Novel Methods for Energy Charging and Data Collection in Wireless Rechargeable Sensor Networks

Bing-Hong Liu, Ngoc-Tu Nguyen, Van-Trung Pham, and Yue-Xian Lin

***Abstract*—In wireless rechargeable sensor networks, sensors are responsible for sensing environment and generating sensed data; and mobile devices are responsible for recharging sensors and/or collecting sensed data to the sink. Due to the rapid development of wireless charging technology, sensors can be recharged when they are within limited charging ranges of mobile devices. In addition, because sensors' electric capacity and memory storage are often limited, sensors must be recharged and their generated data must be collected by mobile devices periodically, or the network cannot provide adequate quality of services. Therefore, the problem of scheduling minimum mobile devices to periodically recharge and collect data from sensors subject to the limited charging range, electric capacity, and memory storage, such that the network lifetime can be guaranteed to be prolonged without limits, termed the Periodic Energy Replenishment and Data Collection (PERDC) problem, is studied in the paper. For the problem, the grid-based algorithm (GBA), the dominating-set-based algorithm (DSBA), and the circle-intersection-based algorithm (CIBA) are proposed to find a set of anchor points. In addition, the mobile device scheduling algorithm (MDSA) is proposed to schedule minimum mobile devices to visit the generated anchor points. Simulation results show that our proposed methods provide good performance.**

***Index Terms*—Wireless rechargeable sensor network, energy replenishment, data gathering, NP-complete.**

I. Introduction

DUE to the improvement in miniature techniques, the size and cost of sensors are decreasing, respectively. In addition, many functions are available for sensors, such as the sensing of humidity, temperature, light, pressure, and sound, and data transmission. Because sensors can be deployed to monitor a certain area and can communicate with each other, a wireless sensor network can thus be constructed [1], [2], [3], [4]. Today, wireless sensor networks have been widely studied for many environment surveillance applications [5], [6], [7]. For example, they can be used to detect environmental [8], health, and traffic conditions. In wireless sensor networks, sensors often need to report the sensory data back to a certain node, called a sink. Data sensing and reporting consume most of the sensors' energy. However, the electric capacity of sensors is limited. Therefore, preventing wireless sensor networks from collapsing because sensors run out of energy is a very important issue. In the paper, we study energy replenishment and data gathering in wireless sensor networks whose sensors can be recharged. Here, these networks are also known as wireless rechargeable sensor networks (WRSNs).

B.H. Liu, N.T. Nguyen, V.T. Pham and Y.X. Lin are with the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, 415, Chien Kung Rd., Kaohsiung 80778, Taiwan.
E-mail: bhliu@cc.kuas.edu.tw, kuastu2011@gmail.com, pvtrung@pdu.edu.vn, skybill@hotmail.com.tw.
Corresponding author: B.H. Liu (Fax: 886 7 3811182).

Many studies have addressed the energy replenishment problem by using natural energy resources, such as thermal, light (solar), and wind energy [9], [10], [11]. In [9], an effective energy management is proposed such that sensors can obtain the most natural energy. In [10], sensors are scheduled to sense the environment and obtain energy from natural energy resources in duty cycles. When the natural energy replenishment rate varies with time, efficient algorithms are proposed in [11] to track instantaneous optimal sampling rates and routes, and to maintain the battery at the desired target level. Because the natural energy is often unstable, and varies with time and the environment, the WRSNs whose sensors are recharged by natural energy provide only low-rate data services.

In WRSNs, wireless charging techniques are often used to charge sensors to replenish their energy. When wireless charging techniques are applied to mobile devices, the mobile devices can be scheduled to recharge sensors in WRSNs. In [12], a wireless charging system is proposed to prolong the network lifetime. In the system, the sink is responsible for collecting energy information reported by sensors. When some sensors need energy, a mobile robot is scheduled to visit the sensors for energy replenishment. In [13], a battery-aware mobile energy replenishment method is proposed to schedule a mobile device to visit locations such that the sensors within the charging range of the mobile device can be recharged. In [14], a limited number of mobile vehicles are scheduled to recharge sensors with the minimum total traveling cost of multiple vehicles when the sensors' energy status is collected by mobile vehicles. However, in the studies, the sensory data often must be relayed by multiple nodes to achieve the sink, which places a heavy burden on relay nodes.

Recently, to save sensors' energy on reporting data to the sink, many studies have investigated efficiently using mobile devices to recharge and collect data from sensors when the sensors' data are assumed to be collected by selected sensors with multi-hop routing [15], [16], [17]. In [15], two stages, including the recharge stage and the data collection stage, are required for WRSNs. In the recharge stage (or the collection stage), sensors are selected for a mobile device to visit for energy replenishment (or data collection). In [16], mobile devices recharge sensors and collect data at the same time. An algorithm is then proposed to schedule mobile devices to go through the selected nodes following a fixed journey formed by a continuous square wave shape, such that the sensors' lifetime is prolonged. In [17], a method based on a traveling salesman problem (TSP) genetic algorithm is proposed to schedule multiple mobile devices to visit the pre-defined sensors. In the studies, every sensor must transmit data to selected nodes, which places a heavy burden on the relay nodes and the selected nodes. Some sensors may be compelled to go to sleep because they run out of energy, which makes it hard for the network to provide stable quality of services.

Because of the rapid development of wireless charging technology, sensors are recharged when they are within limited charging ranges of mobile devices [18], [19], [20], [21]. In addition, because the electric capacity and memory storage [22], [23] are often limited, sensors must be recharged and their generated data must be collected by mobile devices periodically, or the network cannot provide adequate quality of services. Moreover, because mobile devices with charging capability are often costly, they have to be used as little as possible. Therefore, in the paper, the problem of scheduling minimum mobile devices to periodically recharge and collect data from sensors subject to the limited charging range, electric capacity, and memory storage, such that the network lifetime can be guaranteed to be prolonged without limits, termed the Periodic Energy Replenishment and Data Collection (PERDC) problem, is studied. The remaining sections of this paper are organized as follows: Section II introduces the formal problem definition and its difficulty. In Section III, algorithms are proposed for the PERDC problem. In Section IV, the performance of our proposed algorithms are evaluated. Finally, we conclude this paper in Section V.

II. Problem Definition and Its Difficulty

A. System Model

In a WRSN, sensors are static and deployed in a sensing field. In the network, every sensor u can transmit data to other sensors within its transmission range, denoted by $u.R_t$. In addition, every sensor can be recharged by mobile devices. The mobile device m can work as a transmitter of energy power to recharge a sensor u if u is within m's charging range R_c. In addition, m can collect data from sensor u if m is within u's transmission range $u.R_t$. In this paper, we assume that $R_c \leq u.R_t$ for all sensors u in the networks. That is, when a mobile device m can recharge a sensor u, u can send data to m for data collection. When a mobile device m completes collecting data, m will move to a special node, termed the data sink, to report the collected data. Note that because the data generated by sensors can be collected through mobile devices, the WRSN does not necessarily have to be connected.

