

Received February 3, 2020, accepted February 16, 2020, date of publication February 21, 2020, date of current version March 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2975635

# Efficient Wireless Charging Pad Deployment in Wireless Rechargeable Sensor Networks

JINGJING CHEN<sup>1,2</sup>, CHANG WU YU<sup>1,2,3</sup>, AND WEN OUYANG<sup>3</sup>

<sup>1</sup>College of Physics and Mechanical and Electrical Engineering, Longyan University, Longyan 364000, China

<sup>2</sup>Chung Hua University, Hsinchu 30012, Taiwan

<sup>3</sup>Department of Computer Science and Information Engineering, Chung Hua University, Hsinchu 30012, Taiwan

Corresponding author: Chang Wu Yu (cwyu@chu.edu.tw)

This work was supported in part by the Natural Science Foundation of Fujian Province, China, under Grant 2019J01795, and in part by the Young Teacher Education and Research Project of Fujian Province under Grant JAT170551.


**ABSTRACT** The rapid development of wireless power transfer technology brings forth innovative vehicle energy solutions and breakthroughs utilizing *wireless sensor networks* (WSNs). In most existing schemes, *wireless rechargeable sensor networks* (WRSNs) are generally equipped with one or more *wireless charging vehicles* (vehicles) to serve *sensor nodes* (SNs). These schemes solve the energy issue to some extent; however, due to off-road and speed limitations of vehicles, some SNs still cannot be charged in time, negatively affecting the lifetime of the networks. Our work proposes a new WRSN model equipped with one *wireless charging drone* (drone) with a constrained flight distance coupled with several *wireless charging pads* (pads) deployed to charge the drone when it cannot reach the subsequent stop. Our model solves this charging issues effectively and overcomes energy capacity limitations of the drone. Thus, a wireless charging pad deployment problem is formulated, which aims to apply the minimum number of pads so that at least one feasible routing path can be established for the drone to reach every SN in a given WRSN. Four feasible heuristics, three based on graph theory and one on geometry, are proposed for this problem. In addition, a novel drone scheduling algorithm, the shortest multi-hop path algorithm, is developed for the drone to serve charging requests with the assistance of pads. We examine the proposed schemes through extensive simulations. The results compare and demonstrate the effectiveness of the proposed schemes in terms of network density, region size and maximum flight distance.

**INDEX TERMS** Wireless power transfer, wireless charging drone, wireless charging pad, sensor node, wireless rechargeable sensor networks.

## I. INTRODUCTION

*Wireless sensor networks* (WSNs) with advanced technologies capable of obtaining and collecting relevant information quickly are widely used in many applications requiring monitoring of systems, examples being metropolitan operation, military, and environmental monitoring [1]. Due to the difficulty and cost of replacing batteries of sensor nodes (SNs) in relatively-inaccessible environments, the energy problem becomes a critical issue for WSNs. Recent advances in *wireless power transfer* (WPT) have produced a novel network named the *wireless rechargeable sensor network* (WRSN), which has quickly gained extensive scrutiny from many researchers.

Most recent research considers charging sensor nodes using *wireless charging vehicles* (vehicles) equipped with

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

high-capacity batteries and WPT devices. In these schemes, a vehicle can move close to a SN and charge it wirelessly without physical contact. Previous work related to vehicles involve utilizing multi-functional vehicles [2], simultaneous charging of multiple SNs [3], designing mobile charging protocols [4], or planning optimal collaborative charging schedules of multiple vehicles to elevate performance of sensor networks [5]. These schemes can solve the energy problem to some extent. However, two essential drawbacks of vehicles are overlooked:

(1) Off-road limitation: The movement of vehicles is limited in hazardous environments such as a hill or an island where roads are sparse. Furthermore, if a vehicle encounters an obstacle or a road fork, it cannot complete the assigned charging task in time because a vehicle cannot cross the barrier to approach sensor nodes (Fig. 1).

(2) Traveling speed limitation: Due to the limited traveling speed of vehicles, not every node can be charged in time

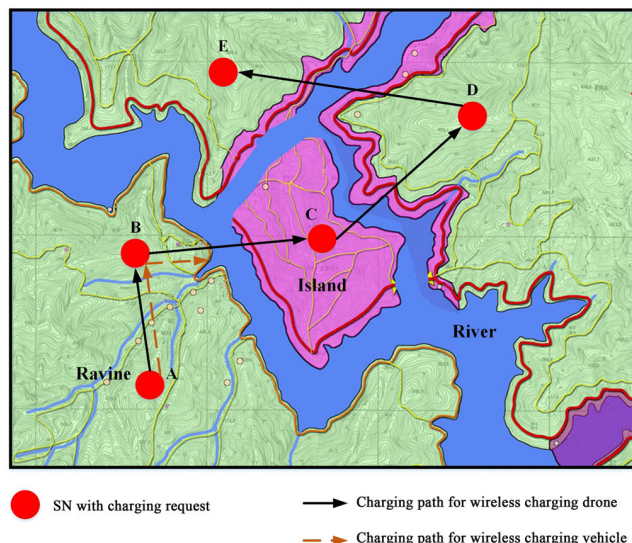


FIGURE 1. Moving scenario of the proposed new model.

when a number of charging requests occur simultaneously. This may result in death of some sensor nodes in the WRSN.

Therefore, it is clear that the two major roadblocks to the success of WRSNs are the off-road and traveling speed limitations. Recently, some studies have considered wireless charging of SNs via *wireless charging drones* (drones) in WRSNs [14]. When the drone power is close to depletion, the drone must return to the *base station* (BS) to replenish its power, causing the drone to fly back and forth frequently between sensor nodes and the base station. Moreover, there has been a breakthrough [6]–[13] in the field of drone wireless charging technology. An automatic landing *wireless charging pad* (pad) for a drone can be implemented, and that, coupled with a high-power and high-efficiency WPT system, can automatically charge a drone within a short period to prevent necessary and frequent return trips to the base station [6].

To combat the off-road and traveling speed limitations, in this work, we propose a novel sensor network model with one single drone and several pads. The new scheme assumes that the charging requests from sensor nodes are sent to the base station, and that the base station computes and plans the optimal scheduling trip for the drone. When receiving a charging mission, the drone departs from the base station to charge sensor nodes according to the assigned schedule. During the trip, if the energy of the drone is below a predefined threshold, it must fly to a neighboring pad for energy replenishment before visiting the next node. When the mission is complete, the drone returns to the base station to wait for the next possible mission.

However, the limited battery capacity continues to constrain the application of drones in the WRSN. Since the battery capacity of drones is limited, the flight distance of the drone is also limited when compared to vehicles. If a drone with an assigned charging tour needs to travel a lot, it may need to land on multiple pads to satisfy all charging requests during a mission.

For example, as shown in Fig. 1, a drone can fly directly from node A to node B, then to node C, and so forth. Moreover, since the speed of a drone can reach 100 to 289 miles per hour [15], by adopting drones instead of vehicles, one can alleviate the traveling speed limitation within the network. Evidently, the effective deployment of pads, especially geographically, becomes a significant issue when establishing the new charging system. If insufficient number of pads are deployed, some nodes will fall outside of the flight range of the drone and will not be charged in time. On the other hand, deploying a surplus of pads simply to ensure maximal coverage increases unnecessary cost.

To resolve the dilemma of efficiency versus cost, in this work, we have formulated the optimization problem for pad deployment using the minimum number of pads to establish, for each SN, at least one feasible routing path from the base station in a given sensor network. Moreover, four feasible heuristic algorithms including *minimum set cover like* (MSC), *tree node coloring* (TNC), *graph node coloring* (GNC) and *disk cover* (DC) for this deployment problem are designed. To further verify the advantages of the proposed algorithms, we develop a novel drone scheduling algorithm, the *shortest multi-hop path* (SMHP) scheduling, to find the optimal charging sequence for the drone after deploying pads.

Extensive simulations and discussions were conducted to evaluate the performance of the proposed WRSN model and the proposed four heuristic algorithms. In summary, the main contributions of this work are listed as follows:

- 1) A new systemic model of using drones for charging sensor nodes in WRSN is proposed. The new charging system is expected to alleviate both off-road limitation and traveling speed limitation.
- 2) The incorporation of pads into WRSNs for the drone to overcome the limited energy issue. The pad deployment problem is formulated and four heuristic algorithms are proposed to determine the positions and the number of required pads in the deployed area.
- 3) The first paradigm computes the expected number of wireless charging pads used in DC for a given deployed rectangle in a systematic way.
- 4) A charging schedule algorithm for the drone based on the new model for deployment of pads is developed.

The remainder of the paper is organized as follows. Section II contains a review of the relevant existing literature. Our novel network model and related terminologies are introduced in Section III. Details of the proposed heuristic algorithms for pad deployment are provided in Section IV, and Section V presents the proposed charging scheduling scheme. Section VI describes and discusses the simulation results. Finally, Section VII concludes this paper.

## II. RELATED WORK

In this section, we review previous work in WRSN and wireless charging technology for drones. First, we review previous work concerning schemes aimed towards resolving

the energy problem in WRSNs. Next, we review literature about wireless charging technology for drones to understand the state of the field today.

### A. RESOLVING ENERGY PROBLEM

Most previous works employ one or more MCs to charge nodes wirelessly in the network. To achieve a better network performance, how to schedule the MCs effectively is an important key to the design of the schemes. At present, charging schedules are mainly divided into two categories: periodical charging-based schemes and on-demand charging based schemes.

Periodical charging-based schemes consider that MCs periodically charge the SNs wirelessly by travelling along a certain path in the sensing area. Zhao *et al.* [2] proposed to utilize multi-functional MCs for wireless charging and data collection. Xie *et al.* [3] considered a MC periodically travelling along an optimized path inside a WSN while simultaneously charging multiple SNs by WPT technology, which addressing the charging scalability problem in a dense WSN. Fu *et al.* [18] planned an optimal charging path for the mobile reader with the objective of minimizing total charging delay. They also proposed a novel energy-synchronized mobile charging (ESync) protocol in [4], which reduces the travel distance of MCs and the charging delay of sensor nodes at the same time. Based on the diversity of nodes' energy consumption, they constructed a set of nested TSP charging tours by selecting energy-hungry nodes. Guo *et al.* [19] considered various source of energy consumption and time-varying characteristics of energy replenishment, then studied a WRSN framework of joint wireless energy replenishment and anchor-point based mobile data gathering problem which was formulated into a network utility maximization problem. Furthermore, they proposed a distributed algorithm, through which a MC dynamically adjusts its optimal sojourn time at each anchor point. Shu *et al.* [20] first identified the optimal velocity problem when an MC periodically travels along a predefined trajectory. Meanwhile, they formulated the problem of arbitrarily shaped irregular trajectories in a two-dimension space for maximizing the charged energy in SNs. Liu *et al.* [21] first considered the disjointed state of nodes for scheduling a single MC and introduced a novel metric named the criticality index (CI) to quantify criticality of nodes. Then they selected SNs with the maximal total CIs in the charging tour with the limited of traveling distance of MC. Liu *et al.* [22] presented a security disjoint routing-based verified message scheme for improving data arrival rate and transmission delay in solar energy harvesting wireless sensor networks. Liu *et al.* [23] used a mobile sink along a fixed trajectory to collect sensor data from rendezvous point. They proposed a quick convex hull-based rendezvous planning scheme for full connectivity and minimizing energy consumption of multihop communication by constructing a near-convex hull trip.

The on-demand based schemes solve the energy problem in WRSNs by optimal path planning. However, the schemes

assume that the energy consumption state of SNs is known in advance by the MCs. He *et al.* [24] laid the theoretical foundation for an on-demand mobile charging problem, where MC charges the SNs which send charging requests when their energy falls below certain lower threshold. Then they proposed nearest-job-next with preemption (NJNP) algorithm which considers the node having the nearest distance to be charged first. Lin *et al.* proposed several different charging scheduling schemes in their work. In [25], they proposed double warning thresholds with double preemption charging scheme. In the scheme, two warning thresholds and comparison rules that determine scheduling priority for charging requests were formulated. In addition, two preemption mechanisms were designed to deal with real characteristics of WRSN. They proposed a temporal and distastial priority charging scheduling algorithm in [26], which considered the distance between nodes and the MC and the arrival time of charging requests the two factors to achieve better performance in terms with throughput, response time and successful charging rate. In [27], they developed a primary and passer-by scheduling (P<sup>2</sup>S) algorithm, in which the MC charged nearby nodes while charging the primary nodes in the charging scheduling. In [28], they proposed an Optimal Path Planning Charging (OPPC) scheme, which evaluates the schedulability of charging tasks and enables the schedulability by discarding several unimportant nodes. In [5], they developed a temporal-spatial charging algorithm for the multiple MCs collaborative charging issue for minimizing the number of dead nodes while maximizing energy efficiency. Kaswan *et al.* [29] first presented a linear programming formulation for the problem of scheduling one MC in an on-demand WRSN. Then they proposed an efficient charging scheduling method based on a gravitational search algorithm presented with a novel agent representation scheme and an efficient fitness function that considers both temporal and spatial priorities of SNs. Wang *et al.* [30] proposed a new model which allows the mobile charger to perform data gathering and energy replenishment simultaneously, consuming energy through short communication distances. Xu *et al.* [31] proposed a novel WRSN model equipped with a MC with separable charger array. The MC can unload chargers onto the location of SNs without waiting charging. Although the on-demand charging schemes are more feasible than other schemes for a complex network environment, mobile chargers continue to be limited by their energy and movement in large scale WRSNs.

### B. CHARGING TECHNOLOGY FOR DRONES

In this subsection, we mainly introduce wireless charging technology for drones and application areas of drones (e.g., charging nodes). Although drones have advantages in space, their applications are limited by the battery capacity. With the development of wireless charging technology for drones, a lot of work considers using drones to charge SNs wirelessly in sensor networks.

