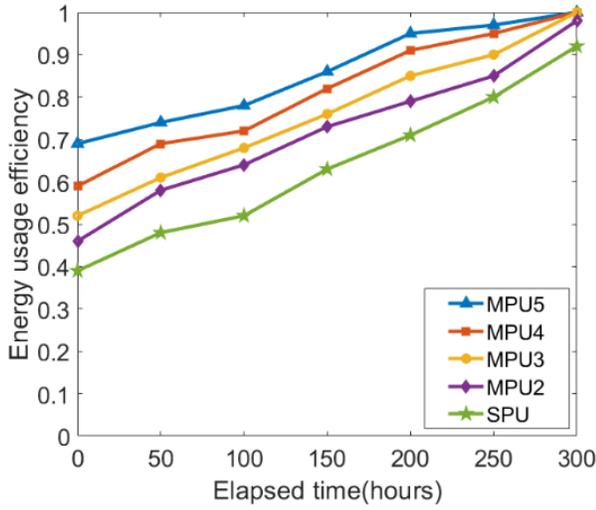
To conduct this experiment, 500 sensor nodes are deployed, and the sensing range of each sensor is set at 10 m. Let SPU, MPU2, MPU3, MPU4, and MPU5 denote the single sensor insertion with the SPU strategy and two, three, four, and five sensors insertion with the MPU strategy to the path. As shown in Fig. 9(a), EUE of SPU, MPU2, MPU3, MPU4, and MPU5 algorithms is generally increased with the elapsed time. This is during the network initialization; all the algorithms construct the shortest Hamiltonian path. Each algorithm inserts the sensors with higher spatial quality as the elapsed time increases; thus, EUE of MC is better utilized. In comparison, the MPU5 yields the best performance. This occurs because MPU5 inserts more sensors that have higher spatial quality compared to all other algorithms. Thus, the MPU5 algorithm achieves higher EUE.

Recall that $T_{i,t}^{\text{wait}}$ denotes the waiting time of the sensor \hat{s}_i , which can be calculated according to (31). Let $T_{i,t}^{\text{avg}}$ denote the AWT of sensors $\hat{s}_i \in \wp^{\text{current}}$ at current time t . The value of AWT can be calculated, as shown in the following equation:

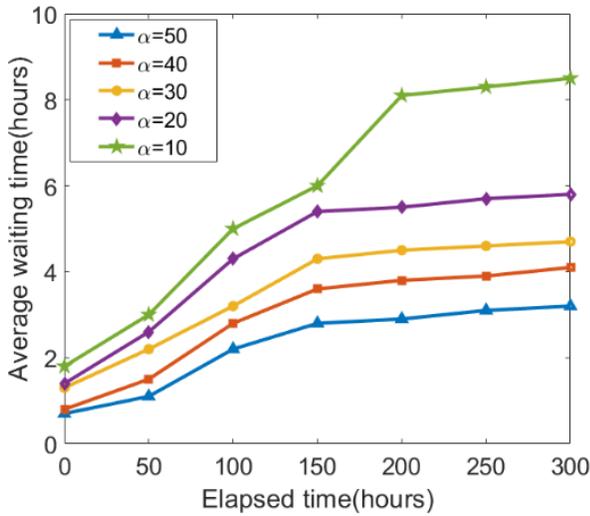
$$T_{i,t}^{\text{avg}} = \frac{\sum_{\hat{i}=1}^{|\hat{n}|} T_{\hat{i},t}^{\text{wait}}}{|\hat{s}_n|}. \quad (37)$$

The simulation settings of Fig. 9(b) are similar to Fig. 9(a). In Fig. 9(b), AWT for the MPU5 algorithm is compared by adjusting the value of α . As shown in Fig. 9(b), when α is set at 50, the value of AWT is very low. This occurs because a large value of α can reduce the recharging time of sensors. That is to say, increasing the charging rate allows the MC to recharge more requested sensors.

Fig. 10 investigates the speed of the MC on the *Recharged Sensors Spatial Quality (RSSQ) Contribution* and total data accuracy. To conduct this experiment, the number of sensors deployed is 700 and 600. The RSSQ is defined by the spatial



(a)



(b)

Fig. 9. Performance comparison of EUE and AWT for SPU, MPU2, MPU3, MPU4, and MPU5 algorithms. (a) EUE versus elapsed time. (b) AWT versus elapsed time.