In a WRSN, sensors often have their own limitations, such as limited electric capacity and memory storage [22], [23]. Due to the limited electric capacity, sensors must be recharged periodically, or their energy may be depleted such that some field no longer be monitored. In addition, because the memory storage of each sensor is limited, the data generated by sensors must be collected periodically by mobile devices, or

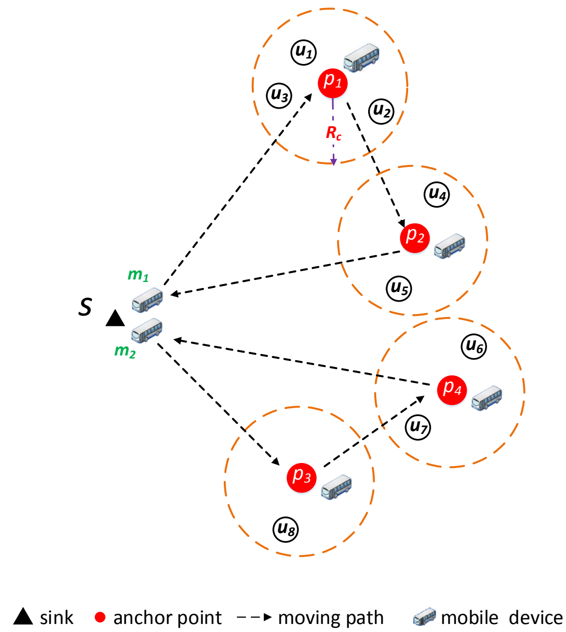

Fig. 1: Example of scheduling mobile devices for energy replenishment and data collection.

their memory storage fills up and some must be abandoned. Therefore, every sensor has to be recharged, and the data stored in every sensor have to be collected periodically within a certain time period. The required time period depends on the sensor's electric capacity and memory storage. Here, the maximum time period for periodically recharging sensor u (or, periodically collecting data from sensor u) by mobile devices is denoted by $u.t_c$ (or, $u.t_d$). Because mobile devices can recharge sensors and collect data from sensors at the same time [17], [16], we assume that it takes at most time δ for any mobile device to recharge the sensors completely and collect all data from the sensors.

Because the mobile device has to stay at a location for time δ to recharge sensors within the charging range R_c and collect all data from sensors whose transmission ranges cover the mobile device, a number of locations, called anchor points hereafter, are selected for mobile devices to be periodically visited to recharge and collect data. Let an anchor list be the list of anchor points (p_1, p_2, ..., p_n). When an anchor list (p_1, p_2, ..., p_n) is determined for a mobile device m, m will follow the movement path $(s, p_1, p_2, \ldots, p_n, s)$ for one round of energy replenishment and data collection; that is, m will start from the data sink s, visit p_1, p_2, ..., p_n sequentially, and move back to s. When m visits p_i ($1 \leq i \leq n$), m will stay at p_i for time δ, recharge sensors that are within m's charging range, and collect data from sensors whose transmission ranges cover p_i. Take Fig. 1, for example. In Fig. 1, an anchor list $L_1 = (p_1, p_2)$ is determined for mobile device m_1; that is, m_1 is used to periodically visit p_1 and p_2. When p_1 (or, p_2) is visited, sensors u_1, u_2, and u_3 (or, u_4 and u_5) can be fully recharged and their generated data can be collected by m_1.

Because every sensor u in a WRSN must be recharged periodically within the time period $u.t_c$ and the generated data must be collected periodically within the time period $u.t_d$, it requires at most time period $\min(u.t_c, u.t_d)$ for mobile devices to periodically recharge and collect data from u. Let p be an anchor point. Also let $\{u_1, u_2, \ldots, u_n\}$ be a set of sensors that are recharged and the data collected by a mobile device m located at p. It requires at most time period $p.T$ for m to periodically visit p such that u_1, u_2, ..., u_n can be recharged and their data can be collected, where $p.T = \min(\min(u_1.t_c,\ u_1.t_d),\ \min(u_2.t_c, u_2.t_d),\ \ldots,\ \min(u_n.t_c, u_n.t_d))$. That is, if an anchor list $L = (p_1,\ p_2,\ \ldots,\ p_q)$ is determined for m, the time to traverse the whole movement path $(p_0,\ p_1,\ p_2,\ \ldots,\ p_q,\ p_{q+1})$ by m cannot be greater than $p.T$, or some of $u_1,\ u_2,\ \ldots,\ u_n$ may deplete its energy or some data may be abandoned, where $p_i = p$ $(1 \leq i \leq q)$ and $p_0 = p_{q+1} = s$. The total distance of the movement path is the distance from p_0, through $p_1, p_2, \ldots, p_q$, sequentially, to p_{q+1}, that is, $\sum_{1\leq i\leq q+1} dist(p_{i-1}, p_i)$, where $dist(p_{i-1}, p_i)$ denotes the distance between anchor points p_{i-1} and p_i. Assume that mobile devices move at a constant speed ν. The time to go through the whole movement path for m without any energy replenishment and data collection is $1/\nu \times \sum_{1\leq i\leq q+1} dist(p_{i-1}, p_i)$. Because q anchor points each are required to be visited and stayed at for time δ, the total time for m to finish one round of energy replenishment and data collection is $q\delta + 1/\nu \times \sum_{1\leq i\leq q+1} dist(p_{i-1}, p_i)$. Because a mobile device requires at most time δ to recharge and collect data from sensors, the maximum waiting time for a sensor u_i $(1 \leq i \leq n)$ to wait for the next round of energy replenishment and data collection is $(q-1)\delta + 1/\nu \times \sum_{1\leq i\leq q+1} dist(p_{i-1}, p_i)$. Because the maximum waiting time for u_i cannot be greater than $p.T$, we have that $(q-1)\delta + 1/\nu \times \sum_{1\leq i\leq q+1} dist(p_{i-1}, p_i) \leq p.T$. Hereafter, an anchor point p is said to satisfy the time constraint for a mobile device m if p is included in the anchor list $L = (p_1,\ p_2,\ \ldots,\ p_q)$ determined for m and the following equation is satisfied:

$$(q-1)\delta + 1/\nu \times \sum_{1\leq i\leq q+1} dist(p_{i-1}, p_i) \leq p.T, \quad (1)$$

where $p_0 = p_{q+1} = s$. The following notations are necessary for the description of our problem.

Definition 1: An anchor list L is said to be a time-constrained anchor list of a mobile device m if every anchor point p in L satisfies the time constraint for m.

Definition 2: A sensor u is said to be operated without limits by a mobile device m if u is recharged and collects the data by m located at anchor point p, where p is included in the time-constrained anchor list of m. In addition, a WRSN is said to be operated without limits if every sensor in the network can be operated without limits by scheduled mobile devices.

Take Fig. 1, for example. Let u_1, u_2, and u_3 be recharged and the data collected when a mobile device visits anchor point p_1. Also let $\min(u_1.t_c,\ u_1.t_d) = \min(u_2.t_c,\ u_2.t_d) = \min(u_3.t_c,\ u_3.t_d) = 7$. We have $p_1.T = 7$. Assume that $dist(s, p_1) = dist(p_1, p_2) = dist(p_2, s) = 2$. Let an anchor list $L_1 = (p_1, p_2)$ determined for mobile device m_1. When the speed ν and the charging time δ are set to 1 for each mobile device, we have that $(2-1) \times 1 + 1/1 \times (2+2+2) = 7 \leq p_1.T$. We thus can say that p_1 satisfies the time constraint for m_1. Assume that $p_2.T > p_1.T$. We have that the anchor list L_1 is a time-constrained anchor list of m_1. We thus have that u_1, u_2, and u_3 can be operated without limits by m_1. Assume that $L_2 = (p_3, p_4)$ is a time-constrained anchor list of mobile device m_2. We can say that the WRSN as shown in Fig. 1 can be operated without limits by m_1 and m_2.