Chae *et al.* [6] suggested an automatic landing system of a drone for the round-the-clock surveillance. They also



implemented the prototype of the system that was tested to verify the feasibility. Simic *et al.* [7] proposed to apply WPT to charge drones. Choi *et al.* [8] proposed a fully automatic drone charging station based on a wireless charging technique. Their proposed system allows the drone to land on the platform without precise position, and then the station detects the position of drones and provides charging service. Wang and Ma [9] developed a wireless magnetic resonant power transfer system which can flexibly and safely supply energy to drone. Campi *et al.* [10] proved the feasibility of a light-weight but high-efficiency WPT charging system applied to a demonstrative drone. In [11], they also proposed a high-power and high-efficiency WPT system based on magnetic resonant coupling, which can automatically recharge the battery of a drone and meet the drone requirements in terms of power and landing precision. Aldhafer *et al.* [12] described a system that allows flying drones to be charged in mid-air. This system enables a 10W micro-drone to be charged whilst hovering in the vicinity of a pad. Mostafa *et al.* [13] developed a wireless battery charging system for a drone using the capacitive power transfer technology, which provides a wide charging area for a drone to land. Therefore, the charging system can make drones obtain power from pads more efficiently. Long *et al.* [14] introduced a new networking paradigm named the energy neutral internet of drones (enIoD). In their paradigm, recharging stations with strong energy harvesting are deployed on the ground to charge drones so that drones can extend flight range via wireless power transfer without human intervention. The paradigm provides technical support for pads in the paper. Zorbas and Douligeris [32] explored the possibility of charging the SNs wirelessly by drones in WRSN when the SNs are deployed in a harsh environment. They proposed a lower complexity drone-positioning heuristic algorithm that operates on hardware with limited processing power. Simulation results showed that their algorithm performs close to the optimal solution obtained by using Integer Linear Programming (ILP). To support the drone service scenarios such as surveillance, remote sensing and mapping, Choi *et al.* [15] designed oneM2M standard platform architecture which supports real-time data delivery for drone management system. The system improves end-to-end delay performance in different network environments.

Different from the previous schemes, we propose a novel scheme which uses one drone, with the help of several pads, to charge sensor nodes in WRSN. In addition, we design a charging scheduling for drones based on the new WRSN model.

### III. SYSTEM MODEL AND TERMINOLOGIES

In this section, we describe the architecture and components of the novel WRSN model in detail, followed by notations and terminologies.

The newly-proposed WRSN model, constituted of a set of randomly deployed rechargeable SNs, a drone, several

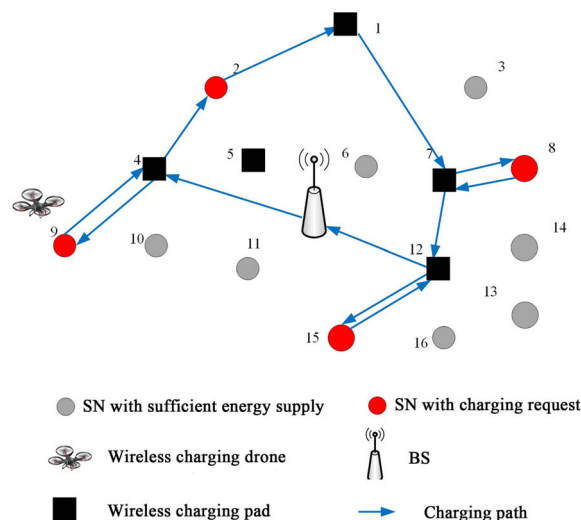


FIGURE 2. Proposed novel WRSN model.

pads, and a base station is shown in Fig. 2. The network assumptions are given as follows:

(1) The SNs are homogeneous and stationary in the sensing area. Each SN has a *unique identity* (ID) and limited capacity of energy. The BS knows the ID and the coordinates of each SN.

(2) Every SN sends a charging request to the BS when its residual energy falls below some pre-defined threshold. The drone can charge only one SN at one time and the power of pad is assumed to be unlimited.

(3) The BS is located in the center of sensing area as a data sink and the service station of the drone. The base station evaluates the charging schedule of the drone. When the schedule is finalized, the BS informs the drone with the coordinates of the requesting SNs and pads. Then the drone flies to and charges SNs according to the schedule. After finishing the assignment, it returns to the BS, waiting for the next charging mission.

(4) The drone charges SNs one by one via direct flies. However, when its energy is below a predefined threshold, the drone needs to fly to one of its neighboring pads to replenish its energy.

(5) Pads are stationary in the given deployed area. They connect with the drone automatically and charge it wirelessly when the drone lands on them. We also assume that every pad can only support the landing of one drone at any moment.

Completely different from the conventional model using vehicles, the new system model uses one single drone for charging SNs and deploys several pads for charging the drone. Instead of vehicles, the drone performs a charging mission with a pre-planned schedule, but it must replenish energy on a pad when its energy is below a predefined threshold. Compared with the traditional model using vehicles, adopting drones avoids off-road limitation and alleviates traveling speed. Thus, the new model has advantages in hazardous environments and large networks. A drone not only conducts

TABLE 1. Symbols and definitions.

| Symbol      | Definition   |
|-------------|--|
| $N$         | Number of SNs  |
| $m$         | Number of pads   |
| $n$         | Number of charging requests in the request queue   |
| $s_i$       | The $i$ th SN  |
| $V_{WCD}$   | Speed of drone (the speed is assumed to be fixed)  |
| $d_{i,j}$   | Flight distance from $s_i$ to $s_j$  |
| $d_{max}$   | Maximum flight distance of drone   |
| $d_{rem}$   | Remaining flight distance of drone   |
| $p_{WCD}$   | Average flight power consumption of drone per unit of time                                 |
| $e_{WCD}$   | Maximum energy capacity of drone   |
| $e_i$       | Power consumption when charging $s_i$  |
| $S$         | Set of SNs   |
| $P$         | Set of pads  |
| $r_{queue}$ | Request queue: the array of IDs of the SNs which have sent the charging requests to the BS |

the charging task for sensor nodes more accurately, but also reduces the execution time of a mission [33].

The symbols used throughout this paper are described in Table 1. Furthermore, we elaborate some necessary definitions in the next section.

#### IV. DEPLOYMENT OF PADS

In the new charging model, the deployment of pads significantly affects the performance of the WRSN. Therefore, the arrangement of as few pads as possible within the sensing area without deferring the charging task becomes the key problem studied in the paper.

**Definition 1 (The Pad Deployment Problem):** given a base station and a set of SNs with their coordinates in a plane, the task is to deduce the minimum number of pads along with their coordinates so that for each SN there exists at least one drone flight path connecting BS to this SN.

Suppose the current flight path of the drone is  $s_0 (= BS) \rightarrow s_1 \rightarrow s_2 \rightarrow \dots \rightarrow s_k$ , the remaining maximum flight distance of a drone  $d_{rem}$  is defined as follows:

$$d_{rem} = \frac{e_{WCD} - \sum_{i=1}^k \left[ \frac{d_{i,i+1}}{V_{WCD}} \times p_{WCD} + e_i \right]}{p_{WCD}} \times V_{WCD} \quad (1)$$

The pad deployment problem asks for a selection of  $m$  locations in the sensing area and the placement of one pad on every selected location so that the deployed pads support at least one connected flying path from BS to every SN for a drone, since it may need to fly to every sensor with the help of pads.

In general, a drone needs to fly close to a sensor node issuing a charging request in order to charge it. Due to the limitation of the drone flight distance [15], after charging a sensor node, the residual energy of a drone must ensure it can reach the nearest pad to replenish energy. Without the need to consider any charging request, we set  $d_{max}$  as the maximum flight distance for every drone. We formally define an available charging flight path below.

**Definition 2:** Given a fixed flight distance limit  $d_{max}$ , a flight path  $BS = p_0 \rightarrow s_{0,1} \rightarrow s_{0,2} \rightarrow \dots \rightarrow s_{0,o(0)} \rightarrow$

$p_1 \rightarrow s_{1,1} \rightarrow s_{1,2} \rightarrow \dots \rightarrow s_{1,o(1)} \rightarrow p_2 \rightarrow s_{2,1} \rightarrow s_{2,2} \rightarrow \dots \rightarrow s_{2,o(2)} \rightarrow \dots \rightarrow p_k \rightarrow s_{k,1} \rightarrow s_{k,2} \rightarrow \dots \rightarrow s_{k,o(k)} \rightarrow p_{k+1} = p_0 = BS$  is called *available* if and only if  $d(p_i, s_{i,1}) + d(s_{i,1}, s_{i,2}) + \dots + d(s_{i,o(k)}, p_{i+1}) \leq d_{max}$  (for  $0 \leq i \leq k$ ), where  $P = BS \cup \{p_1, p_2, \dots, p_m\}$  is a pad set and  $\{s_1, s_2, \dots, s_N\}$  contains sensor nodes that have issued a charging request in the charging path.

Evidently, an available flight path ensures that a drone can finish its assigned charging tasks when bounded by a given maximum flight distance. In this work, we try to deploy the minimum number of pads so that for every sensor node  $s$  requesting a charging service, there always exists an available flight path from the BS to  $s$ .

**Definition 3: (The Pad Deployment With Flight Distance Limit Problem):** given a BS, a set of SNs with their coordinates in a plane, and the maximum flight distance, the problem is to find the minimum number of pads with their coordinates so that there exists one available drone flight path connecting BS to every SN.

Evidently, the pad deployment with the flight distance limit problem becomes a new and simplified version of the pad deployment problem. Initially, it seems difficult to model the pad deployment with the flight distance limit problem because there are many possible feasible available paths for a specific requested sensor. However, the following solution can tackle it indirectly in an efficient way.

**Definition 4:** Given a BS with  $N$  SNs  $\{s_1, s_2, \dots, s_N\}$  in the plane, the *pad cover problem* is to discover a minimum set of pad  $\{p_1, p_2, \dots, p_m\}$  such that the following two conditions are satisfied:

**Condition (1):** For every  $s_i$  in  $\{s_1, s_2, \dots, s_N\}$ , there exists at least a  $p_j$  in  $\{p_1, p_2, \dots, p_m\}$  such that  $d(s_i, p_j) \leq d_{max}/2$ . Here we say that  $p_j$  covers  $s_i$ .

**Condition (2):** The constructed graph  $G = (V, E)$  is connected, where  $V = \{BS = p_0, p_1, p_2, \dots, p_m\}$  and  $(p_i, p_j) \in E$  if and only if  $p_i \in V$  and  $p_j \in V$  and  $d(p_i, p_j) \leq d_{max}$ . Here  $d$  indicates the Euclidean distance function.

When we use a disk with a radius  $r$  to indicate the service area of a deployed pad, let  $r = d_{max}/2$ . Condition (1) ensures for every sensor  $s$  there exists at least a nearby pad  $p_i$  that supports additional energy for a drone to fly to  $s$  for the assigned charging task and then fly back to  $p_i$  again. Intuitively, the pad cover problem uses the minimum number of disks to cover all sensor nodes.

On the other hand, Condition (2) ensures the reachability of every deployed pad for a drone. Since the distances between arbitrary two pads is less than  $d_{max}$ , a drone can fly from one pad to another one directly, get recharged, and then fly to next destination. By repeating above process, the drone can reach every pad due to the connectivity of the underlying graph (flight map). Then we have the next theorem.

**Theorem 1:** The pad deployment with a flight distance limit problem is equivalent to the pad cover problem.

**Proof of Theorem 1:** For a pad deployment  $P = \{p_1, p_2, \dots, p_m\}$ , we want to show that there is an available flight path from BS to every sensor for the pad deployment taking into

account the flight distance limit problem if and only if set  $P = \{p_1, p_2, \dots, p_m\}$  satisfies the two conditions in the pad cover problem.

Suppose that there is a set  $P = \{p_1, p_2, \dots, p_m\}$  satisfying the two conditions in the pad cover problem. We would need to show that there always exists an available flight path from BS to each sensor  $s$ . Since each sensor  $s$  is covered by at least one pad  $p$  in  $P$  by Condition (1), we have  $d(s, p) \leq d_{max}/2$ . Moreover, since the constructed graph  $G = (V, E)$  is connected (where  $V = \{\text{BS} = p_0, p_1, p_2, \dots, p_m\}$  and  $(p_i, p_j) \in E$  if and only if  $p_i \in V$  and  $p_j \in V$  and  $d(p_i, p_j) \in d_{max}$ ), there is a flight from BS to  $p$ :  $\text{BS} = p_0 \rightarrow p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_k = p \rightarrow \dots \rightarrow p_2 \rightarrow p_1 \rightarrow p_0 = \text{BS}$  where  $d(p_i, p_{i+1}) \leq d_{max}$  for  $i = 0$  to  $k - 1$ . Combining  $\text{BS} = p_0 \rightarrow p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_k = p \rightarrow \dots \rightarrow p_2 \rightarrow p_1 \rightarrow p_0 = \text{BS}$  with  $p \rightarrow s \rightarrow p$ , we obtain an available flight:  $\text{BS} = p_0 \rightarrow p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p \rightarrow s \rightarrow p \rightarrow \dots \rightarrow p_2 \rightarrow p_1 \rightarrow p_0 = \text{BS}$ .

On the other hand, suppose that there always exists an available flight path from BS to every sensor, the least used pads form a set  $P = \{p_1, p_2, \dots, p_m\}$ . We need to prove that  $P = \{p_1, p_2, \dots, p_m\}$  satisfies the two conditions in the pad cover problem. The proof is shown below by contradiction.

Suppose that Condition (1) is not satisfied. There is a sensor  $s$  which is not covered by any  $p_j$  in  $\{p_1, p_2, \dots, p_m\}$ . That means for all  $p_j$  in  $\{p_1, p_2, \dots, p_m\}$ , we have  $d(s_i, p_j) > d_{max}/2$ . However, since there is an available flight path from BS to  $s$  and back to BS, we have  $\text{BS} = p_0 \rightarrow s_{0,1} \rightarrow \dots \rightarrow s_{0,o(0)} \rightarrow p_1 \rightarrow s_{1,1} \rightarrow \dots \rightarrow s_{1,o(1)} \rightarrow p_2 \rightarrow s_{2,1} \rightarrow \dots \rightarrow s_{2,o(2)} \rightarrow \dots \rightarrow \dots \rightarrow p_i \rightarrow s_{i,1} \rightarrow \dots \rightarrow s_{i,j} = s \rightarrow \dots \rightarrow s_{i,o(i)} \rightarrow p_{i+1} \rightarrow \dots \rightarrow p_k \rightarrow s_{k,1} \rightarrow \dots \rightarrow s_{k,o(k)} \rightarrow p_{k+1} = p_0 = \text{BS}$ . Moreover,  $d(p_i, s_{i,1}) + d(s_{i,1}, s_{i,2}) + \dots + d(s_{i,j-1}, s_{i,j} = s) + d(s_{i,j} = s, s_{i,j+1}) + \dots + d(s_{i,o(k)}, p_{i+1}) \leq d_{max}$  by Definition 2. That means either  $d(p_i, s_{i,1}) + d(s_{i,1}, s_{i,2}) + \dots + d(s_{i,j-1}, s_{i,j} = s) \leq d_{max}/2$  or  $d(s_{i,j} = s, s_{i,j+1}) + \dots + d(s_{i,o(k)}, p_{i+1}) \leq d_{max}/2$ . By triangular inequality, we have either  $d(p_i, s) \leq d_{max}/2$  or  $d(s, p_{i+1}) \leq d_{max}/2$ . In other words, either  $p_i$  covers  $s$  or  $p_{i+1}$  covers  $s$ . A contradiction occurs. As a result, Condition (1) holds.