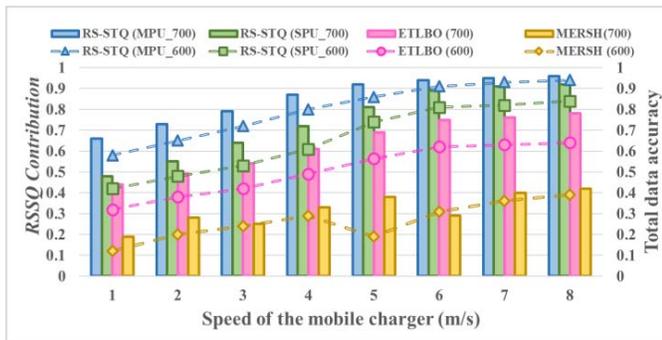
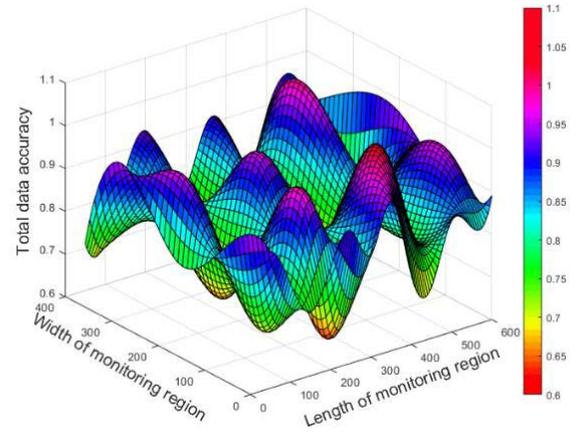
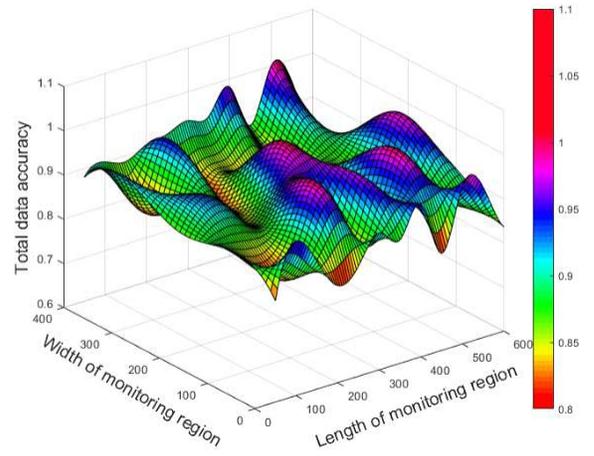


Fig. 10. Performance comparison of RSSQ and total data accuracy by varying the speed of the MC.

quality obtained by the recharged sensors divide by the size of the monitoring region R . Let Ω_i represent the working time length of the sensor s_i . Recall that A_i^x denotes the spatial



(a)



(b)

Fig. 11. Total data accuracy of the network considering the SPU and MPU strategies. (a) With considering the SPU strategy. (b) With considering the MPU strategy.

quality contribution of \hat{s}_i at time slot $t_x \in T$. The RSSQ can be evaluated, as shown in the following equation:

$$RSSQ = \frac{\sum_{i=1}^n \sum_{j=t_1}^{t_m} (\zeta_{i,j}^{crg} \times \sum_{x=t}^{t+\Omega_i} A_i^x)}{R}. \quad (38)$$

As shown in Fig. 10, RSSQ of RS-STQ(MPU) and RS-STQ(SPU) is increased with the speed v . This occurs because, when the speed of the MC increases, it can recharge more number of requested sensors whose spatial quality is higher, leading to higher RSSQ. Besides the RSSQ of ETLBO and MERSH algorithms increases and decreases regardless of the speed v . This is because the spatial quality of recharging sensors is ignored. On the other hand, the data accuracy is decreased with the speed v . This occurs because, when the speed of the MC is decreased, the number of recharged sensors decreases. Thus, the data accuracy is reduced with speed v .