B. Periodic Energy Replenishment and Data Collection Problem and Its Difficulty

Given a WRSN with limited charging range, electric capacity, and memory storage, our problem is to schedule minimum mobile devices for energy replenishment and data collection such that the network can be operated without limits, termed the Periodic Energy Replenishment and Data Collection (PERDC) problem hereafter. The PERDC problem is formally illustrated as follows:

INSTANCE: Given a sink s, a set of deployed sensors $U = \{u_1, u_2, \ldots, u_n\}$, the speed of mobile devices ν, the charging range of mobile devices R_c, the charging time δ, and $k_M \in \mathbb{Z}^+$, where each sensor $u \in U$ has its own position, $u.t_c$, and $u.t_d$.

QUESTION: Does there exist a schedule of mobile devices in a WRSN by dispatching no more than k_M mobile devices for energy replenishment and data collection such that the WRSN can be operated without limits?

Take Fig. 1, for example. In Fig. 1, two mobile devices m_1 and m_2 can be scheduled for energy replenishment and data collection such that the WRSN can be operated without limits.

In this paper, the TSP [24] is used to show the difficulty of the PERDC problem, as shown in Theorem 1.

Theorem 1: The PERDC problem is NP-complete.

Proof: In the TSP [24], while given a number of cities, the problem is to find the shortest route that visits each city exactly once and returns to the original city. That is, while given a set of cities $V = \{v_1, v_2, \ldots, v_m\}$, where each city $v \in V$ has its own position. The TSP is to find a shortest route that starts from one city, through other cities, and back to the original city. It is clear that in the PERDC problem, when the speed of mobile devices ν is set to 1, the charging range of mobile devices R_c is set to 0, the charging time δ is set to 0, k_M is set to 1, and $u.t_c$ $(u.t_d)$ is set to ∞ for each $u \in U$, the TS problem is also a PERDC problem. Therefore, we have that the TSP is a subproblem of the PERDC problem. In addition, because the TSP is NP-hard [24] and the PERDC problem clearly belongs to the NP class, the proof is thus completed. ■

III. The Proposed Methods

Because every sensor in a WRSN must be recharged and data collected within a time period periodically, anchor lists must be determined for mobile devices such that the WRSN can be operated without limits. In the following, when a set of all sensors U deployed in a sensor field $FIELD$ is given, three heuristics, called the grid-based algorithm (GBA), the dominating-set-based algorithm (DSBA), and the circle-intersection-based algorithm (CIBA), are proposed to find a set of anchor points A, and are illustrated, respectively, in Sections III-A, III-B, and III-C. In addition, for each proposed algorithm, we calculate the number of sensors that can be recharged and data collected when a mobile device visits a point $p \in A$, denoted by $p.\omega$. In addition, we also calculate $p.T$ for each $p \in A$. The $p.\omega$ and $p.T$ for each $p \in A$ are used for the scheduling of mobile devices. In Section III-D, an algorithm, termed mobile device scheduling algorithm (MDSA), is proposed to schedule minimum mobile devices to visit the anchor points generated by the GBA, the DSBA, or the CIBA.

A. The Grid-Based Algorithm

In the subsection, we propose an algorithm, termed the grid-based algorithm (GBA), to find a set of anchor points for mobile devices. In the GBA, our idea is to divide the sensor field into grids of squares with length λ, where $\lambda = \sqrt{2}R_c$. It is clear that when the center of a grid is visited by a mobile device, the sensors located in the grid can be recharged and the data collected. Therefore, the centers of grids can be treated as anchor point candidates. Because some grids may have no sensors located, we therefore select the centers of the grids in which the sensors are located as anchor points. Algorithm 1 shows the GBA in details.

Algorithm 1 Grid-Based Algorithm $(FIELD, U)$

1: Let P be a set of center points $p_{x,y}$ in the grids labeled with coordinates (x, y), where the grids are obtained by dividing the sensor field $FIELD$ into grids of squares with length $\sqrt{2}R_c$
2: **for** each $p_{x,y} \in P$ **do**
3: $\quad p_{x,y}.T \leftarrow \infty$
4: $\quad p_{x,y}.\omega \leftarrow 0$
5: **end for**
6: **for** each $u \in U$ **do**
7: $\quad$ Let $p_{x,y}$ be the center point of the grid in which u is located
8: $\quad p_{x,y}.\omega \leftarrow p_{x,y}.\omega + 1$
9: $\quad$ **if** $p_{x,y}.T > \min(u.t_c, u.t_d)$ **then**
10: $\quad\quad p_{x,y}.T \leftarrow \min(u.t_c, u.t_d)$
11: $\quad$ **end if**
12: **end for**
13: Let A be the set of points $p_{x,y} \in P$ with $p_{x,y}.\omega \neq 0$
14: **return** A;

In Algorithm 1, the division of a sensor field and the initialization of $p.\omega$ and $p.T$ for each anchor point candidate p are illustrated in Lines $1-5$. In Lines $6-12$, we calculate each $p.\omega$ to be the total number of sensors located in the grid with a center point p. In addition, we also calculate each $p.T$ to be the minimum value of $\min(u.t_c, u.t_d)$ for all sensors u located in the grid with a center point p. Then A can be obtained by selecting the candidates of anchor points p with $p.\omega \neq 0$.

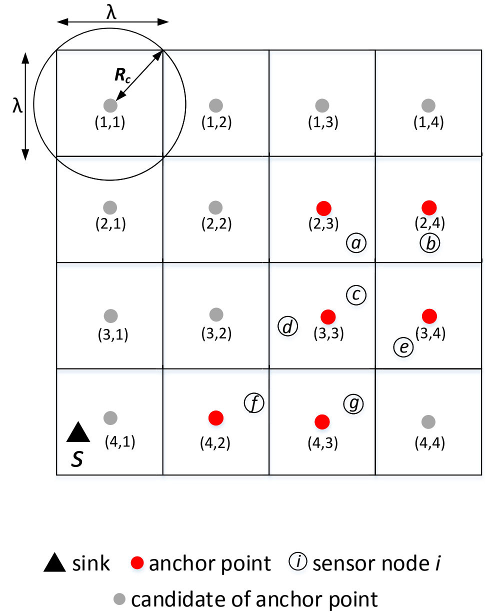

Fig. 2: Example of the grid-based algorithm, where the sensor field is divided into grids of squares, and the numbers in parentheses indicate the coordinates of the centers in the corresponding grids.

Take Fig. 2, for example. Fig. 2 shows a sensor field that is divided into 16 grids of squares that has length λ. Because the grid with a coordinate $(3, 3)$ has two sensors located, we have that $p_{3,3}.\omega = 2$, where $p_{3,3}$ denotes the center point in the grid with a coordinate $(3, 3)$. Because only the grids with coordinates $(2, 3)$, $(2, 4)$, $(3, 3)$, $(3, 4)$, $(4, 2)$, and $(4, 3)$ have sensors located, we have that $A = \{p_{2,3}, p_{2,4}, p_{3,3}, p_{3,4}, p_{4,2}, p_{4,3}\}$.

Theorem 2 shows an analysis of the time complexity of the GBA.

Theorem 2: The time complexity of the GBA is bounded in $O\left(\left(\frac{\ell}{\sqrt{2}R_c}\right)^2 + n\right)$, where ℓ and n denote the side length of a sensor field and the number of deployed sensors, respectively.