Suppose that Condition (2) is not satisfied. The constructed graph  $G = (V, E)$  is disconnected. Where  $V = \{\text{BS} = p_0, p_1, p_2, \dots, p_m\}$  and  $(p_i, p_j) \in E$  if and only if  $p_i \in V$  and  $p_j \in V$  and  $d(p_i, p_j) \leq d_{max}$ . Since  $G$  is disconnected, assume that  $p_x$  and  $p_y$  belong to different connected components of  $G$ . That implies that there is no path connecting  $p_x$  and  $p_y$  in  $G$ . We claim that there exists no available flight path visiting  $p_x$  (or  $p_y$ ) starting from BS. Otherwise, there is an available flight path visiting  $p_x$  and another available flight path visiting  $p_y$ , all originating from BS. By combining these two paths, we can find an available flight path connecting  $p_x$  and  $p_y$ . This also implies that there is a path connecting  $p_x$  and  $p_y$  in  $G$ . A contradiction occurs.

Theorem 1 assures that we can solve the pad deployment with the flight distance limit problem by finding a solution of the pad cover problem. Later in Section IV-B to IV-F,

we will show that it is easier to design efficient pad deployment schemes for the pad cover problem than for the pad deployment with flight distance limit problem.

### A. PAD DEPLOYMENT PROBLEM WITH ITS FORMULA

According to Theorem 1, the formulation of the aforesaid problem can be defined as follows.

$$\text{Minimize : } |P = \{p_1, p_2, \dots, p_m\}| \quad (2)$$

Subject to :

$$\sum_{p_j \in P \cup \{p_0\}} c_{ij} \geq 1, \quad \forall s_i \in S \quad (3)$$

$$\sum_{p_i \in P \cup \{p_0\}} e_{ij} \geq 1, \quad \forall p_j \in P \cup \{p_0\} \quad (4)$$

$$e_{ij} = \begin{cases} 1, & \text{if } d(p_i, p_j) \leq d_{max} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$c_{ij} = \begin{cases} 1, & \text{if } d(s_i, p_j) \leq d_{max}/2 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$\sum_{\substack{\Omega \text{ is a permutation of} \\ \text{a subset of } P \cup \{p_0\} - \{p_i, p_j\}}} \left( e_{i, \Omega(1)} \times \left( \prod_{k=1}^{|\Omega|-1} e_{\Omega(k), \Omega(k+1)} \right) \times e_{\Omega(|\Omega|), j} \right) \geq 1, \quad \forall p_i, p_j \in P \cup \{p_0\} \quad (7)$$

where  $p_i \neq p_j$  when  $i \neq j$ , and  $p_0 = \text{BS}$ .

Constraints (3) and (6) indicate that every SN is covered by at least one pad (Condition (1) in Definition 4). Constraints (4) and (5) state that each pad is connected to at least another pad. Constraints (7) implies that at least one available path exists between any two pads.

In the following subsections, we consider a *simplified pad deployment problem*, which considers deploying pads (with flight distance limit to drone) only on the locations of SNs. Since we assume that the deployed SNs form a connected network, there always exists a solution for a simplified pad deployment problem. Specifically, four schemes are proposed and discussed. Three schemes including the MSC scheme, the TNC scheme and GNC scheme are used to solve the simplified pad deployment problem while the DC scheme works by placing pads on specified locations, which may or may not be SNs in the deployed area.

The pad cover problem originally allows pads to be deployed to any point in the defined area. However, in this work, algorithms MSC, TNC, and GNC only use sensor location as potential deployment positions for the pads. The simplified pad deployment problem is quite similar to the geometric connected dominating set [34], which is known to be NP-complete. That strongly hints that all the pad deployment problem considered in this work might all be NP-complete. Thus, we consider designing heuristic algorithms for pad deployment. Since our model considers a large number of

sensor nodes to be uniformly deployed and dispersed in the interested area, it's believed that sensors locations are suitable potential positions for the pad.

### B. MINIMUM SET COVER LIKE SCHEME

The first scheme is called the *minimum set cover like* (MSC) scheme because it modifies the minimum set cover technique to solve the simplified pad deployment problem. The minimum set cover problem is essentially a bipartite graph problem, described as follows. Let a bipartite graph  $H = (A, B, E)$  contain two disjointed vertex nodes  $A$  and  $B$ . Nodes  $u, v$  for each edge  $(u, v)$  in  $E$  belong to  $A$  and  $B$  respectively; here node  $v$  is said to be covered by node  $u$ . Assume that for each node  $v$  in  $B$ , there is at least some node  $u$  in  $A$  covers  $v$ . The *set cover problem* is finding a minimum subset  $C$  of set  $A$  such that each node in set  $B$  can be covered by at least a node in set  $C$ . However, the simplified pad deployment problem is different from the traditional set cover problem. The difference is that the simplified pad problem selects a minimum subset  $C$  of set  $A$  so that not only each node in set  $B$  is covered by a node in set  $C$  (Condition (1)), but also each node in this selected subset  $C$  should be connected to each other (Condition (2)). Since the set cover problem is NP-hard [35], [36], we believe that the simplified pad problem is also NP-hard.

The intuitive idea of the MSC algorithm is that first, one node is greedily selected (from set  $A$ ) that covers most uncovered nodes (in set  $B$ ) to join set  $C$  (initialized to be empty). Then it is removed from set  $A$  and the nodes covered are removed from set  $B$ . Next, another node in  $A$  is chosen that is connected to a node in  $C$  within two hops and covers most uncovered nodes in set  $B$ . Repeat the above steps until all the nodes in set  $B$  are covered. The detailed steps are described in Algorithm MSC.

Evidently, Algorithm MSC assures that Condition (1) in Definition 4 is satisfied by Step 4 and Step 5. Since every node  $p_i(s_i)$  in  $C$  is connected within two hops to another node  $p_j(s_j)$  in  $C$ , we have a path  $p_i \rightarrow s_k \rightarrow p_j$ . Moreover, since  $d(p_i, s_k) \leq d_{max}/2$  and  $d(s_k, p_j) \leq d_{max}/2$ ,  $d(s_i, s_j) \leq d(s_i, s_k) + d(s_k, s_j) \leq d_{max}$ , Condition (2) in Definition 4 is thus satisfied. As a result, Algorithm MSC obtains a feasible solution for the simplified pad problem by Theorem 1.

The complexity of MSC is analyzed as follows. Let  $n$  denote the number of sensor nodes. Step 1 takes  $O(1)$  time for initializing node sets, and Step 2 takes  $O(n^2)$  time for constructing a bipartite graph. Step 3 takes  $O(n^2)$  time to compute the neighbor node set for every sensor node. Step 4 and Step 5 takes  $O(n)$  time to choose the locations of deployed pads. At most, MSC repeats Step 5 at most  $O(n)$  times. Totally, Algorithm MSC takes  $O(n^2)$  time.

### C. TREE\_NODE COLORING SCHEME

In this subsection, we introduce the *tree node coloring* (TNC) scheme to solve the simplified pad deployment problem. To obtain a near optimal solution, we combine the concept

### Algorithm 1 MSC Algorithm

Input:  $G = (V, E_G)$  and  $V = \{s_1, s_2, \dots, s_N\}$ , there is an edge  $(s_i, s_j) \in E_G$  if and only if  $d(s_i, s_j) \leq d_{max}/2$ .

Output: Set  $C$  containing a set of sensor nodes where is to be deployed with pads.

Step 1: Let node set  $A = V$ , node set  $B = V$ , and node set  $C = \emptyset$ .

Step 2: Construct a bipartite graph  $H = (A, B, E_H)$  by putting an edge in  $E_H$  between  $s_i$  of  $A$  and  $s_j$  of  $B$  if they are adjacent to each other in  $G$  (i.e., if  $d(s_i, s_j) \leq d_{max}/2$ ).

Step 3: Compute  $N(v)$  for every node  $v$  in  $A$  of  $H = (A, B, E_H)$ , where  $N(v)$  denotes the set of vertices in  $B$  which are adjacent to  $v$  in  $H$ .

Step 4: Select a node  $s_i$  into  $C$  from set  $A$  which has the largest  $\|N(s_i)\|$  value; (i.e.,  $C \leftarrow s_i$ );  
Remove node  $s_i$  (from  $A$ ) and the nodes covered by node  $s_i$  from  $B$ ;  
i.e.,

$$\begin{aligned} A &= A/\{s_i\}, \\ B &= B/\{s_i\}, \\ B &= B/N(v). \end{aligned}$$

Step 5: Update  $N(v)$  for every node  $v$  in  $A$  of  $H = (A, B, E_H)$ , where  $N(v)$  denotes the set of vertices in  $B$  which are adjacent to  $v$  in  $H$ . Select a node  $s_i$  into  $C$  from set  $A$  which is connected to a node in  $C$  with a two hop path in  $H$  and has the largest number of uncovered nodes in set  $B$ . That is,  $C \leftarrow s_i$ .  
Remove node  $s_i$  from  $A$  and the nodes covered by node  $s_i$ ; i.e.,

$$\begin{aligned} A &= A/\{s_i\}. \\ B &= B/\{s_i\}. \\ B &= B/N(v). \end{aligned}$$

Step 6: Repeat Step 5 until  $B = \emptyset$ .

Step 7: Output  $C$ .

of maximal degree selection with a node-coloring technique so that a spanning tree is obtained.

A node is white when it's neither covered nor chosen as a pad yet. A grey node denotes that it has been covered by some pad, and a black node means it has been chosen as a pad. Next, Definition 5 helps to understand the proposed algorithm.

*Definition 5:* Given a grey node or white node  $v$ , the set of white neighbor nodes of  $v$  is represented with  $NW(v) = \{\forall u | u \text{ is colored white, } u \in N(v)\}$ .

The idea of the scheme is to greedily choose a grey node with maximum  $|NW(v)|$  in each step until there are no white nodes in the graph. Initially all nodes in set  $V$  are in white, spanning tree set  $T = \emptyset$ . The first step computes  $NW^*$  of all nodes, colors the node  $v$  with maximum  $|NW(v)|$  black and colors its white neighbors grey, then uses the node  $v$  as the root of the spanning tree  $T$ , and join its neighbors in  $T$ . Step 2 computes  $NW(x)$  for each white node  $x$  in  $NW^*$  of grey nodes, colors the node  $x$  with maximum  $|NW(x)|$  black



and colors its white neighbors grey, then joins it and its grey neighbors nodes in  $T$ . Step 2 is repeated until there are no white nodes in  $V$ . We describe the specific steps in Algorithm TNC.

---

**Algorithm 2** TNC Algorithm
 

---

Input:  $G = (V, E)$ , and  $V = \{s_1, s_2, \dots, s_N\}$ ,  $(s_i, s_j) \in E$  if and only  $d(s_i, s_j) \leq d_{max}/2$ , a tree  $T = \emptyset$ .  
 Output: a black node set  $C$ .

Step 1: Compute  $NW(v)$  of each node  $v$  in  $V$ .  
     Choose the node  $v$  with  $\max NW(v)$ .  
      $v$ . color=black.  
      $T \leftarrow v$ .  
   for each node  $u$  in  $NW(v)$   
      $u$ . color=grey.  
      $T \leftarrow u$ .  
      $root\_node = v$ .

Step 2: for each node  $u$  with color grey in  $N(root\_node)$   
   for each node  $x$  in  $NW(u)$   
     Compute  $NW(x)$  of each node  $x$  with color white in  $V$ .  
     choose the node  $x$  with  $\max NW(x)$ .  
      $NW(v) = NW(v) \setminus \{u\}$ .  
      $x$ . color=black.  
      $T \leftarrow x$ .  
     for each node  $y$  in  $NW(x)$   
        $y$ . color=grey.  
        $T \leftarrow y$ .  
    $root\_node = v$ .

Step 3: Repeat Step 2 until there are no nodes with white color in  $V$ .

Step 4: Output the nodes with color black in  $T$ .

---

The complexity of TNC is analyzed as follows. Step 1 takes  $O(n^2)$  time to compute the neighbor node set of every nodes. Step 2 takes  $O(n^3)$  time for choosing the locations of pads. TNC repeats Step 2 to Step 3 at most  $O(n)$  times. Totally, Algorithm TNC takes  $O(n^4)$  time.

#### D. GRAPH NODE COLORING SCHEME

In this subsection, we use *graph node coloring* (GNC) scheme to solve the simplified pad deployment problem by combining the concept of maximum degree selection with node-coloring technique as shown in Section IV-C. To be more efficient, GNC selects pads differently from TNC.

The notations of node colors is the same as in TNC. GNC greedily chooses a node that colors most white nodes in each step until there are no more white nodes in the graph. However, to find a proper node more efficiently, this scheme expands the scope for selecting pads. It chooses a node as pad by comparing  $|NW(*)|$  value of every white node which is  $d_{max}/2$  to  $d_{max}$  away from any pad.

In GNC scheme, initially two different and independent connected graphs  $G_1$  and  $G_2$  with SNs are constructed, where  $G_1$  is constructed based on Condition (1), and  $G_2$  based on

Condition (2) of Definition 4. That is, GNC constructs  $G_1 = (V_1, E_1)$  by putting an edge between  $s_i$  and  $s_j$  in  $V_1$  if and only if  $d(s_i, s_j) \leq d_{max}/2$ . GNC constructs  $G_2 = (V_2, E_2)$  by putting an edge between  $s_i$  and  $s_j$  in  $V_2$  if and only if  $d_{max}/2 \leq d(s_i, s_j) \leq d_{max}$ .

At the beginning, all nodes in the two graphs are colored in white. Then, GNC computes the set  $NW(v)$  of every node  $v$  in  $G_1$  and computes the set  $N(v)$  of every node  $v$  in  $G_2$ , where  $N(v)$  denotes the set of white vertices in  $V_2$  which are adjacent to  $v$  in  $G_2$ . Select the node  $v$  with maximum  $|NW(v)|$  nodes in  $G_1$ , then color  $v$  black and all  $NW(v)$  grey. Then the same coloring operation is applied to the corresponding nodes in  $G_2$ ; also add  $v$  into set  $C$ . Next, for every white node  $u$  in  $N(v)$  of a node  $v \in C$  in  $G_2$ ,  $NW(u)$  in  $G_1$  is computed. Select a node  $x$  with maximum  $|NW(x)|$  value, color  $x$  in black and  $NW(x)$  grey in  $G_1$  and  $G_2$ . Also include  $x$  into set  $C$ . Repeat above steps until there are no white node in  $G_1$ . The detailed steps are described in Algorithm GNC.