Fig. 11 further analyzes the total data accuracy of the network by considering the SPU and MPU strategies. To conduct

this experiment, 700 sensor nodes are randomly deployed, while the sensing range of each sensor is set at 10 m. The speed of the MC is set at 2 m/s. Fig. 11(a) measures the total data accuracy of the network by considering the SPU strategy. That is to say, the MC constructs the shortest Hamiltonian path to the requested sensors in ζ and recharges them. The MC inserts or removes a single sensor at most, from the current schedule. The total data accuracy of the network is not stable compared with Fig. 11(b). As shown in Fig. 11(a), the highest and lowest data accuracies obtained are 0.68 and 0.991, respectively.

On the other hand, Fig. 11(b) depicts the total data accuracy of the network by applying the MPU strategy. That is to say, the MC inserts or removes more than one sensor at a time from the current schedule. The MPU strategy provides more flexibility to the MC. In case the sensors in ζ contribute larger data accuracy than the sensors on the recharging path, it can quickly remove the low-contribution sensors and insert high-contribution sensors to achieve higher data accuracy. Therefore, the total data accuracy of the network is stable. The highest and lowest data accuracies obtained in Fig. 11(b) are 0.816 and 1, respectively. The total data accuracy obtained in Fig. 11(a) is lower than in Fig. 11(b). In comparison, the total data accuracy is improved by 13.6%.

Fig. 12 further compares the uncharged sensors DAL by varying the value of α and the sleep threshold e^{\min} . The values of α and e^{\min} are varied ranging from 20 to 50 and 0.3 to 0.7, respectively. To conduct this experiment, 700 sensors are randomly deployed, while the sensing range is set at 10 m. The uncharged sensors DAL is defined by the spatial quality loss caused by the requested sensors divided by the area R . The value of uncharged sensors DAL can be calculated, as shown in the following equation:

$$DAL = \frac{\sum_{i=1}^{\hat{n}} \sum_{j=t_1}^{t_m} \left(\mathcal{A}_i^t \times 1 - \zeta_{i,j}^{wrk} \right)}{R}. \quad (39)$$

As shown in Fig. 12, DAL is decreased with the increasing value of α . This occurs because a higher recharging rate and the lower discharging rate reduce the recharging time. Therefore, most of the requested sensors can be recharged, reducing the value of DAL. On the other hand, the DAL is decreased with e^{\min} . This is due to the higher value of e^{\min} leads the requested sensors to switch to the sleep state, resulting in a higher DAL. In comparison, the proposed RS-STQ(MPU) algorithm yields the lowest DAL compared to the RS-STQ(SPU), ETLBO, and MERSH algorithms. This is because the proposed RS-STQ(MPU) considers the spatial quality of the recharging requested sensors and inserts multiple sensors at a time into the path, aiming to achieve higher data accuracy. Besides, the existing two algorithms ignored the spatial quality contributions of recharging requested sensors.

Fig. 13 investigates the performance of the proposed two algorithms, RS-STQ(MPU) and RS-STQ(SPU), against the existing studies ETLBO and MERSH in terms of control overheads. The number of sensor nodes is varied ranging from 300 to 1000. As shown in Fig. 13, the RS-STQ(MPU) and RS-STQ(SPU) algorithms generate more control packets with the number of deployed sensors. This occurs because

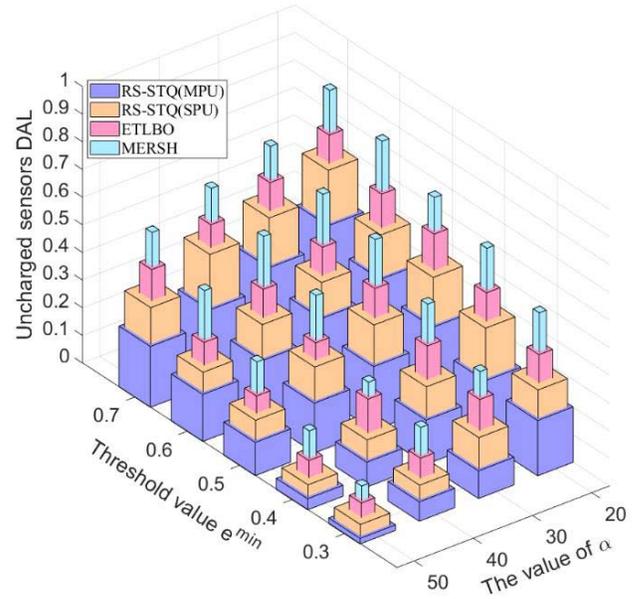


Fig. 12. Performance comparison of uncharged sensors DAL.