Proof: Because ℓ is the side length of a sensor field, the division of the sensor field requires at most $O\left(\frac{\ell}{\sqrt{2}R_c}\right)^2$. In addition, because n sensors are considered for calculating $p.\omega$ and $p.T$ for some anchor points' candidates p, the time complexity of the GBA is therefore bounded in $O\left(\left(\frac{\ell}{\sqrt{2}R_c}\right)^2 + n\right)$. This completes the proof. ■

B. The Dominating-Set-Based Algorithm

Because the GBA is related to the size of a sensor field, the GBA may have higher time complexity when the sensor field gets wider. In the subsection, we propose another algorithm, termed the dominating-set-based algorithm (DSBA), which is independent of the size of the sensor field. In the DSBA, we first construct a graph $G(V, E)$ with a given set of sensors U in the sensor field. In the construction of the graph $G(V, E)$, for each sensor $u \in U$ located at (x, y), denoted by $u_{x,y}$, a node $v_{x,y}$ is added to V. In addition, an edge $(v_{x,y}, v_{i,j})$ is added to E if there exists two sensors $u_{x,y}$ and $u_{i,j}$ whose distance is not greater than R_c. It is clear that if there exists an edge $(v_{x,y}, v_{i,j}) \in E$, the sensor located at (i, j), that is, $u_{i,j}$, can be recharged and data collected when a mobile device visits $p_{x,y}$, where $p_{x,y}$ denotes a point positioned at (x, y). Therefore, the idea of the DSBA is to find a set of nodes D in G such that each node $v_{x,y} \in G$ is either in D or a neighbor node of one node in D. That is, the idea is to find a minimum dominating set D in G [25] to minimize the number of anchor points. Because the dominating set problem is NP-hard, a greedy method is proposed to find a dominating set D in G. Subsequently, the set of anchor points can then be constructed as the set of $p_{x,y}$ for all $v_{x,y} \in D$. Algorithm 2 shows the DSBA in details.

Algorithm 2 Dominating-Set-Based Algorithm (U)

1: Construct a graph $G(V, E)$ by U, where V is the set of nodes $v_{x,y}$ for all $u_{x,y} \in U$; $u_{x,y}$ denotes a sensor located at (x, y); and E is the set of edges $(v_{x,y}, v_{i,j})$ for all $u_{x,y}$ and $u_{i,j}$ in U with a distance not greater than R_c
2: $A \leftarrow \emptyset$
3: **while** $V \neq \emptyset$ **do**
4:     Select an element $v_{x,y}$ from V such that $v_{x,y}.nbr > v_{i,j}.nbr$ or ($v_{x,y}.nbr = v_{i,j}.nbr$ and $dist(s, u_{x,y}) \leq dist(s, u_{i,j})$) for all $v_{i,j} \in V - \{v_{x,y}\}$, where $v_{x,y}.nbr$ denotes the number of edges $(v_{x,y}, v') \in E$ for all $v' \in V$, and $dist(s, u_{x,y})$ denotes the distance between the data sink s and $u_{x,y}$
5:     $p_{x,y}.\omega \leftarrow v_{x,y}.nbr$
6:     $p_{x,y}.T \leftarrow \min(u_{x,y}.t_c, u_{x,y}.t_d)$
7:     **for** each edge $(v_{x,y}, v_{f,g}) \in E$ **do**
8:         Let $u_{f,g}$ denote the sensor located at position (f, g)
9:         **if** $p_{x,y}.T > \min(u_{f,g}.t_c, u_{f,g}.t_d)$ **then**
10:             $p_{x,y}.T \leftarrow \min(u_{f,g}.t_c, u_{f,g}.t_d)$
11:         **end if**
12:         $G \leftarrow G'$, where G' is a subgraph of G induced by $V - \{v_{f,g}\}$
13:     **end for**
14:     $G \leftarrow G'$, where G' is a subgraph of G induced by $V - \{v_{x,y}\}$
15:     $A \leftarrow A \cup \{p_{x,y}\}$
16: **end while**
17: **return** A;

In Algorithm 2, the construction of a graph $G(V, E)$ is illustrated in Line 1. In Lines $3 - 16$, the while loop is used to repeatedly find a dominator node $v_{x,y}$ from V until all selected dominator nodes can dominate the graph G. A node $v_{x,y} \in V$ is selected as a dominator node if $v_{x,y}$ has the largest nbr value in V, where $v_{x,y}.nbr$ denotes the number of edges $(v_{x,y}, v') \in E$ for all $v' \in V$. If k ($k > 1$) nodes have the same largest nbr value, the node $v_{x,y}$, which has $dist(s, u_{x,y}) \leq dist(s, u_{i,j})$ for all other $k - 1$ nodes $v_{i,j}$, is selected as a dominator node, where $dist(s, u_{x,y})$ denotes the distance between the data sink s and $u_{x,y}$. When a node $v_{x,y}$ is selected as a dominator node, $p_{x,y}$ is added to the set of anchor points A. In addition, $p_{x,y}.\omega$ is set to $v_{x,y}.nbr$, and $p_{x,y}.T$ is set to the minimum value of $\min(u.t_c, u.t_d)$ for all $u \in H$, where H denotes the set of $u_{x,y}$ and $u_{f,g}$ for all edges $(v_{x,y}, v_{f,g}) \in E$. G is updated as a subgraph of G induced by $V - \{v_{x,y}\} - \{v_{f,g}\}$ for all edges $(v_{x,y}, v_{f,g}) \in E$. When $V = \emptyset$, it implies that all nodes in the original G are dominated by the selected dominator nodes, and the set of anchor points A is thus constructed.

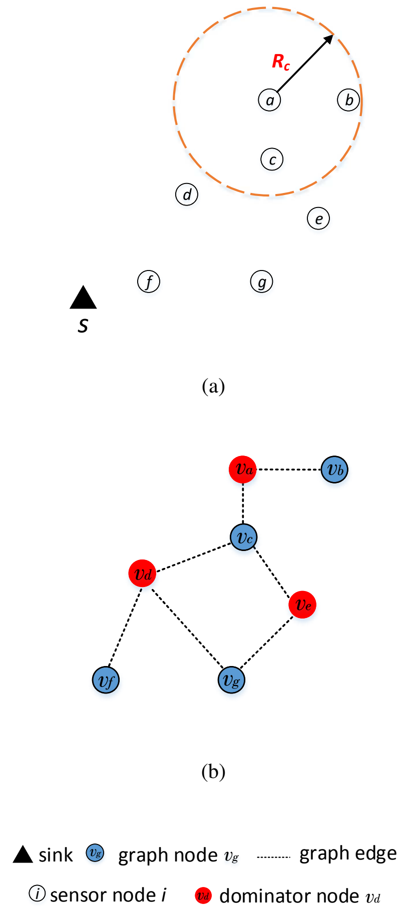

Fig. 3: Example of the dominating-set-based algorithm. (a) Sensors are deployed in a sensor field. (b) Anchor points are selected by the DSBA.