---

**Algorithm 3** GNC Algorithm
 

---

Input: A sensor node set  $V = \{s_1, s_2, \dots, s_N\}$  with a node set  $C = \emptyset$ .  
 Output: a selected node set  $C$ .

Step 1: Construct a graph  $G_1 = (V_1, E_1)$  by putting an edge between  $s_i$  and  $s_j$  in  $V_1$  if and only  $d(s_i, s_j) \leq d_{max}/2$ .  
 Construct a graph  $G_2 = (V_2, E_2)$  by putting an edge between  $s_i$  and  $s_j$  in  $V$  if and only  $d_{max}/2 \leq d(s_i, s_j) \leq d_{max}$ .

Step 2: Compute  $NW(v)$  for every white node  $v$  in  $V_1$  of  $G_1$ .  
 Compute  $N(v)$  of each uncolored node  $v$  in  $V_2$  of  $G_2$ .  
 Choose a white node  $v$  in  $G_2$  with the maximum  $NW(v)$  value in  $G_1$ .  
 $v$ . color=black.  
 $C \leftarrow v$ .  
 for each node  $u$  in  $NW(v)$   
    $u$ . color=grey.

Step 3: for each node  $v$  in set  $C$   
   for each white node  $x$  in  $N(v)$  of  $G_2$   
     Compute  $NW(x)$  of each node  $x$  in  $V_1$  of  $G_1$ .  
     Choose the node  $x$  with the maximum  $NW(x)$  value.  
      $x$ . color=black.  
      $C \leftarrow x$ .  
   for each node  $y$  in  $NW(x)$   
      $y$ . color=grey.

Step 4: Repeat Step 3 until there is no white nodes in  $G_1$ .  
 Step 5: Output the nodes in black color (i.e., set  $C$ ).

---

Algorithm 3 chooses a node as the pad in  $N(v)$  of a node  $v \in C$  in  $G_2$ , which ensures pads obtained by Algorithm 3 is connected (i.e., every pad is at least  $d_{max}/2$  to  $d_{max}$  away from one other pad.). Moreover, all sensor nodes are covered



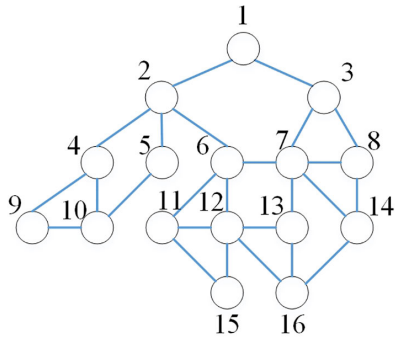


FIGURE 3. An example of the given network graph.

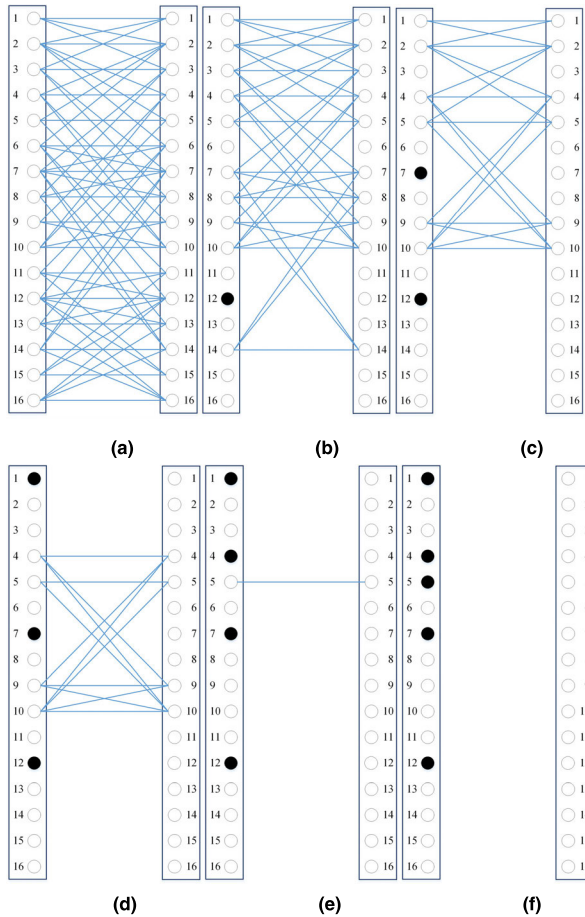


FIGURE 4. Steps when applying the MSC scheme: (a) construction of a bipartite graph, (b) selection of node 12 as the location of a wireless charging pad, (c) choose node 7, (d) choose node 1, (e) choose node 4, and (f) choose node 5.

by the pads, because all nodes are colored grey (i.e. are covered) or black (i.e., are chosen as pads) after the execution of Algorithm 3.

Step 1 in GNC takes  $O(n^2)$  time for constructing two different graphs. Step 2 takes at most  $O(n)$  time to locate the first pad. Step 3 takes  $O(n^2)$  time to locate the rest of pads. GNC repeats Step 3 at most  $O(n)$  times until all the node are colored. Thus, the time complexity of Algorithm GNC is  $O(n^3)$ .

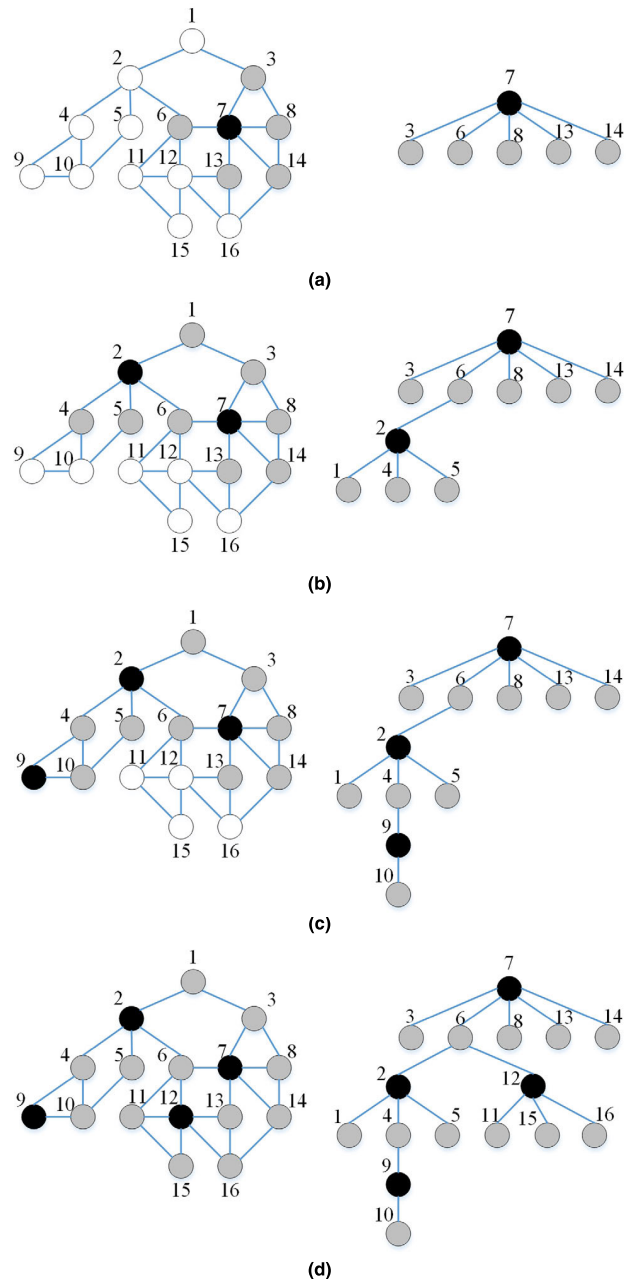
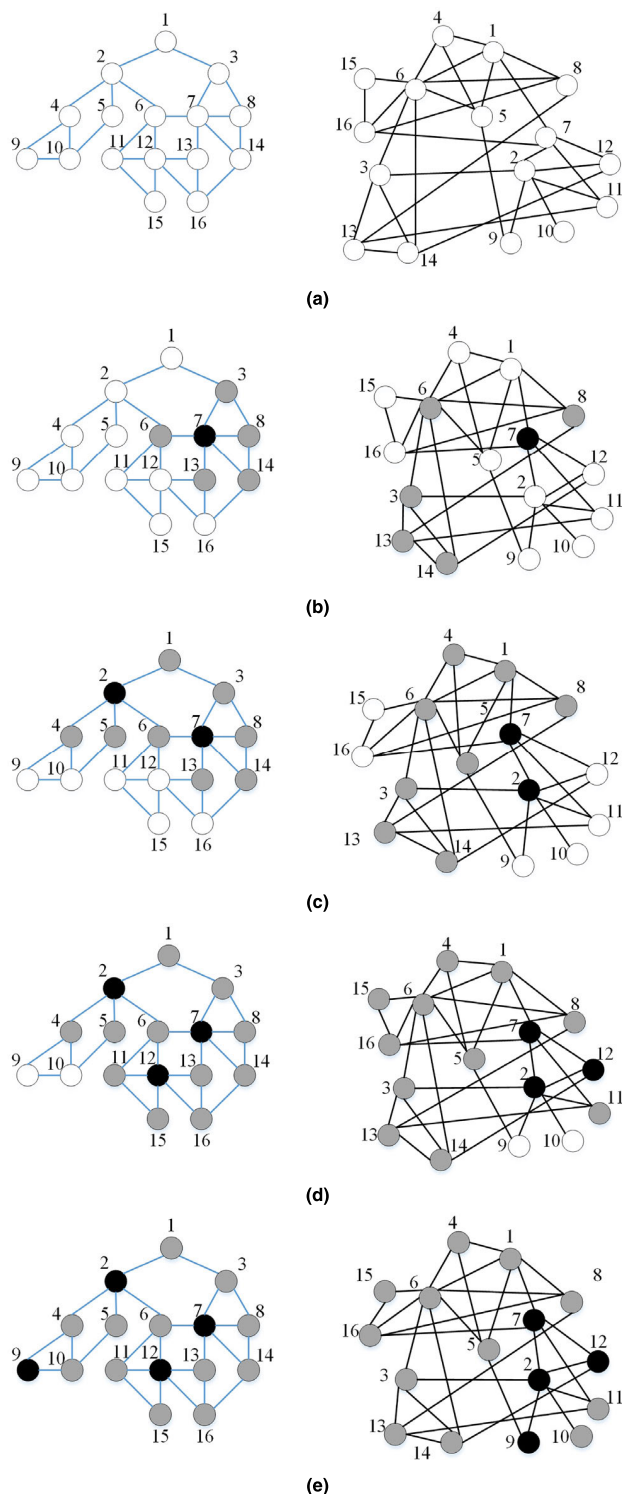


FIGURE 5. Steps when applying TNC scheme: (a) select node 7 as the location of a wireless charging pad, (b) choose node 2, (c) choose node 9, (d) choose node 12.

E. CASE STUDY

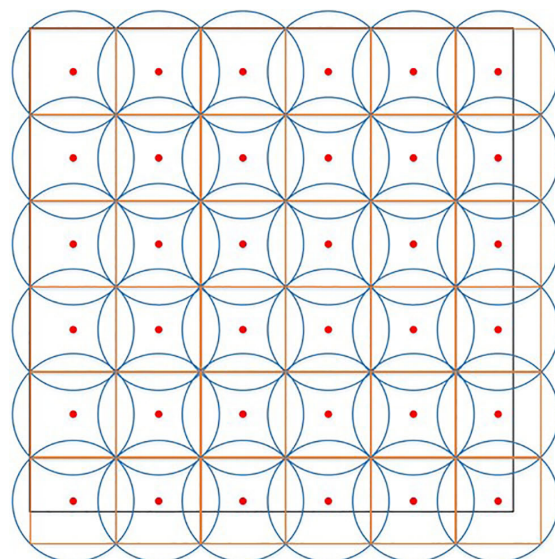
In this subsection, we present a few examples to demonstrate the intuitive ideas of the MSC, TNC and GNC schemes through a network graph containing 16 nodes (Fig. 3). In the graph, the IDs of the nodes are labeled beside the nodes and a solid line represents a distance between any two nodes that is less than or equal to  $d_{max}/2$ . The execution scenario of the MSC, TNC and GNC algorithm are shown in Fig. 4, Fig. 5 and Fig. 6, respectively. The resulting pad sets for MSC, TNC and GNC are node sets  $\{1, 4, 5, 7, 12\}$ ,  $\{2, 7, 9, 12\}$  and  $\{2, 7, 9, 12\}$ , respectively.



**FIGURE 6.** Steps when applying the GNC scheme: (a) construct  $G_1$  and  $G_2$ , (b) choose node 7 as the location of a wireless charging pad, (c) choose node 2, (d) choose node 12, (e) choose node 9.

**F. DISK COVER SCHEME**

The *disk cover* (DC) scheme solves the problem of pad deployment quite differently. Given a rectangle area with randomly deployed sensors, the scheme initially places pads to cover the whole deployed rectangle. Then, it removes



**FIGURE 7.** Regular deployment of wireless charging pads by dc.

redundant pads. Along the way, the DC scheme must not only cover the deployed rectangle, but also guarantees at least one available drone flight path connecting BS to every sensor node. That is, the DC scheme needs to ensure that the constructed flight graph  $G = (V, E)$  is connected, where  $V = \{BS = p_0, p_1, p_2, \dots, p_m\}$  and  $(p_i, p_j) \in E$  if and only if  $p_i \in V$  and  $p_j \in V$  and  $d(p_i, p_j) \leq d_{max}$ .

For example, in Fig. 7, to cover the rectangle, DC scheme locates the pads as indicated by the red points, where the blue circles show the flying range of the drone from pads. Notice that the rectangle is covered by the deployed pads and the constructed graph is also connected. Theorem 2 gives a formula for calculating the upper bound of the number of pads needed by the DC scheme within a given deployed rectangle.

*Theorem 2:* If the sides of a rectangle region are  $l$  and  $m$  respectively, the upper limit of the number of pads required is  $\lceil \frac{2l}{\sqrt{2} d_{max}} \rceil \lceil \frac{2m}{\sqrt{2} d_{max}} \rceil$ .