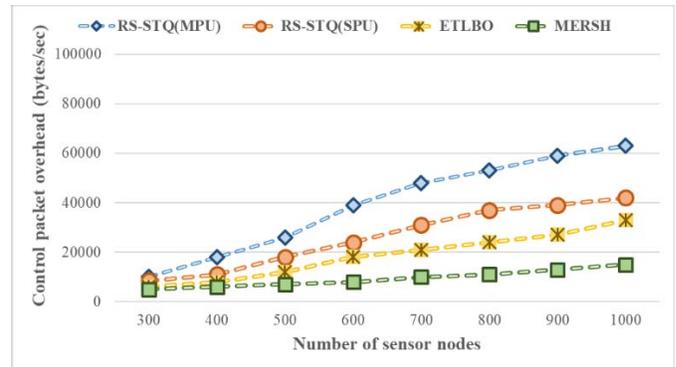


Fig. 13. Performance comparison of control packet overhead by varying the number of sensor nodes.

each deployed sensor should construct its Voronoi diagram and exchange its location information (i.e., control packets) with neighboring sensors. If the number of deployed sensors keeps increasing, there will be more communication between the neighboring sensors, making the connectivity more robust. On the other hand, the control overheads of the ETLBO and MERSH algorithms are slightly increased with the number of deployed sensors. This occurs because of proactive route information maintenance.

VI. CONCLUSION

This article proposed an efficient-energy recharging scheduling algorithm, called RS-STQ for the WRSNs. The proposed algorithm consists of three phases: TD, RPC, and SRC phases. The first phase formulated an adaptive threshold value for the sensor nodes that each of them determines locally. In the second phase, the MC considered the spatial quality of each recharging requested sensor to construct its recharging path. In addition, SPU and MPU strategies are proposed to insert and delete the requested sensors to the current path based

on their spatial quality contribution. In the final phase, the sensing time sequences of two neighboring requested sensors are adjusted, aiming to improve the temporal data accuracy.

Our future work is to focus on dispatching multiple MCs based on the prediction of energy consumption rate and the future request rate of sensors.