Take Fig. 3, for example. In Fig. 3(a), seven sensors, a, b, c, d, e, f, and g, are deployed in a sensor field. By Fig. 3(a), the DSBA first constructs a graph $G(V, E)$ as shown in Fig. 3(b), where $V = \{v_a,\ v_b,\ v_c,\ v_d,\ v_e,\ v_f,\ v_g\}$ and $E = \{(v_a, v_b),\ (v_a, v_c),\ (v_c, v_d),\ (v_c, v_e),\ (v_d, v_f),\ (v_d, v_g),\ (v_e, v_g)\}$. Because $v_c.nbr = v_d.nbr = 3$ and d is closer to the sink than c, v_d (or, the point on d) is first selected as a dominant node (or, anchor point). Then $G(V, E)$ is updated, where $V = \{v_a,\ v_b,\ v_e\}$, and E is updated to $\{(v_a, v_b)\}$. Subsequently, because $v_a.nbr = v_b.nbr = 1$ and a has smaller distance to the sink than b, v_a (or, the point on a) is selected as a dominant node (or, anchor point). Then $G(V, E)$ is updated, where $V = \{v_e\}$, and E is updated to $\emptyset$. Next, because v_e has the largest nbr value, $v_e.nbr = 0$, v_e (or, the point on e) is selected as a dominant node (or, anchor point). Therefore, the set of anchor points is constructed by the points on a, d, and e.

Theorem 3 shows an analysis of the time complexity of the DSBA.

Theorem 3: The time complexity of the DSBA is bounded in $O(n^3)$, where n denotes the number of deployed sensors.

Proof: Because there are at most n nodes and $\frac{n(n-1)}{2}$ edges in the graph $G(V, E)$, the construction of $G(V, E)$ requires at most $O(n^2)$. In the while loop of the DSBA, it is clear that at most n dominator nodes are selected. In addition, in each iteration, it requires at most $O(n^2)$ to find a suitable dominator node and obtain an induced subgraph from G. The while loop requires at most $O(n^3)$. Therefore, the time complexity of the DSBA is bounded in $O(n^2 + n^3) = O(n^3)$, which completes the proof. ■

C. Circle-Intersection-Based Algorithm

In the DSBA, only sensors' locations are used as anchor point candidates. However, when more locations are considered candidates, we could have fewer anchor points. Therefore, our idea here is to discover more candidates by the intersection points of circles (charging areas) centered at the sensors. Take Fig. 4, for example. In Fig. 4, d, f, and g are three sensors deployed in a sensor field. Let dg_1 and dg_2 be the points intersected by two circles with radii R_c centered, respectively, at sensors d and g. It is clear that when a mobile device visits dg_2, sensors d, f, and g can be recharged and data collected. Therefore, intersection points can be considered anchor point candidates, which motivates us to propose the circle-intersection-based algorithm (CIBA). In the CIBA, the sensors' locations and all possible intersection points of the circles centered at the sensors are considered anchor point candidates. Let $Z_{x,y}$ be a set of sensors that are within the circle with radius R_c centered at position (x, y). We have that the sensors in $Z_{x,y}$ will be recharged and data collected if the point positioned at (x, y), denoted by $p_{x,y}$, is visited by a mobile device. Let P be the set of all anchor point candidates. We then find a minimum set of points $A \subseteq P$ such that the union of $Z_{x,y}$ for all $p_{x,y} \in A$ is equal to the set of all sensors. Because the problem of finding a minimum set of points $A \subseteq P$ is equal to the set cover problem [26], which is a NP-hard problem, a greedy algorithm is used here. Algorithm 3 shows the CIBA in detail.

In Algorithm 3, ζ is computed to be the set of $Z_{x,y}$ for all candidates' locations (x, y), including the locations of sensors and intersection points, in Lines $1 - 10$. Because the idea is to find a minimum set of points A such that the union of $Z_{x,y}$ for all $p_{x,y} \in A$ is equal to the set of all sensors, that is, U, the while loop in Lines $13 - 28$ is used to repeatedly find a $Z_{x,y} \in \zeta$ until all sensors can be covered. In each iteration, a $Z_{x,y}$ is selected if it includes the largest uncovered sensors; that is, $|Z_{x,y}|$, where $|Z_{x,y}|$ denotes the size of $Z_{x,y}$. If k ($k > 1$) sets have the same size, the set $Z_{x,y}$, which has $dist(s, p_{x,y}) \leq dist(s, p_{i,j})$ for all other $k - 1$ sets $Z_{i,j}$, is selected, where $dist(s, p_{i,j})$ denotes the distance between the data sink s and point $p_{i,j}$. When a set $Z_{x,y}$ is selected, $p_{x,y}.\omega$ is set to the number of uncovered sensors, that is, $|Z_{x,y}|$. And $p_{x,y}.T$ is set to the minimum value of $\min(u.t_c, u.t_d)$ for all $u \in Z_{x,y}$. Then, $Z_{x,y}$ is deleted from ζ. Furthermore, for each $Z_{i,j} \in \zeta$, $Z_{i,j}$ is updated by $Z_{i,j} - Z_{x,y}$. In addition, C is updated by the union of C and $Z_{x,y}$. Finally, $p_{x,y}$ is added to the set of anchor points A. When C is equal to U, all sensors are covered, and A is thus constructed.

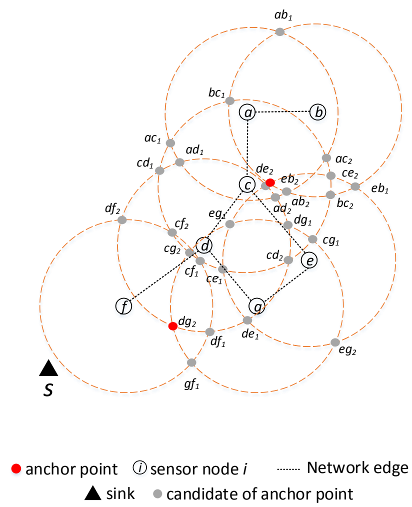

Fig. 4: Example of the circle-intersection-based algorithm.

Take Fig. 4, for example. In Fig. 4, the sensor nodes and all possible intersection points are given in a sensor field. It is clear that point eb_2 is first selected as an anchor point in A because a circle with radius R_c centered at eb_2 can cover sensors a, b, c, and e. Then, point dg_2 is selected as an anchor point in A because a circle with radius R_c centered at dg_2 can cover sensors d, f, and g. Because all sensors are covered, we have that $A = \{eb_2, dg_2\}$.

Theorem 4 shows an analysis of time complexity of the CIBA.

Theorem 4: The time complexity of the CIBA is bounded in $O(n^4)$, where n denotes the number of deployed sensors.

Proof: Because at most n sensors and $n(n-1)$ intersection points exist in the network, at most n^2 points are considered candidates. Because $Z_{x,y}$ has at most n sensors included for any $Z_{x,y} \in \zeta$, it requires at most $O(n \times n^2) = O(n^3)$ to construct ζ. In the while loop of the CIBA, it is clear that at least one sensor can be covered in each iteration. Therefore, at most n iterations are required in the while loop. Because at most n sensors are in $Z_{i,j}$ for any $Z_{i,j} \in \zeta$, and at most n^2 elements exist in ζ, it requires at most $O(n^3)$ to update the information of $Z_{i,j}$ for any $Z_{i,j} \in \zeta$ in each iteration. Therefore, the while loop requires at most $O(n^4)$. Then, we have that the time complexity of the CIBA is bounded in $O(n^3 + n^4) = O(n^4)$. ■

D. Mobile Device Scheduling Algorithm

In the subsection, a scheduling algorithm, termed the mobile device scheduling algorithm (MDSA), is proposed to schedule

Algorithm 3 Circle-Intersection Algorithm (U)