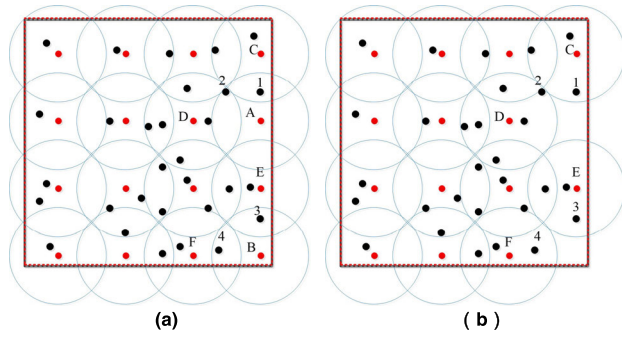
*Proof of Theorem 2:* As shown in Fig. 7, the region is covered regularly by several squares with side length  $\frac{\sqrt{2} d_{max}}{2}$ . To completely cover the rectangle region, the number of pads required is at most  $\lceil \frac{2l}{\sqrt{2} d_{max}} \rceil \lceil \frac{2m}{\sqrt{2} d_{max}} \rceil$ .

Since sensors are randomly deployed, there may be areas without any sensors. If there is a disk  $p$  such that after removing it every sensor can still be covered by at least one disk and the resulting flight graph is still connected, such disk  $p$  is redundant. In the DC scheme, such redundant deployed pads should be removed.

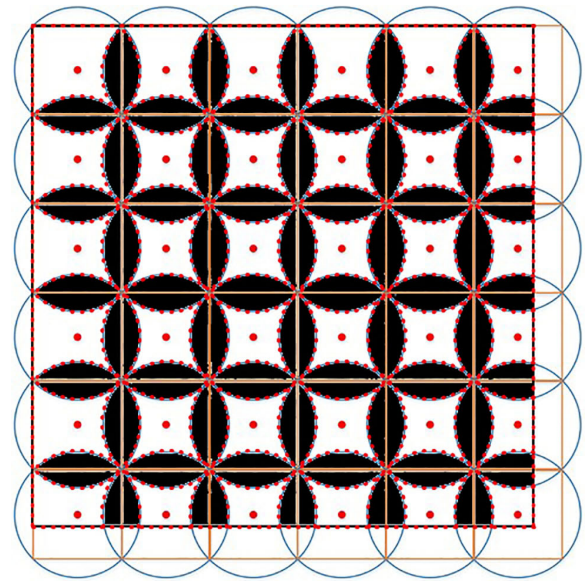
In Fig. 8, black points are sensors. Fig. 8(a) shows that sensor 1 is covered by disks A and C, sensor 2 by disks A and D, sensor 3 by disks B and E, and sensor 4 by disks B and F. Therefore, disks A and B are redundant which means pads A and B can be removed as shown in Fig. 8 (b).

**G. ANALYSIS OF DISK COVER SCHEME**

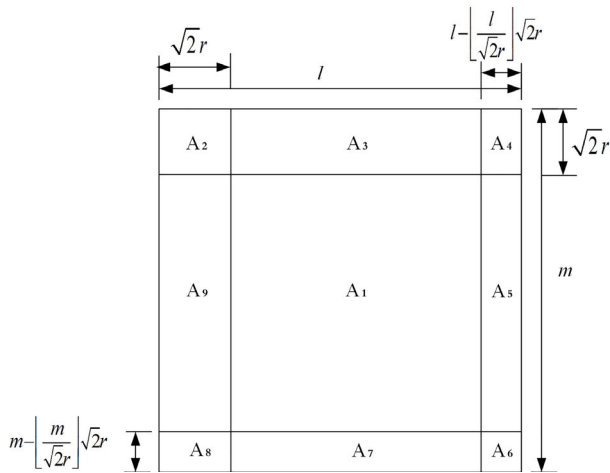
To estimate the number of pads needed, we present in this subsection the first paradigm to compute in a systematic way



**FIGURE 8.** An example of removing redundant deployed wireless charging pads: (a) A deployed area with wireless charging pads, (b) The same area after removing redundant wireless charging pads.



**FIGURE 10.** Regular deployment of wireless charging pads by dc.



**FIGURE 9.** Regions partitioning an  $l \times m$  rectangle  $R$ .

the probabilities of redundant pads which will be removed in the DC scheme for a given deployed rectangle. Based on the following theoretical results, we can formulate and calculate the expected number of wireless charging pads used in DC, which is the base line for the other proposed heuristic algorithms.

In the following definition, we formally define the concept of redundant pads.

**Definition 6:** Given a rectangle  $R$  with a set of sensors  $S = \{s_1, s_2, \dots, s_N\}$  and pads  $P = \{p_1, p_2, \dots, p_m\}$  deployed in  $R$ , a pad  $p$  is called *redundant* if  $P - \{p\}$  satisfies the two conditions in the following.

**Condition (1):** For every  $s_i$  in  $S$ , there exists at least a  $p_j$  in  $P - \{p\}$  such that  $d(s_i, p_j) \leq d_{max}/2$ .

**Condition (2):** The constructed graph  $G = (V, E)$  is connected, where  $V = \{BS\} \cup P - \{p\}$  and  $(p_i, p_j) \in E$  if and only if  $p_i \in V$  and  $p_j \in V$  and  $d(p_i, p_j) \leq d_{max}$ .

According to the border effect of coverage, we divide rectangle region  $R$  into nine sub-regions:  $A_1, A_2, \dots$ , and  $A_9$ , as depicted in Fig. 9.

Let  $E_i$  represent the event that a specific pad  $p$  in sub-region  $A_i$  is indispensable, for  $1 \leq i \leq 9$ . This means there is a sensor node located in the white area of a specific disk in

sub-regions  $A_i$  for  $1 \leq i \leq 9$  respectively (Fig. 10). Notice that every white area of a disk in the same sub-region has the same area size in the DC scheme. Also note that whenever there is an SN located in the white area of a specific disk in sub-region  $A_i$  in the DC scheme, pad  $p$  needs to be deployed in the white area; otherwise, there is no pad covering this SN.

In the following theorem and corollary,  $\mu_i$  denotes the number of pads being placed in  $A_i$  initially.

$$\begin{aligned} \mu_1 &= \left( \left\lceil \frac{2 \times l}{\sqrt{2} \times d_{max}} \right\rceil - 2 \right) \times \left( \left\lceil \frac{2 \times m}{\sqrt{2} \times d_{max}} \right\rceil - 2 \right), & \mu_2 &= 1, \\ \mu_3 &= \left( \left\lceil \frac{2 \times l}{\sqrt{2} \times d_{max}} \right\rceil - 2 \right), & \mu_4 &= 1 \\ \mu_5 &= \left( \left\lceil \frac{2 \times m}{\sqrt{2} \times d_{max}} \right\rceil - 2 \right), & \mu_6 &= 1, \\ \mu_7 &= \left( \left\lceil \frac{2 \times l}{\sqrt{2} \times d_{max}} \right\rceil - 2 \right), & \mu_8 &= 1, \\ \mu_9 &= \left( \left\lceil \frac{2 \times m}{\sqrt{2} \times d_{max}} \right\rceil - 2 \right). \end{aligned}$$

Here  $X$  denotes the number of redundant pads,  $r$  equals to  $d_{max}/2$ , and  $\phi_i$  is the size of the white area of a disk in sub-region  $A_i$  for  $1 \leq i \leq 9$ . Also  $\phi_{ij}$  denotes the size of the common covering area of two neighboring disks ( $i \neq j$ ) in sub-region  $A_i$  and  $A_j$  (shown in Fig. 13(a)), and  $Pr(E_i)$  denotes probability of an event that after tossing a sensor a specific pad  $p$  in sub-region  $A_i$  is not redundant, for  $1 \leq i \leq 9$ .

**Theorem 3:** Given an  $l \times m$  rectangle with a set of  $N$  uniformly deployed sensors  $S = \{s_1, s_2, \dots, s_N\}$  and pads  $P = \{p_1, p_2, \dots, p_m\}$  deployed using the DC scheme, we have

$$\begin{aligned} (1) \ Pr(X \geq 1) &= \sum_{j=1}^9 \mu_j \left( 1 - \frac{\phi_j}{l \times m} \right)^N \end{aligned} \tag{8}$$



$$\begin{aligned}
 (2) Pr(X \geq 2) &= \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, \alpha \neq \beta \\ d(p_\alpha, p_\beta) = \sqrt{2} r}} 1 - \frac{(\phi_i + \phi_j + \phi_{ij})}{lm} \right)^N \\
 &+ \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, \alpha \neq \beta \\ d(p_\alpha, p_\beta) > \sqrt{2} r}} 1 - \frac{(\phi_i + \phi_j)}{lm} \right)^N \quad (9)
 \end{aligned}$$

$$\begin{aligned}
 (3) Pr(X \geq 3) &= \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, p_\gamma \in A_k, \alpha \neq \beta \neq \gamma \\ d(p_\alpha, p_\beta) = \sqrt{2} r, d(p_\alpha, p_\gamma) = \sqrt{2} r}} 1 - \frac{(\phi_i + \phi_j + \phi_k + \phi_{ij} + \phi_{ik})}{lm} \right)^N \\
 &+ \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, p_\gamma \in A_k, \alpha \neq \beta \neq \gamma \\ d(p_\alpha, p_\beta) = \sqrt{2} r, d(p_\alpha, p_\gamma) > \sqrt{2} r \\ d(p_\beta, p_\gamma) > \sqrt{2} r}} 1 - \frac{(\phi_i + \phi_j + \phi_{ij}) + \phi_k}{lm} \right)^N \\
 &+ \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, p_\gamma \in A_k, \alpha \neq \beta \neq \gamma \\ d(p_\alpha, p_\beta) > \sqrt{2} r, d(p_\alpha, p_\gamma) > \sqrt{2} r \\ d(p_\beta, p_\gamma) > \sqrt{2} r}} 1 - \frac{\phi_i + \phi_j + \phi_k}{lm} \right)^N \\
 &- \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, p_\gamma \in A_k, \alpha \neq \beta \neq \gamma \\ p_\alpha, p_\beta, p_\gamma \text{ are one of four forbidden cases.}}} 1 - \frac{\phi_i + \phi_j + \phi_k}{lm} \right)^N \quad (10)
 \end{aligned}$$

*Proof of Theorem :*

By definition, we have

$$\begin{aligned}
 Pr(E_i) &= \frac{\phi_i}{l \times m}, \\
 Pr(\bar{E}_i) &= 1 - \frac{\phi_i}{l \times m}.
 \end{aligned}$$

After tossing  $N$  sensors in an  $l \times m$  rectangle, the probability of the occurrence of the number of redundant pad  $X$  is listed below (for  $X = 0, 1, 2, 3$ ).

We have

$$Pr(X \geq 1) = \left( \sum_{j=1}^9 \mu_j Pr(\bar{E}_j) \right)^N = \left( \sum_{j=1}^9 \mu_j \left( 1 - \frac{\phi_j}{l \times m} \right) \right)^N,$$

since the probability of the occurrence of at least one redundant pad is easy to derive by intentionally keeping the white area of a specific disk from being tossed in any sensor. Note that even a sensor randomly deployed in the area (excluding white areas) of a specific disk, the disk still can be removed because it is covered by another neighboring disk.

Similarly, we have

$$\begin{aligned}
 Pr(X \geq 2) &= \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, \alpha \neq \beta \\ d(p_\alpha, p_\beta) = \sqrt{2} r}} 1 - \frac{(\phi_i + \phi_j + \phi_{ij})}{lm} \right)^N \\
 &+ \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, \alpha \neq \beta \\ d(p_\alpha, p_\beta) > \sqrt{2} r}} 1 - \frac{(\phi_i + \phi_j)}{lm} \right)^N.
 \end{aligned}$$

By keeping the white areas of two specific disks from being tossed in any sensor, we would like to remove two disks at the same time. However, when the selected two disks are placed side by side (i.e., their distance is exactly  $\sqrt{2}r$ ) in DC scheme and there is a sensor placed in their intersection area, these two disks cannot be removed at the same time; otherwise, the sensor will not be covered by any disk. The above two disks are called a pair of *neighboring disks* hereafter. To remove a pair of neighboring disks, we need to keep both their white areas and their intersection area vacant.

For the case of removing three disks at the same time, we have

$$\begin{aligned}
 Pr(X \geq 3) &= \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, p_\gamma \in A_k, \alpha \neq \beta \neq \gamma \\ d(p_\alpha, p_\beta) = \sqrt{2} r, d(p_\alpha, p_\gamma) = \sqrt{2} r}} 1 - \frac{(\phi_i + \phi_j + \phi_k + \phi_{ij} + \phi_{ik})}{lm} \right)^N \\
 &+ \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, p_\gamma \in A_k, \alpha \neq \beta \neq \gamma \\ d(p_\alpha, p_\beta) = \sqrt{2} r, d(p_\alpha, p_\gamma) > \sqrt{2} r \\ d(p_\beta, p_\gamma) > \sqrt{2} r}} 1 - \frac{(\phi_i + \phi_j + \phi_{ij}) + \phi_k}{lm} \right)^N \\
 &+ \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, p_\gamma \in A_k, \alpha \neq \beta \neq \gamma \\ d(p_\alpha, p_\beta) > \sqrt{2} r, d(p_\alpha, p_\gamma) > \sqrt{2} r \\ d(p_\beta, p_\gamma) > \sqrt{2} r}} 1 - \frac{\phi_i + \phi_j + \phi_k}{lm} \right)^N \\
 &- \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, p_\gamma \in A_k, \alpha \neq \beta \neq \gamma \\ p_\alpha, p_\beta, p_\gamma \text{ are one of four forbidden cases.}}} 1 - \frac{\phi_i + \phi_j + \phi_k}{lm} \right)^N.
 \end{aligned}$$

When we consider three disks A, B, and C, there are the following three situations: (1) {A, B} and {B, C} are two pairs of neighboring disks, (2) only two of {A, B, C} form a pair of neighboring disks, (3) any pair of disks picked from {A, B, C} is not a pair of neighboring disks. In order to remove a pair of neighboring disks, we need to keep both the white areas of these two disks and their intersection area empty. According to its situation, we can obtain the corresponding probability. However, when we try to remove three disks at the same time, there exist four forbidden cases which are shown in Fig. 11. Note that although there is no sensor node in one of these three disks in the forbidden case, at least one of the three disks needs to remain to keep the network connected.

Since  $Pr(X = 0) = 1 - Pr(X \geq 1)$ ,  $Pr(X = 1) = Pr(X \geq 1) - Pr(X \geq 2)$ , and  $Pr(X = 2) = Pr(X \geq 2) - Pr(X \geq 3)$ , we have Corollary 4 immediately.