REFERENCES

- [1] 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.
- [2] 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.
- [3] C. Jamroen, P. Komkum, C. Fongkerd, and W. Krongpha, "An intelligent irrigation scheduling system using low-cost wireless sensor network toward sustainable and precision agriculture," *IEEE Access*, vol. 8, pp. 172756–172769, 2020.
- [4] 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.
- [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] L. Angrisani, P. Arpaia, A. Esposito, and N. Moccaldi, "A wearable brain–computer interface instrument for augmented reality-based inspection in industry 4.0," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1530–1539, Apr. 2020.
- [7] X. Yue, Y. Liu, J. Wang, H. Song, and H. Cao, "Software defined radio and wireless acoustic networking for amateur drone surveillance," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 90–97, Apr. 2018.
- [8] J. Wang, Y. Liu, S. Niu, and H. Song, "Extensive throughput enhancement for 5G-enabled UAV swarm networking," *IEEE J. Miniaturization Air Space Syst.*, vol. 2, no. 4, pp. 199–208, Dec. 2021.
- [9] M. Hajikhani, F. Labeau, and B. L. Agba, "Power allocation for a self-sustainable power substation monitoring system using wireless transfer of energy," *IEEE Access*, vol. 7, pp. 141456–141465, 2019.
- [10] Q. Qian, A. Y. S. Pandiyan, and D. E. Boyle, "Optimal recharge scheduler for drone-to-sensor wireless power transfer," *IEEE Access*, vol. 9, pp. 59301–59312, 2021.
- [11] H. Dai et al., "ROSE: Robustly safe charging for wireless power transfer," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 2180–2197, Jun. 2022.
- [12] J. Chen, C. W. Yu, and W. Ouyang, "Efficient wireless charging pad deployment in wireless rechargeable sensor networks," *IEEE Access*, vol. 8, pp. 39056–39077, 2020.
- [13] Y. Shu, K. G. Shin, J. Chen, and Y. Sun, "Joint energy replenishment and operation scheduling in wireless rechargeable sensor networks," *IEEE Trans. Ind. Informat.*, vol. 13, no. 1, pp. 125–134, Feb. 2017.
- [14] G. Jiang, S.-K. Lam, Y. Sun, L. Tu, and J. Wu, "Joint charging tour planning and depot positioning for wireless sensor networks using mobile chargers," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2250–2266, Aug. 2017.
- [15] Z. Chen, X. Chen, D. Zhang, and F. Zeng, "Collaborative mobile charging policy for perpetual operation in large-scale wireless rechargeable sensor networks," *Neurocomputing*, vol. 270, pp. 137–144, Dec. 2017.
- [16] W. Xu, W. Liang, X. Jia, Z. Xu, Z. Li, and Y. Liu, "Maximizing sensor lifetime with the minimal service cost of a mobile charger in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 11, pp. 2564–2577, Nov. 2018.
- [17] Y. Ma, W. Liang, and W. Xu, "Charging utility maximization in wireless rechargeable sensor networks by charging multiple sensors simultaneously," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1591–1604, Aug. 2018.
- [18] Z. W. Lu et al., "Periodic charging planning for a mobile WCE in wireless rechargeable sensor networks based on hybrid PSO and GA algorithm," *Appl. Soft Comput.*, vol. 75, no. 1, pp. 388–403, Feb. 2019.
- [19] T. T. Huong, P. L. Nguyen, H. T. T. Binh, K. Nguyenz, N. M. Hai, and L. T. Vinh, "Genetic algorithm-based periodic charging scheme for energy depletion avoidance in WRSNs," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–6.
- [20] B. Dande, S.-Y. Chen, H.-C. Keh, S.-J. Yang, and D. S. Roy, "Coverage-aware recharging scheduling using mobile charger in wireless sensor networks," *IEEE Access*, vol. 9, pp. 87318–87331, 2021.
- [21] Y. Ren, A. Liu, X. Mao, and F. Li, "An intelligent charging scheme maximizing the utility for rechargeable network in smart city," *Pervas. Mobile Comput.*, vol. 77, Oct. 2021, Art. no. 101457.
- [22] C. Lin, J. Zhou, C. Guo, H. Song, G. Wu, and M. S. Obaidat, "TSCA: A temporal-spatial real-time charging scheduling algorithm for on-demand architecture in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 1, pp. 211–224, Jan. 2018.
- [23] C. Lin et al., "GTCCS: A game theoretical collaborative charging scheduling for on-demand charging architecture," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12124–12136, Oct. 2018.
- [24] K. Wang, L. Wang, M. S. Obaidat, C. Lin, and M. Alam, "Extending network lifetime for wireless rechargeable sensor network systems through partial charge," *IEEE Syst. J.*, vol. 15, no. 1, pp. 1307–1317, Mar. 2021.
- [25] C. Lin, Y. Sun, K. Wang, Z. Chen, B. Xu, and G. Wu, "Double warning thresholds for preemptive charging scheduling in wireless rechargeable sensor networks," *Comput. Netw.*, vol. 148, pp. 72–87, Jan. 2019.
- [26] A. Tomar, L. Muduli, and P. K. Jana, "An efficient scheduling scheme for on-demand mobile charging in wireless rechargeable sensor networks," *Pervas. Mobile Comput.*, vol. 59, Oct. 2019, Art. no. 101074.
- [27] A. Tomar, L. Muduli, and P. K. Jana, "A fuzzy logic-based on-demand charging algorithm for wireless rechargeable sensor networks with multiple chargers," *IEEE Trans. Mobile Comput.*, vol. 20, no. 9, pp. 2715–2727, Sep. 2021.
- [28] X. Cao, W. Xu, X. Liu, J. Peng, and T. Liu, "A deep reinforcement learning-based on-demand charging algorithm for wireless rechargeable sensor networks," *Ad Hoc Netw.*, vol. 110, Jan. 2021, Art. no. 102278.
- [29] R. Kumar and J. C. Mukherjee, "On-demand vehicle-assisted charging in wireless rechargeable sensor networks," *Ad Hoc Netw.*, vol. 112, Mar. 2021, Art. no. 102389.
- [30] A. Kaswan, A. Tomar, and P. K. Jana, "An efficient scheduling scheme for mobile charger in on-demand wireless rechargeable sensor networks," *J. Netw. Comput. Appl.*, vol. 114, no. 15, pp. 123–134, Jul. 2018.
- [31] Y. Feng, L. Guo, X. Fu, and N. Liu, "Efficient mobile energy replenishment scheme based on hybrid mode for wireless rechargeable sensor networks," *IEEE Sensors J.*, vol. 19, no. 21, pp. 10131–10143, Nov. 2019.
- [32] C. Zhao, H. Zhang, F. Chen, S. Chen, C. Wu, and T. Wang, "Spatiotemporal charging scheduling in wireless rechargeable sensor networks," *Comput. Commun.*, vol. 152, pp. 155–170, Feb. 2020.
- [33] J. Yang, J.-S. Bai, and Q. Xu, "An online charging scheme for wireless rechargeable sensor networks based on a radical basis function," *Sensors*, vol. 20, no. 1, p. 205, Dec. 2019.
- [34] C. Pang, G.-G. Xu, G.-L. Shan, and Y.-P. Zhang, "A new energy efficient management approach for wireless sensor networks in target tracking," *Defence Technol.*, vol. 17, no. 3, pp. 932–947, Jun. 2021.
- [35] Y. Deng, C. Han, J. Guo, and L. Sun, "Temporal and spatial nearest neighbor values based missing data imputation in wireless sensor networks," *Sensors*, vol. 21, no. 5, p. 1782, Mar. 2021.
- [36] D. Parker, M. Stojanovic, and C. Yu, "Exploiting temporal and spatial correlation in wireless sensor networks," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Nov. 2013, pp. 442–446.