$\zeta \leftarrow \emptyset$
for any two sensors $u_{x,y}, u_{i,j} \in U$, where $u_{x,y}$ (or, $u_{i,j}$) denotes a sensor located at (x, y) (or, (i, j)) **do**
  Let $p_{b,d}$ and $p_{f,g}$ be the intersection points generated by two circles with radii R_c centered at (x, y) and (i, j), respectively, where $p_{b,d}$ (or, $p_{f,g}$) denotes a point positioned at (b, d) (or, (f, g))
  Let $Z_{b,d}$ and $Z_{f,g}$ be the sets of sensors that are within circles with radii R_c centered at $p_{b,d}$ and $p_{f,g}$, respectively.
  $\zeta \leftarrow \zeta \cup \{Z_{b,d}, Z_{f,g}\}$
end for
for any sensor $u_{x,y} \in U$ **do**
  Let $Z_{x,y}$ be the set of sensors that are within a circle with radius R_c centered at (x, y)
  $\zeta \leftarrow \zeta \cup \{Z_{x,y}\}$
end for
$C \leftarrow \emptyset$
$A \leftarrow \emptyset$
while C is not equal to U **do**
  Let $Z_{x,y}$ be the element in ζ such that $|Z_{x,y}| > |Z_{i,j}|$ or ($|Z_{x,y}| = |Z_{i,j}|$ and $dist(s, p_{x,y}) \leq dist(s, p_{i,j})$) for all $Z_{i,j} \in \zeta$, where $|Z_{i,j}|$ denotes the size of $Z_{i,j}$ and $dist(s, p_{i,j})$ denotes the distance between the data sink s and point $p_{i,j}$
  $p_{x,y}.\omega \leftarrow |Z_{x,y}|$
  $p_{x,y}.T \leftarrow \infty$
  for each sensor $u \in Z_{x,y}$ **do**
    if $p_{x,y}.T > \min(u.t_c, u.t_d)$ **then**
      $p_{x,y}.T \leftarrow \min(u.t_c, u.t_d)$
    end if
  end for
  $\zeta \leftarrow \zeta - \{Z_{x,y}\}$
  for each $Z_{i,j} \in \zeta$ **do**
    $Z_{i,j} \leftarrow Z_{i,j} - Z_{x,y}$
  end for
  $C \leftarrow C \cup Z_{x,y}$
  $A \leftarrow A \cup \{p_{x,y}\}$
end while
return A

minimum mobile devices to visit the anchor points generated by the GBA, the DSBA, or the CIBA. In the MDSA, our idea is first to select an anchor point p with maximum $p.\omega$ from A and add it into an empty anchor list L_1. Then, another anchor point q with the next higher $q.\omega$ is considered to be appended to the anchor list L_1. If two or more anchor points have the same next higher ω value, the point q that has $dist(e, q) + dist(q, s) \leq dist(e, i) + dist(i, s)$ for other points i, is considered, where e denotes the end point of the anchor list, s denotes the data sink s, and $dist(e, q)$ denotes the distance between points e and q. Here, the anchor point that is considered to be appended to an anchor list L is called L's successor candidate, which is formally defined in Def. 3. Let L'_1 be the anchor list L_1 appended by L_1's successor candidate q. If L'_1 is still a time-constrained anchor list, q is appended to L_1; otherwise, q is not appended to L_1, and we initialize another empty anchor list L_2 for the remaining anchor points. The new anchor list L_2 is then constructed as the same process for L_1. The process is repeated until all anchor points are included in anchor lists. Algorithm 4 shows the MDSA in details.

Definition 3: Let A be a set of anchor points, and L be a time-constrained anchor list. The $q \in A$ is said L's successor candidate if $q.\omega > x.\omega$, or ($q.\omega = x.\omega$ and $dist(p, q) + dist(q, s) \leq dist(p, x) + dist(x, s)$) for any $x \in A - \{q\}$, where p denotes the end point in L, s denotes the data sink s, and $dist(q, s)$ denotes the distance between points q and s.

Algorithm 4 Mobile Device Scheduling Algorithm (A)

$\Gamma \leftarrow \emptyset$
$i \leftarrow 1$
while A is not empty **do**
  Initialize a list L_i
  Let p be an element in A with maximum $p.\omega$
  Append p to the end of the list L_i
  $A \leftarrow A - \{p\}$
  while L'_i is a time-constrained anchor list, where L'_i is the anchor list L_i appended by L_i's successor candidate q **do**
    Append q to the end of the list L_i
    $A \leftarrow A - \{q\}$
  end while
  $\Gamma = \Gamma \cup \{L_i\}$
  $i = i + 1$;
end while
return Γ;

Theorem 5 shows an analysis of time complexity of the MDSA.

Theorem 5: The time complexity of the MDSA is bounded in $O(\theta^2)$, where θ denotes the number of anchor points.

Proof: Because at most θ anchor lists are constructed, the outer while loop requires at most θ iterations. In addition, because the inner while loop requires at most $O(\theta)$ to find a successor candidate, we have that the time complexity of the MDSA is bounded in $O(\theta * \theta) = O(\theta^2)$, which thus completes the proof. ∎

IV. Performance Evaluation

In this section, simulations were used to evaluate the performance of the proposed methods. In the simulations, 100 to 1000 sensors were uniformly distributed in a 100×100 square area, and the data sink was located in a corner of the area. Each mobile device had a charging range R_c, the movement speed $\nu = 1$, and the charging time $\delta = 30$. The t_c and t_d of each sensor were randomly chosen from the interval $[300 - \sigma, 300 + \sigma]$. To compare the proposed methods, we first evaluated the GBA, the DSBA, and the CIBA in terms of the number of generated anchor points, in Section IV-A. In addition, the GBA, the DSBA, and the CIBA were applied with the MDSA, which are called the GBA+MDSA, the DSBA+MDSA, and the CIBA+MDSA, hereafter, were evaluated in other subsections in terms of

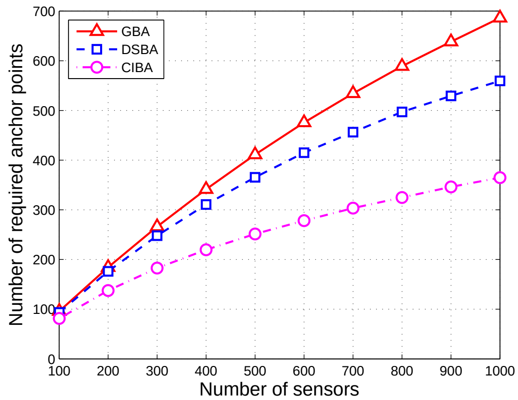

Fig. 5: The number of anchor points required by the GBA, the DSBA, and the CIBA in networks that have 100 to 1000 sensors.

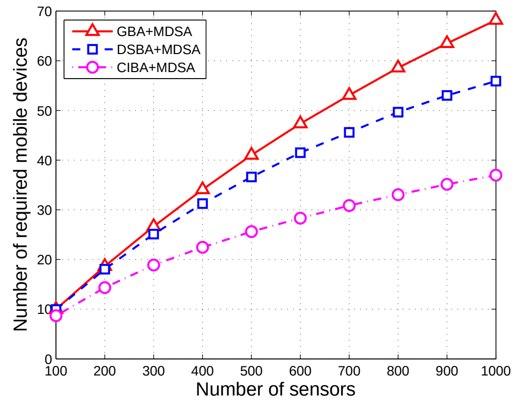

Fig. 6: The number of mobile devices required by the GBA+MDSA, the DSBA+MDSA, and the CIBA+MDSA in networks that have 100 to 1000 sensors.

the number of required mobile devices and the total required moving distance. In the following subsections, the data were obtained by averaging the data of 100 simulations.