*Corollary 4:* Given an  $l \times m$  rectangle with a set of  $N$  uniformly deployed sensors  $S = \{s_1, s_2, \dots, s_N\}$  and pads  $P = \{p_1, p_2, \dots, p_m\}$  deployed using the DC scheme, we have

$$\begin{aligned}
 Pr(X = 0) &= 1 - \left( \sum_{j=1}^9 \mu_j \left( 1 - \frac{\phi_j}{l \times m} \right) \right)^N \quad (11)
 \end{aligned}$$



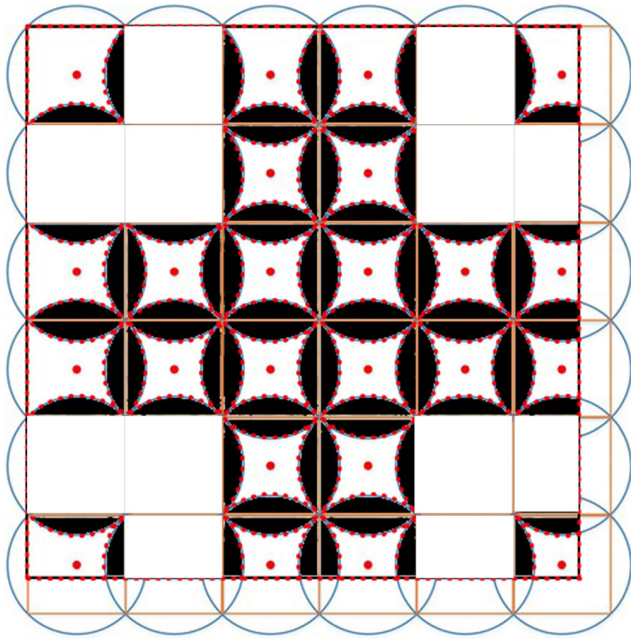


FIGURE 11. Four forbidden cases.

$$\begin{aligned}
 &Pr(X = 1) \\
 &= \left( \sum_{j=1}^9 \mu_j \left( 1 - \frac{\phi_j}{l \times m} \right) \right)^N \\
 &\quad - \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, \alpha \neq \beta \\ d(p_\alpha, p_\beta) = \sqrt{2}r}} 1 - \frac{(\phi_i + \phi_j + \phi_{ij})}{lm} \right)^N \\
 &\quad - \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, \alpha \neq \beta \\ d(p_\alpha, p_\beta) > \sqrt{2}r}} 1 - \frac{(\phi_i + \phi_j)}{lm} \right)^N. \tag{12} \\
 &Pr(X = 2) \\
 &= \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, \alpha \neq \beta \\ d(p_\alpha, p_\beta) = \sqrt{2}r}} 1 - \frac{(\phi_i + \phi_j + \phi_{ij})}{lm} \right)^N \\
 &\quad + \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, \alpha \neq \beta \\ d(p_\alpha, p_\beta) > \sqrt{2}r}} 1 - \frac{(\phi_i + \phi_j)}{lm} \right)^N \\
 &\quad - \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, p_\gamma \in A_k, \alpha \neq \beta \neq \gamma \\ d(p_\alpha, p_\beta) = \sqrt{2}r, d(p_\alpha, p_\gamma) = \sqrt{2}r}} 1 - \frac{(\phi_i + \phi_j + \phi_k + \phi_{ij} + \phi_{ik})}{lm} \right)^N \\
 &\quad - \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, p_\gamma \in A_k, \alpha \neq \beta \neq \gamma \\ d(p_\alpha, p_\beta) = \sqrt{2}r, d(p_\alpha, p_\gamma) > \sqrt{2}r \\ d(p_\beta, p_\gamma) > \sqrt{2}r}} 1 - \frac{(\phi_i + \phi_j + \phi_{ij}) + \phi_k}{lm} \right)^N \\
 &\quad - \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, p_\gamma \in A_k, \alpha \neq \beta \neq \gamma \\ d(p_\alpha, p_\beta) > \sqrt{2}r, d(p_\alpha, p_\gamma) > \sqrt{2}r \\ d(p_\beta, p_\gamma) > \sqrt{2}r}} 1 - \frac{\phi_i + \phi_j + \phi_k}{lm} \right)^N \\
 &\quad + \left( \sum_{\substack{p_\alpha \in A_i, p_\beta \in A_j, p_\gamma \in A_k, \alpha \neq \beta \neq \gamma \\ p_\alpha, p_\beta, p_\gamma \text{ are one of four forbidden cases.}}} 1 - \frac{\phi_i + \phi_j + \phi_k}{lm} \right)^N \tag{13}
 \end{aligned}$$

Here we shall see how the values of  $\phi_1, \phi_2, \dots, \phi_9$  and  $\phi_{ij}$  in Theorem 3 are calculated. The specific calculations for  $\phi_1, \phi_2, \dots, \phi_9$  (Fig. 12) are shown as follows.

The values of  $\phi_1, \phi_2,$  and  $\phi_3$  can be easily derived from Fig. 12(a)-12(d), that,

$$\begin{aligned}
 \phi_1 &= 4r^2 - \pi r^2 \\
 \phi_2 &= 3r^2 - \frac{\pi}{2}r^2 \\
 \phi_3 &= \phi_9 = 3\frac{1}{2}r^2 - \frac{3\pi}{4}r^2
 \end{aligned}$$

As shown in Fig. 12(e)-12(i), depending on the location of the region's sides and the maximum flight distance, three cases are possible.

Case (a): The center of the circle is inside the region.

Case (b): The center of circle is on the boundary of the region.

Case (c): The center of the circle is outside the region.

Let  $\phi_{4a}, \phi_{4b},$  and  $\phi_{4c}$  denote the white area of region  $A_4$  for cases a, b and c, respectively. We have,

$$\begin{aligned}
 \phi_{4a} &= \left( \theta + \frac{\beta}{2} \right) r^2 + \frac{1}{4}r^2 + \frac{\sqrt{2}}{2}vr + \frac{1}{2}vr \sin \alpha \\
 &\quad - \left( \alpha - \frac{1}{2}tg\alpha \right) r^2 - \left( v - \frac{\sqrt{2}}{2}rtg\beta \right)^2 tg\alpha - \left( \frac{3\pi}{4}r^2 - \frac{3}{2}r^2 \right) \tag{14}
 \end{aligned}$$

where  $\theta = \arcsin \frac{u}{r},$

$$u = \frac{\sqrt{\frac{1}{2}r^2 + (\sqrt{2}r + r)^2}}{2} = \frac{\sqrt{3.5 + 2\sqrt{2}}}{2}r,$$

$$v = \left( m - \left\lfloor \frac{m}{\sqrt{2}r} \right\rfloor \sqrt{2}r \right) - \frac{\sqrt{2}}{2}r,$$

$$\alpha = \arccos \frac{v}{r},$$

$$\beta = \frac{\pi}{2} - \alpha.$$

Obviously, we have

$$\phi_{4b} = \theta r^2 + \frac{1}{4}r^2 - \left( \frac{3\pi}{4}r^2 - \frac{3}{2}r^2 \right) \tag{15}$$

where  $\theta = \arcsin \frac{u}{r},$

$$u = \frac{\sqrt{\frac{1}{2}r^2 + (\sqrt{2}r + r)^2}}{2} = \frac{\sqrt{3.5 + 2\sqrt{2}}}{2}r.$$

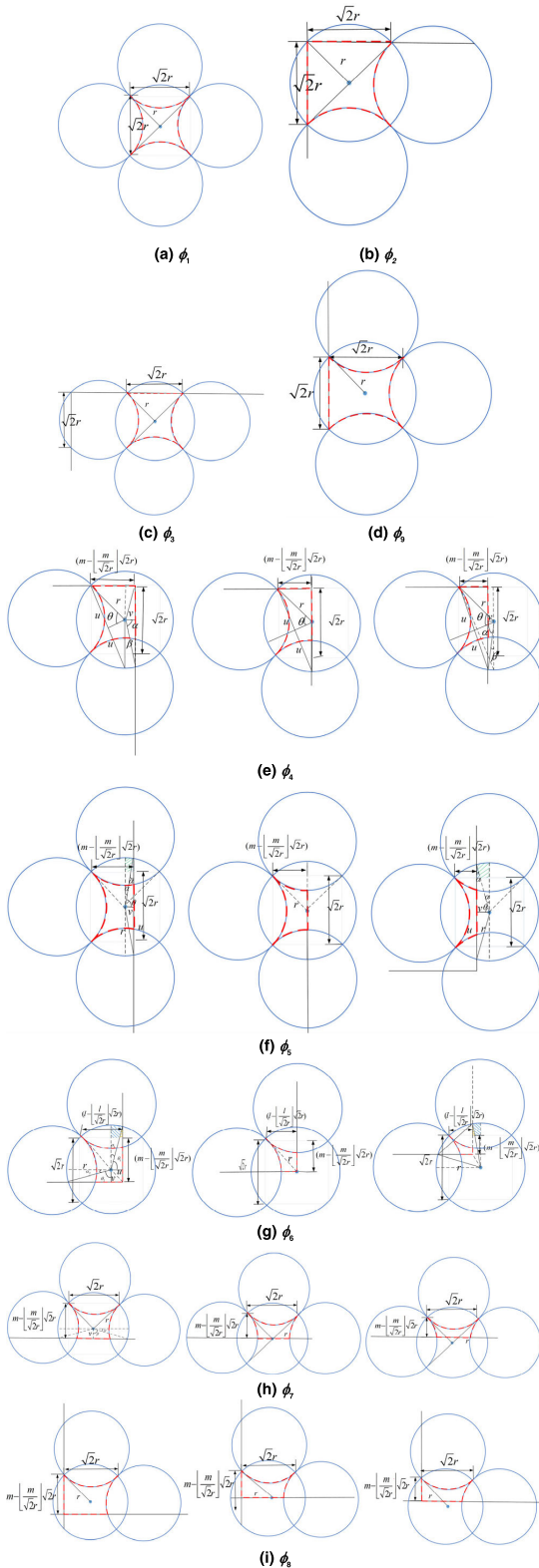
Similarly, we can obtain

$$\begin{aligned}
 \phi_{4c} &= \theta r^2 + \frac{1}{4}r^2 - \left( \frac{3\pi}{4}r^2 - \frac{3}{2}r^2 \right) - \frac{1}{2}vr \sin \alpha \\
 &\quad - \frac{\sqrt{2}}{2}vr - \frac{\beta}{2}r^2 + \left( \alpha - \frac{1}{2}tg\alpha \right) r^2 + \left( v - \frac{\sqrt{2}}{2}rtg\beta \right)^2 tg\alpha \tag{16}
 \end{aligned}$$

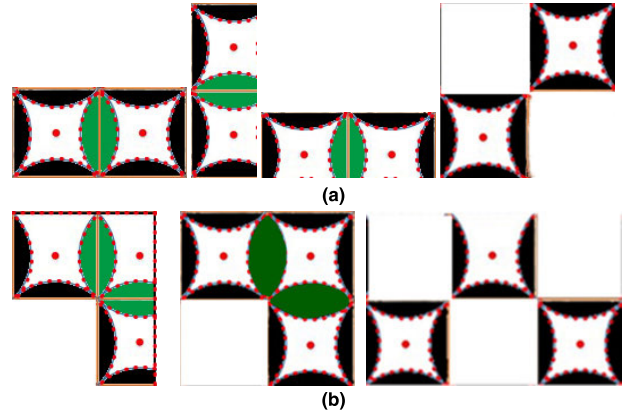
where  $\theta = \arcsin \frac{u}{r},$

$$u = \frac{\sqrt{\frac{1}{2}r^2 + (\sqrt{2}r + r)^2}}{2} = \frac{\sqrt{3.5 + 2\sqrt{2}}}{2}r,$$

$$v = \frac{\sqrt{2}}{2}r - \left( m - \left\lfloor \frac{m}{\sqrt{2}r} \right\rfloor \sqrt{2}r \right),$$



**FIGURE 12.** Wireless charging pads located in different regions: (a) a wireless charging pad in region  $A_1$ , (b) a wireless charging pad in region  $A_2$ , (c) a wireless charging pad in region  $A_3$ , (d) a wireless charging pad in region  $A_0$ , (e) three cases of a wireless charging pad in region  $A_5$ , (f) three cases of a wireless charging pad in region  $A_4$ , (g) three cases of a wireless charging pad in region  $A_6$ , (h) three cases of a wireless charging pad in region  $A_7$ , and (i) three cases of a pad in region  $A_8$ .



**FIGURE 13.** Cases for redundant wireless charging pads: (a) two redundant wireless charging pads (adjacent or no adjacent), (b) three redundant wireless charging pads (adjacent or no adjacent).

$$\alpha = \arccos \frac{v}{r}, \quad \beta = \frac{\pi}{2} - \alpha.$$

When  $v = 0$ , equation (14) is simplified to (15). Moreover, when  $v < 0$ , equation (14) is equivalent to (16). Therefore, we use (14) to compute  $\phi_4$  in all cases (i.e.,  $\phi_{4a}$ ,  $\phi_{4b}$ , and  $\phi_{4c}$ ).

Similarly, the computational process used to obtain  $\phi_5$  (Fig. 12(f)) is similar to  $\phi_4$ (Fig. 12(e)). Thus we have,

$$\begin{aligned} \phi_5 = & \frac{\pi}{2}r^2 + (\theta r^2 - vr \sin \theta) - [2(\alpha - \frac{1}{2} \tan \alpha)r^2 \\ & + 2(v - \frac{\sqrt{2}}{2}r \tan \alpha)^2 \tan \theta + (\pi r^2 - 2r^2)] \end{aligned} \quad (17)$$

where  $\theta = \arccos \frac{v}{r}$ ,  $\alpha = \frac{\pi}{2} - \theta$ ,

$$v = (m - \lfloor \frac{m}{\sqrt{2}r} \rfloor) \sqrt{2}r - \frac{\sqrt{2}}{2}r.$$

Specifically, when  $v = 0$ , equation (17) is reduced to case (b), and when  $v < 0$ , equation (17) is reduced to case c.