Bhargavi Dande received the M.S. and Ph.D. degrees in computer science and information engineering from Tamkang University, New Taipei, Taiwan, in 2019 and 2022, respectively.

She is currently a Postdoctoral Researcher with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan. Her current research interests are the Internet of Things, wireless sensor networks, and artificial intelligence.



Chih-Yung Chang (Member, IEEE) received the Ph.D. degree in computer science and information engineering from National Central University, Taoyuan, Taiwan, in 1995.

He is currently a Distinguished Professor with the Department of Computer Science and Information Engineering and the Department of Artificial Intelligence, Tamkang University, New Taipei, Taiwan. His current research interests include artificial intelligence, deep learning and machine learning, the Internet of Things, and wireless

sensor networks.

Dr. Chang is the Co-Chair of ACM Sigmobile in Taiwan. He has served/has been serving as an Associate Guest Editor for several science citation index (SCI)-indexed journals, including *International Journal of Ad Hoc and Ubiquitous Computing* (IAHUC) since 2011, *Journal of Applied Science and Engineering* (JASE) since 2018, *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.



Chin-Hwa Kuo received the B.S. degree in mechanical engineering from Chung Yuan Christian University, Taoyuan, Taiwan, in 1980, the M.S. degree in electrical engineering from Marquette University, Milwaukee, WI, USA, in 1989, and the Ph.D. degree in electrical engineering from the University of Notre Dame, South Bend, IN, USA, in 1994.

He is currently a Full Professor with the Department of Computer Science and Information Engineering and the Chief of Information Officer with Tamkang University, New Taipei, Taiwan. His research interests include multimedia processing and applications, learning analytics, and large-scale deployment of information and communication technologies (ICT) in education.



Diptendu Sinha Roy received the Ph.D. degree in engineering from the Birla Institute of Technology, Mesra, India, in 2010.

In 2016, he joined the Department of Computer Science and Engineering, National Institute of Technology (NIT) Meghalaya, Shillong, India, as an Associate Professor, where he has been the Chair of the Department of Computer Science and Engineering since January 2017. Prior to his stint at 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 & cloud computing, and the Internet of Things (IoT), specifically applications of artificial intelligence/machine learning for smart integrated systems.

Dr. Roy is a member of the IEEE Computer Society.