A. *Number of Required Anchor Points*

Fig. 5 shows the simulation results in terms of the number of required anchor points with the number of sensors ranging from 100 to 1000 when $R_c = 2$. It is clear that the higher the number of sensors, the higher the number of anchor points required by the GBA, the DSBA, or the CIBA. This is because more sensors must be recharged and data collected. In addition, the GBA has a higher number of required anchor points than the DSBA and the CIBA. This is because more center points of grids are selected as anchor points when the sensors are uniformly distributed in the sensor field. Note that the CIBA has the lowest number of required anchor points. This is because more candidates can be considered for selecting anchor points.

B. *Number of Required Mobile Devices*

Fig. 6 shows the simulation results concerning the number of required mobile devices with the number of sensors ranging from 100 to 1000 when $R_c = 2$ and $\sigma = 50$. As observed in Fig. 5, the number of mobile devices required by the GBA+MDSA, the DSBA+MDSA, or the CIBA+MDSA increases as the number of sensors increases, because more sensors are required to be recharged and collected data. Note that the number of mobile devices required by the GBA+MDSA (or the CIBA+MDSA) is higher (or lower) than the others. This is because the anchor points required by the GBA+MDSA (or the CIBA+MDSA) is higher (or lower) than the others, as described in Section IV-A.

Fig. 7 shows the simulation results in terms of the number of required mobile devices with R_c ranging from 2 to 10 when the number of sensors is 500 and $\sigma = 50$. It is clearly that the longer the charging range R_c, the lower the number of mobile devices required by the GBA+MDSA, the DSBA+MDSA, or the CIBA+MDSA. This stems from the

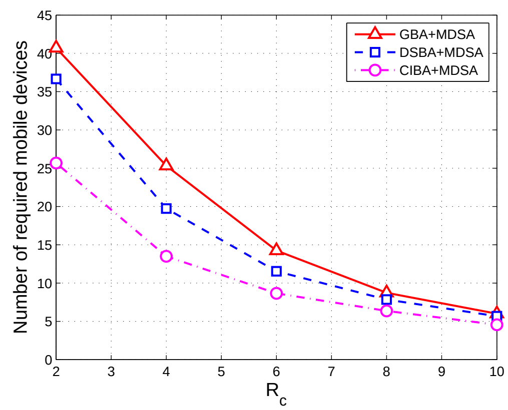

Fig. 7: The number of mobile devices required by the GBA+MDSA, the DSBA+MDSA, and the CIBA+MDSA in networks with R_c ranging from 2 to 10.

fact that more sensors may be covered when an anchor point is visited, and thus, fewer anchor points are required by the GBA+MDSA, the DSBA+MDSA, or the CIBA+MDSA. Also note that the CIBA+MDSA outperforms the GBA+MDSA and the DSBA+MDSA because the CIBA+MDSA requires fewer anchor points than the others.

Fig. 8 shows the simulation results concerning the number of required mobile devices with σ ranging from 30 to 150 when the number of sensors is 500 and $R_c = 2$. Note that the higher the σ value, the higher the number of mobile devices required by the GBA+MDSA, the DSBA+MDSA, or the CIBA+MDSA. This is because the t_c (or, t_d) value may decrease for some sensors u such that more additional mobile devices are thus required to recharge urgent sensors. Note that the CIBA+MDSA provides better performance than the others in terms of the number of required mobile devices because the CIBA+MDSA requires fewer anchor points than the others, as observed in the previous simulation.

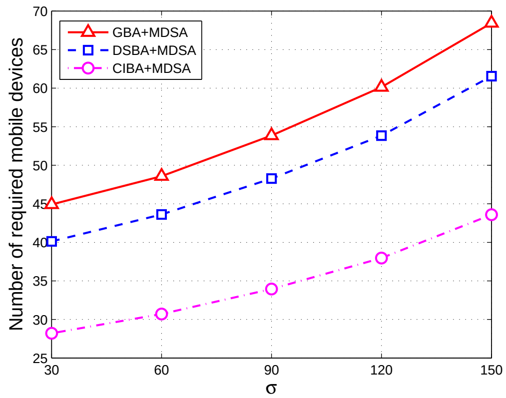

Fig. 8: The number of mobile devices required by the GBA+MDSA, the DSBA+MDSA, and the CIBA+MDSA in networks with σ ranging from 30 to 150.

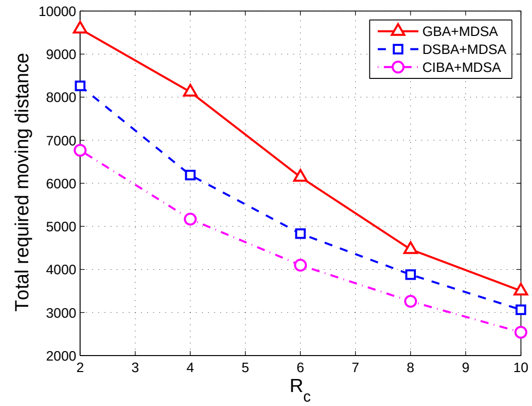

Fig. 10: The total moving distance required by the GBA+MDSA, the DSBA+MDSA, and the CIBA+MDSA in networks with R_c ranging from 2 to 10.

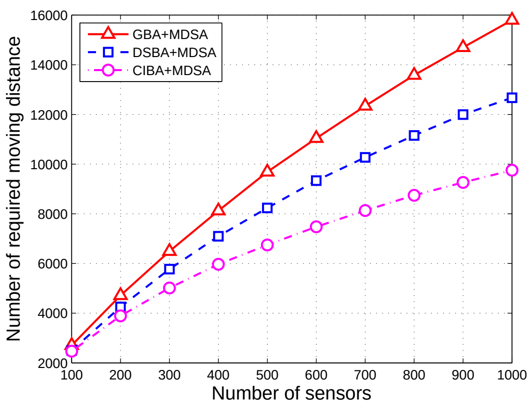

Fig. 9: The total moving distance required by the GBA+MDSA, the DSBA+MDSA, and the CIBA+MDSA in networks that have 100 to 1000 sensors.

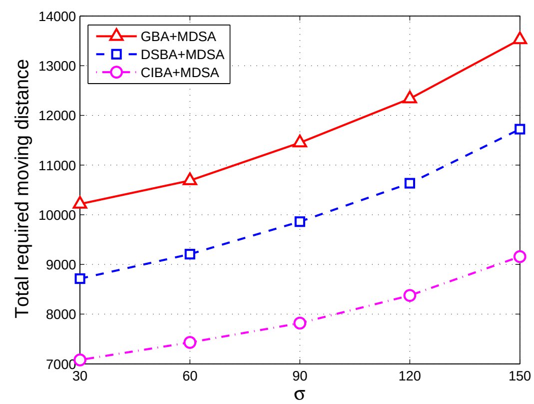

Fig. 11: The total moving distance required by the GBA+MDSA, the DSBA+MDSA, and the CIBA+MDSA in networks with σ ranging from 30 to 150.

C. Total Required Moving Distance

Fig. 9 shows the simulation results concerning the total required moving distance with the number of sensors ranging from 100 to 1000 when $R_c = 2$ and $\sigma = 50$. Note that the larger the number of sensors, the higher the total moving distance. This is because more mobile devices are required to recharge sensors as observed in Fig. 6. In addition, the CIBA+MDSA has a lower total moving distance than the others because the CIBA+MDSA requires fewer mobile devices than the others.