Again, the value of  $\phi_6$  (Fig. 12(g)) is obtained and listed as follows.

$$\begin{aligned} \phi_6 = & \frac{\pi}{4}r^2 + \frac{1}{2}u(r+v) + \frac{1}{2}v(r+u) + \frac{1}{2}(\alpha_1 + \alpha_2)r^2 \\ & - [(\frac{\pi}{2}r^2 - r^2) + (\alpha_1 - \frac{1}{2} \tan \alpha_1)r^2 \\ & + (v - \frac{\sqrt{2}}{2}r \tan \alpha_1)^2 \tan \theta_1 + (\alpha_2 - \frac{1}{2} \tan \alpha_2)r^2 \\ & + (v - \frac{\sqrt{2}}{2}r \tan \alpha_2)^2 \tan \theta_2] \end{aligned} \quad (18)$$

where

$$\begin{aligned} v = & (m - \lfloor \frac{m}{\sqrt{2}r} \rfloor) \sqrt{2}r - \frac{\sqrt{2}}{2}r, \quad u = (l - \lfloor \frac{l}{\sqrt{2}r} \rfloor) \sqrt{2}r, \\ \theta_1 = & \arccos \frac{u}{r}, \quad \theta_2 = \arccos \frac{v}{r}, \\ \alpha_1 = & \frac{\pi}{2} - \theta_1, \quad \alpha_2 = \frac{\pi}{2} - \theta_2. \end{aligned}$$

Similarly, when  $v = 0$ ,  $u = 0$ , equation (18) is reduced to case (b), and when  $v < 0$ ,  $u < 0$ , equation (18) is reduced to

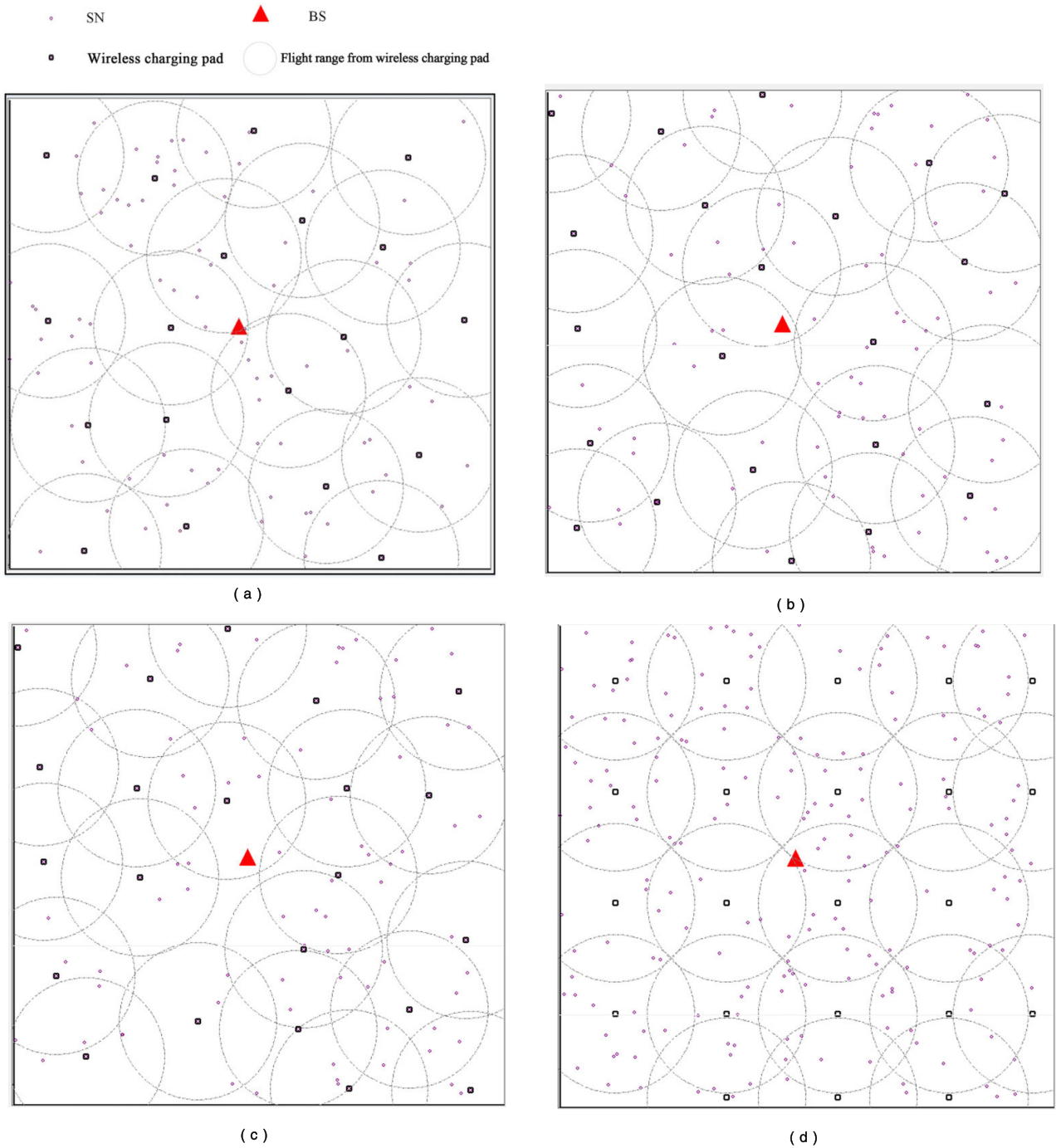


FIGURE 14. A snapshot for (a) MS, (b) TNC, (c) GNC, (d) dc scheme.

case (c). We also observe that the computational process for  $\phi_7$  (Fig. 12(h)) is similar to that for  $\phi_5$  (Fig. 12(f)), and that  $\phi_7$  is computed by using (17). Similarly, the computational process for  $\phi_8$  (Fig. 12(i)) is similar to that for  $\phi_4$  (Fig. 12(e)). Finally, we have

$$\begin{aligned} \phi_8 = & (\theta + \frac{\beta}{2})r^2 + \frac{1}{4}r^2 + \frac{\sqrt{2}}{2}vr + \frac{1}{2}vr \sin \alpha \\ & - (\alpha - \frac{1}{2}tg\alpha)r^2 - (v - \frac{\sqrt{2}}{2}rtg\beta)^2tg\alpha - (\frac{3\pi}{4}r^2 - \frac{3}{2}r^2) \end{aligned} \quad (19)$$

Recall that  $\phi_{ij}$  denotes the size of the common covered area of two neighboring disks ( $i \neq j$ ) in sub-region  $A_i$  and  $A_j$ . The specific calculations for  $\phi_{ij}$  (Fig. 13) are shown as follows.

Fig. 13(a) shows some different cases for existing two redundant pads. When  $i$  and  $j$  are a pair of neighboring disks, it can be observed that in the first case  $\phi_{ij} = \frac{\pi}{2}r^2 - r^2$ . In the second and third case, we have

$$\begin{aligned} \phi_{ij} = & 2(\alpha - \frac{1}{2} \tan \alpha)r^2 + 2(v - \frac{\sqrt{2}}{2}r \tan \alpha)^2 \tan \theta \\ & + (\pi r^2 - 2r^2) \end{aligned} \quad (20)$$

where  $\theta = \arccos \frac{v}{r}$ ,  $\alpha = \frac{\pi}{2} - \theta$ ,

$v = (m - \lfloor \frac{m}{\sqrt{2}r} \rfloor \sqrt{2}r) - \frac{\sqrt{2}}{2}r$ . In the fourth case,  $\phi_{ij} = 0$ .

On the other hand, Fig. 13(b) shows some different cases for three redundant pads. When  $(i, j)$  and  $(j, k)$  are two pairs of neighboring disks, it can be easily observed that in the first case, when  $\phi_{ij} = \frac{\pi}{2}r^2 - r^2$ :

$$\phi_{jk} = 2(\alpha - \frac{1}{2} \tan \alpha)r^2 + 2(v - \frac{\sqrt{2}}{2}r \tan \alpha)^2 \tan \theta + (\pi r^2 - 2r^2) \quad (21)$$

In the second case, we have  $\phi_{ij} = \phi_{jk} = \frac{\pi}{2}r^2 - r^2$ . Finally, in the fourth case,  $\phi_{ij} = \phi_{jk} = 0$ . Note that the total white area is  $\mu_1\phi_1 + \mu_2\phi_2 + \mu_3\phi_3 + \mu_4\phi_4 + \mu_5\phi_5 + \mu_6\phi_6 + \mu_7\phi_7 + \mu_8\phi_8 + \mu_9\phi_9$ .

## V. DRONE SCHEDULING SCHEME

In this section, we present a novel charging scheduling scheme in the proposed WRSN model. A simple recharging scheduling algorithm named *shortest multi-hop path* (SMHP) based on the proposed model is presented. In the SMHP scheme, the shortest path between any two pads and the neighboring pad of any sensor nodes are constructed and known to the base station. Therefore, two static routing tables are constructed in advance to store the shortest distance and shortest path between any two SNs (including BS) using Dijkstra's shortest path algorithm.

Then, the routing path between every two charging requests can be determined based on the two tables. Moreover, we propose three scheduling schemes, and two of which sort the charging requests in advance by EDF or NJNP. A routing path is inserted between every two charging requests obtained by SMHP between two SNs. The third scheme, called *shortest-flight first* (SFF), is similar to NJNP, but it sorts the charging requests according to the shortest flight distance instead of the Euclidean distance between two SNs. The SN with the shortest flight distance should be charged first in SFF scheme.

The routing table construction and the SMHP scheme are presented in Algorithms 4 and 5, respectively.

Notice that Algorithm 4 constructs the routing table  $D(i, j)$  containing only the routes (edges in graph terms) between two pads or one pad and one sensor, not between sensors. This is because that according to Condition (1) and Condition (2) of Definition 4 and Theorem 1, for each SN, there always exists an available drone flight path between the BS and this SN in the routing table  $D(i, j)$ . Actually, every flight path in  $D(i, j)$  is available. Thus Algorithm 4 can find an available flight path by simply applying shortest path algorithm.

For Algorithm 4, Step 1 uses  $O(n^2)$  time for constructing an initial distance matrix. Step 2 takes  $O(n^3)$  time to update distance matrix. Totally, Algorithm 4 takes  $O(n^3)$  time.

For Algorithm 5, both Step 1 and Step 2 takes  $O(1)$  time. Step 3 takes  $O(n)$  time to update the charging path. Next, Step 4 takes  $O(1)$  time to find a shortest path for returning to BS. Totally, Algorithm 5 takes  $O(n)$  time.

## Algorithm 4 Routing Table Construction

Input: A node set  $V$ ,  $N = |V|$ , a distance cost matrix  $C$  of nodes,  $d_{max}$ .

Output: A routing table *Router* and a flight distance matrix  $D$  between nodes.

```

Step 1: for  $i \leftarrow 1$  to  $N$ 
        for  $j \leftarrow 1$  to  $N$ 
            if  $s_i$  is pad and  $s_j$  is pad
                if  $C(i, j) \leq d_{max}$ 
                     $D(i, j) \leftarrow C(i, j)$ .
                else
                     $D(i, j) \leftarrow \infty$ .
            else if  $s_i$  is pad or  $s_j$  is pad
                if  $C(i, j) \leq d_{max}/2$ 
                     $D(i, j) \leftarrow C(i, j)$ .
                else
                     $D(i, j) \leftarrow \infty$ .
            else
                 $D(i, j) \leftarrow \infty$ .

```

Step 2: Compute all pair shortest paths of matrix  $D$  by calling Dijkstra's shortest path algorithm.

Step 3: Output *Router* and  $D$ .

## VI. SIMULATION RESULTS

In this section, extensive simulations are conducted to evaluate the effectiveness and efficiency of the proposed algorithms described in Sections IV and V. We present the first simulation setup, followed by a comparison and discussion of simulation results. The number of pads is an important performance metric in this work. We examine the number of pads used by the four proposed algorithms via various network parameters such as  $d_{max}$ , network density and the size of the sensing area.

After deploying pads, the constructed flying networks are combined with the aforementioned three scheduling algorithms EDF, NJNP, and SFF. Based on SMHP, the merits of the new network model are validated. The second simulation setup is presented, followed with comparison and discussion of simulation results by comparing the performance using metrics such as the lifetime of WRSN, number of successfully-charged SNs, average flight distance and total flight distance.

The simulations were performed on a PC with a quad-core 3.2 GHZ Intel i5 processor and 16 GB RAM, and the algorithms were implemented in Visual Studio C# 2017.

### A. PERFORMANCE OF THE PROPOSED ALGORITHMS

Static rechargeable SNs are deployed randomly over an  $l \times l$  square region. The BS is located in the center of the region. We take the average of 100 different outcomes for every value in the following figures according to the same network parameters.

In this subsection, we compare the MSC, TNC, GNC and DC scheme by analyzing the impact of various network



**Algorithm 5** SMHP Algorithm

Input: A sorted charging requests  $r$  created by applying one of three schemes (i.e. EDF, NJNP or SFF), the number of nodes  $N$ , a BS  $v_0$ , a drone, a path  $\Pi = \emptyset$ , routing table *Router*, a distance matrix  $D$ .

Output: A charging schedule path  $\Pi$ .

Step 1: Set BS as the first position of the schedule; that is,  $\Pi = v_0$ .

Step 2:  $v \leftarrow v_0$ .

Step 3: for every  $r(i)$  in the sorted charging requests,  $u \leftarrow r(i)$ .

Look up table *Router* to find a shortest path  $C$  from  $v$  to  $u$ .

$\Pi \leftarrow \Pi \cup C$ .

$\Pi \leftarrow \Pi \cup \{u\}$ .

$v \leftarrow u$ .

Step 4:  $u \leftarrow v_0$ .

Search table *Router* to find a shortest path  $C$  from  $v$  to  $u$ .

$\Pi \leftarrow \Pi \cup C$ .

$\Pi \leftarrow \Pi \cup \{u\}$ .

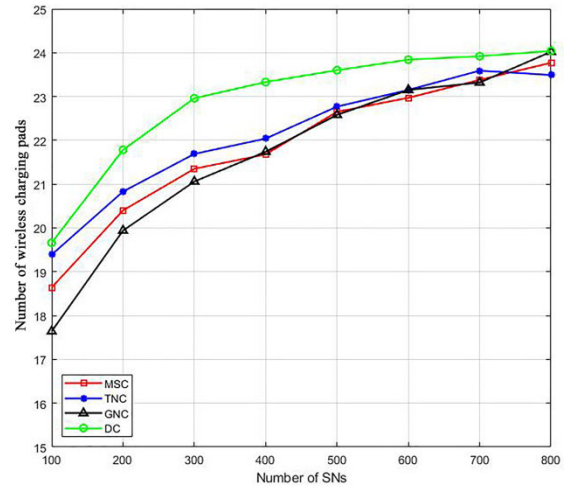
Step 5: Output the obtained charging path  $\Pi$ .

parameters on the number of required pads. First, snapshot for MSC, TNC, GNC and DC are depicted in Fig. 14 (a)-(d), respectively. In these figures, black and violet small circles represent pads and SNs, respectively. The red triangle is the BS. The large, grey dotted-line circles depicts the flight range of the drone from pads. We set the deployed region size as  $6000\text{m} \times 6000\text{m}$ , the number of SNs as  $N = 100$ , and the maximum flight distance as  $d_{max} = 2000\text{m}$ . We can see that the positions and the numbers of deployed pads obtained by MSC, TNC, GNC and DC are different in Fig. 14 (a)-(d).