Fig. 10 shows the simulation results in terms of the total required moving distance with R_c ranging from 2 to 10 when the number of sensors is 500 and $\sigma = 50$. Clearly, the longer the charging range, the lower the total required moving distance. This is because fewer anchor points are required to be visited. In addition, Fig. 11 shows the simulation results concerning the total required moving distance with σ ranging from 30 to 150 when the number of sensors is 500 and $R_c = 2$. Note that the higher the σ value, the higher the required total moving distance. This is because more mobile devices are scheduled for urgent sensors, as observed in Section IV-B.

V. Conclusion

In the paper, the problem of scheduling minimum mobile devices to periodically recharge and collect data from sensors subject to the limited charging range, electric capacity, and memory storage, such that the network lifetime can be guaranteed to be prolonged without limits, termed the Periodic Energy Replenishment and Data Collection (PERDC) problem, was introduced. We first showed that the PERDC problem is NP-complete. In addition, three algorithms, including the grid-based algorithm (GBA), the dominating-set-based algorithm (DSBA), and the circle-intersection-based algorithm (CIBA), were proposed to find a set of anchor points. Based on the generated anchor points, the mobile device scheduling algorithm (MDSA) was proposed to schedule minimum mobile devices for energy replenishment and data collection. Moreover, theoretical analysis showed that the GBA had the lowest time complexity compared with the DSBA and the CIBA when the side length of the sensor field and the sensor's charging range are constant.

In the simulations, the performance of the GBA, the DSBA, and the CIBA was evaluated in terms of the number of generated anchor points. Simulation results showed that the CIBA had a lower number of generated anchor points. In addition, the GBA, the DSBA, and the CIBA were applied with the MDSA, which were called the GBA+MDSA, the DSBA+MDSA, and the CIBA+MDSA, and were evaluated in terms of the number of required mobile devices and the total required moving distance. The simulation results showed that although the CIBA+MDSA had higher time complexity than the others, the CIBA+MDSA provided lower number of required mobile devices and shorter total required moving distance than the others. In addition, the simulation results showed that the GBA+MDSA and the DSBA+MDSA provided comparable performance when fewer sensors existed in the sensor field.

VI. Acknowledgements

This work was supported by the Ministry of Science and Technology under Grant MOST 104-2221-E-151-014.

References

[1] R. Mittal and M. P. S. Bhatia, "Wireless sensor networks for monitoring the environmental activities," in *Proceedings of IEEE ICCIC*, Dec 2010, pp. 1–5.

[2] Y. Lei, Y. Zhang, and Y. Zhao, "The research of coverage problems in wireless sensor network," in *Proceedings of IEEE WNIS*, Dec 2009, pp. 31–34.

[3] C.-T. Cheng, C. Tse, and F. Lau, "A delay-aware data collection network structure for wireless sensor networks," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 699–710, March 2011.

[4] M. Zhao, M. Ma, and Y. Yang, "Mobile data gathering with multiuser mimo technique in wireless sensor networks," in *Proceedings of IEEE GLOBECOM*, Nov 2007, pp. 838–842.

[5] L. Mo, Y. He, Y. Liu, J. Zhao, S.-J. Tang, X.-Y. Li, and G. Dai, "Canopy closure estimates with greenorbs: Sustainable sensing in the forest," in *Proceedings of ACM ENSS*, New York, NY, USA, 2009, pp. 99–112.

[6] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," in *Proceedings of ACM WSNA*, New York, NY, USA, 2003, pp. 115–121.

[7] L. Lin and H. Lee, "Distributed algorithms for dynamic coverage in sensor networks," in *Proceedings of ACM PODC*, New York, NY, USA, 2007, pp. 392–393.

[8] I. Khemapech, I. Duncan, and A. Miller, "Energy preservation in environmental monitoring wsn," in *Proceedings of IEEE SUTC*, June 2010, pp. 312–319.

[9] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embedded Computing Systems*, vol. 6, no. 4, Sep. 2007.

[10] J. Hsu, S. Zahedi, A. Kansal, M. Srivastava, and V. Raghunathan, "Adaptive duty cycling for energy harvesting systems," in *Proceedings of IEEE ISLPED*, Oct 2006, pp. 180–185.

[11] R.-S. Liu, P. Sinha, and C. Koksal, "Joint energy management and resource allocation in rechargeable sensor networks," in *Proceedings of IEEE INFOCOM*, March 2010, pp. 1–9.

[12] Y. Peng, Z. Li, W. Zhang, and D. Qiao, "Prolonging sensor network lifetime through wireless charging," in *Proceedings of IEEE RTSS*, Nov 2010, pp. 129–139.

[13] C. Wang, Y. Yang, and J. Li, "Stochastic mobile energy replenishment and adaptive sensor activation for perpetual wireless rechargeable sensor networks," in *Proceedings of IEEE WCNC*, April 2013, pp. 974–979.

[14] C. Wang, J. Li, F. Ye, and Y. Yang, "Multi-vehicle coordination for wireless energy replenishment in sensor networks," in *Proceedings of IEEE IPDPS*, May 2013, pp. 1101–1111.

[15] M. Zhao, J. Li, and Y. Yang, "Joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks," in *Proceedings of IEEE ITC*, 2011, pp. 238–245.

[16] J. Li, M. Zhao, and Y. Yang, "Ower-mdg: A novel energy replenishment and data gathering mechanism in wireless rechargeable sensor networks," in *Proceedings of IEEE GLOBECOM*, Dec 2012, pp. 5350–5355.

[17] S. Guo, C. Wang, and Y. Yang, "Mobile data gathering with wireless energy replenishment in rechargeable sensor networks," in *Proceedings of IEEE INFOCOM*, April 2013, pp. 1932–1940.

[18] S. He, J. Chen, F. Jiang, D. K. Y. Yau, G. Xing, and Y. Sun, "Energy provisioning in wireless rechargeable sensor networks." in *Proceedings of IEEE INFOCOM*, 2011, pp. 2006–2014.

[19] L. Fu, P. Cheng, Y. Gu, J. Chen, and T. He, "Minimizing charging delay in wireless rechargeable sensor networks," in *Proceedings of IEEE INFOCOM*, April 2013, pp. 2922–2930.

[20] F. Jiang, S. He, P. Cheng, and J. Chen, "On optimal scheduling in wireless rechargeable sensor networks for stochastic event capture," in *Proceedings of IEEE MASS*, Oct 2011, pp. 69–74.

[21] P. Cheng, S. He, F. Jiang, Y. Gu, and J. Chen, "Optimal scheduling for quality of monitoring in wireless rechargeable sensor networks," *IEEE Trans. Wireless Communications*, vol. 12, pp. 3072–3084, June 2013.

[22] S. Nath, "Energy efficient sensor data logging with amnesic flash storage," in *Proceedings of IEEE IPSN*, April 2009, pp. 157–168.

[23] F. Ordonez and B. Krishnamachari, "Optimal information extraction in energy-limited wireless sensor networks," *IEEE Selected Areas in Communications*, vol. 22, no. 6, pp. 1121–1129, Aug 2004.

[24] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. Princeton, NJ, USA: Princeton University Press, 2007.

[25] L. Ruan, H. Du, X. Jia, W. Wu, Y. Li, and K.-I. Ko, "A greedy approximation for minimum connected dominating sets," *Theoretical Computer Science*, pp. 325 – 330, 2004.

[26] S. Yoo, "A novel mask-coding representation for set cover problems with applications in test suite minimisation," in *Proceedings of IEEE SSBSE*, Sept 2010, pp. 19–28.