To analyze the impact of network density on the number of deployed pads, the number of SNs varies between 100 and 800 in the simulations. In Fig. 15, we can see that when the number of SNs increases, the number of deployed pads also increases for all four proposed schemes.

However, when the number of SNs reaches a certain large value, the number of pads becomes nearly constant. When the network density becomes higher, more SNs are deployed, and thus more pads are required. When the number of SNs is large enough, every point of the deployed region is covered by a least one pad; therefore, the number of deployed pads reaches the upper limit proposed in Theorem 2.

When the number of SNs is less than 400, as shown in Fig. 15, the relationship between the four schemes is  $GNC < MSC < TNC < DC$ . That is, when the density is lower, the number of pads obtained by MSC, TNC and GNC is lower than that obtained by DC in Fig. 15. Since MSC, TNC and GNC always greedily choose the next position covering the maximal SNs when deploying pad, pads are mainly deployed in areas with dense SNs. Different from the three schemes, DC deploys pads evenly in the simulation region. If some



**FIGURE 15.** Comparison of number of wireless charging pads for diverse network density.

pad covers at least one SN that is not yet covered by other pads, the pad remains. Otherwise, it is removed. Therefore, the number of pads obtained by DC is higher than that obtained by MSC, TNC and GNC in lower densities.

In addition, the number of pads obtained by GNC is slightly smaller than MSC and TNC in lower density, because GNC uses the flight range to choose the next optimal position which has bigger area than that from MSC and TNC which uses two hops.

On the other hand, as shown in Fig. 15, when the number of SNs is between 500 and 700, the relationship between the four schemes is  $GNC \approx MSC < TNC < DC$ . Because the choosing range of GNC and MSC is close in high density.

When the number of SNs is 800, MSC, TNC and GNC is close to DC. That is, in high density, the differences among the four schemes become very small.

The impact of the region size is analyzed by varying the length of region sides from  $l = 1000\text{m}$  to  $6000\text{m}$ . For this purpose, the number of SNs is fixed as  $N = 100$ , and the flight distance is fixed at  $d_{max} = 2000\text{m}$ . The number of pads obtained by the four deployment schemes for different region sizes is depicted in Fig. 16. In Fig. 16, we can easily see that the number of pads increases in tandem with increases of region size. A larger regions requires more pads, so this result makes sense.

The numbers of pads obtained by MSC, TNC and GNC are almost the same. However, numbers obtained by these three schemes are lower than the one obtained from DC when the region is large, and higher when the region is small.

Note that when the region size is  $1000\text{m} \times 1000\text{m}$ , the numbers of pads obtained by the four schemes are the same, and the maximum degree in the constructed graph is 99. Since when  $N = 100$ , only one pad is deployed to cover the entire area of the region.

As shown in Fig. 16, when the region size is between  $1000\text{m} \times 1000\text{m}$  and  $4000\text{m} \times 4000\text{m}$ , the number of pads

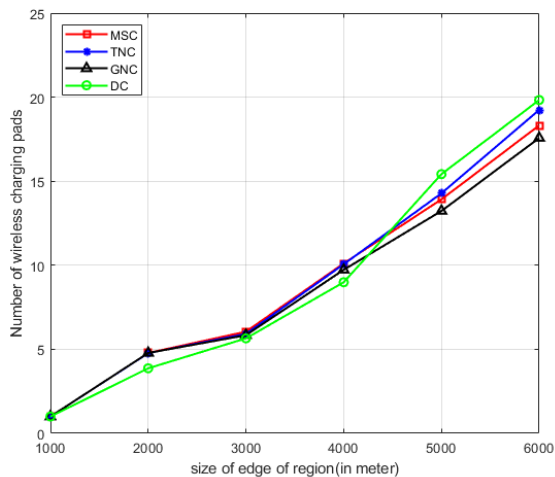


FIGURE 16. Number of required wireless charging pads when region size changes.

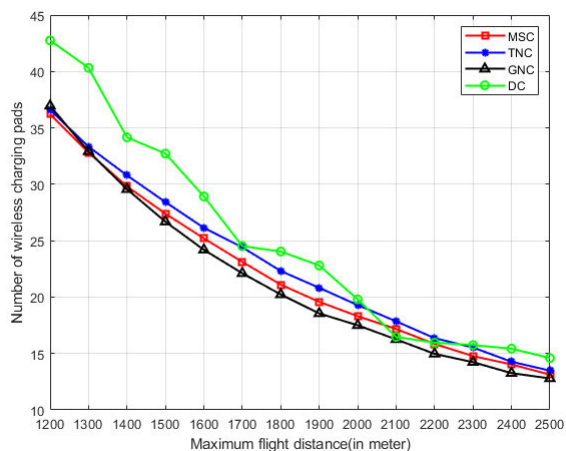


FIGURE 17. Number of deployed wireless charging pads when maximum flight distance varies.

obtained by DC is a little less than MSC, TNC and GNC. Because in small region, the pads obtained by DC are located in center of region. However, MSC, TNC and GNC are located in high density, which results in requirement for more pads to cover some SNs located near boundary.

On the other hand, when the region size is between 5000m × 5000m and 6000m × 6000m, the number of pads obtained by DC is a little higher than that of MSC, TNC and GNC. Therefore, in large regions, a dense distribution of pads is more efficient.

In Fig. 17, we consider the number of pads derived by MSC, TNC, GNC and DC for different values of  $d_{max}$ . For this simulation, we keep the region size at 6000m × 6000m, and the number of SNs at  $N = 100$ . We compute the number of pads with varying  $d_{max}$  between a range of 1200 m to 2500 m. In Fig. 17, the number of pads decreases as the maximum flight distance increases because when the maximum flight distance is longer, the area covered by one pad is bigger,

TABLE 2. Simulation parameters of second simulation.

| Parameters                     | Values    |
|--------------------------------|-----------|
| Region size (m <sup>2</sup> )  | 6000×6000 |
| Number of SNs                  | 200       |
| $d_{max}$ (m)                  | 2000      |
| Speed of drone (m/s)           | 10~50     |
| Energy threshold of SNs (s)    | 200~1300  |
| Energy consumption rate (KJ/s) | 0.002     |
| Initial energy (KJ)            | 10        |

and thereby the number of pads that the same region requires is smaller.

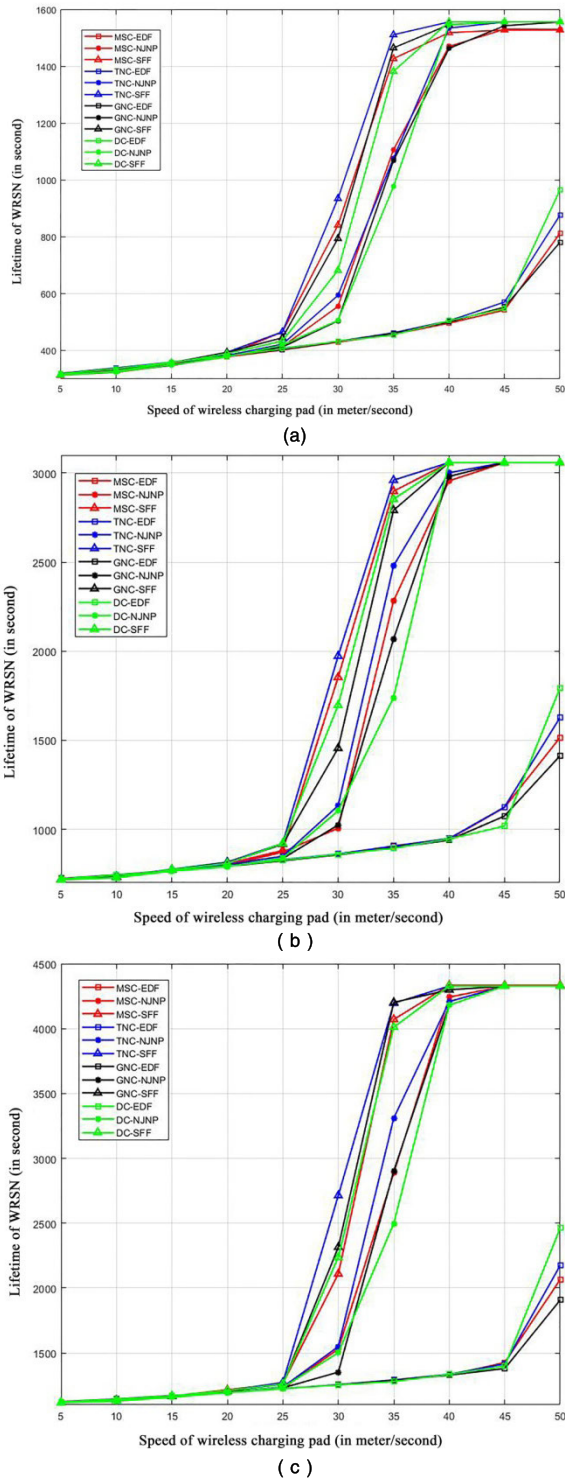
The numbers of pads obtained by MSC, TNC and GNC are generally lower than that of DC when  $d_{max}$  ranges between 1200 and 2000. However, when  $d_{max}$  is slightly higher at 2100 or 2200, the number of pads obtained by DC is less than that of MSC and TNC due to the boundary effect. Because the area covered by the pads near the boundary is very small, having more pads has high probability to be redundant, and to be removed by DC. In addition, the number of pads obtained by GNC is generally lower than that of MSC, TNC and DC for the same reason discussed before.

Note that the number of pads obtained by DC is a little bit zigzagging, when graphed, due to the boundary effect, an effect that rapidly declines when  $d_{max}$  is equal to 1700 and 2100. When  $d_{max}$  is equal to 1600, according to Theorem 2, the upper limit of the number of pads is 36, but on the boundary, we have removed an average of seven pads due to redundancy. And when  $d_{max}$  is equal to 1700, by Theorem 2, the upper limit of the number of pads becomes 25, and almost no pad is removed. Similarly, when  $d_{max}$  is equal to 2100, the upper limit of number of pads required is 25, but the area covered by pads located within the boundary effect region is very small, and most of the pads located in that region are redundant. Thus, the number of pads declines rapidly in some cases.

### B. SMHP ALGORITHM COMBINED WITH THE PROPOSED PAD DEPLOYMENT SCHEMES

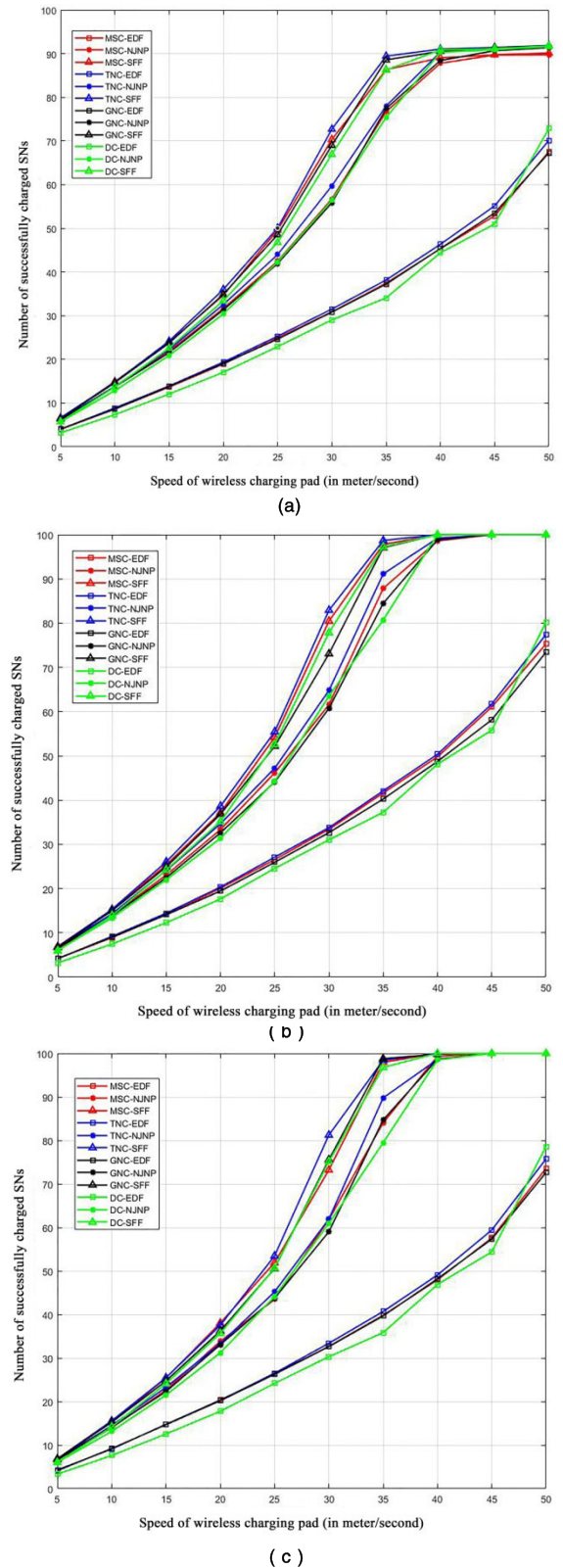
In this subsection, we consider a specific scenario, in which there are 200 static rechargeable SNs over a 6000m × 6000m square region. The location of the BS is in the center with coordinates (3000, 3000). Other simulation parameters are shown in Table 2.

In this subsection, we conduct simulations to compare the SMHP scheduling algorithm based on earliest-deadline first (EDF) [16], nearest-job-next preemption (NJNP) [17], [24] and shortest-flight first (SFF) scheduling algorithms in order to analyze the impact of various network parameters such as lifetime of network, number of successfully charged SNs and average flight distance. For easy reference, we call each simulation with the pad deployment method name followed by the scheduling method name. For example, the scheduling according to EDF based on the deployment of MSC is called MSC\_EDF.



**FIGURE 18.** Comparison of lifetime under various speeds of wireless charging drone with different workloads as follows, (a) heavy network workload, (b) medium network workload, (c) light network workload.

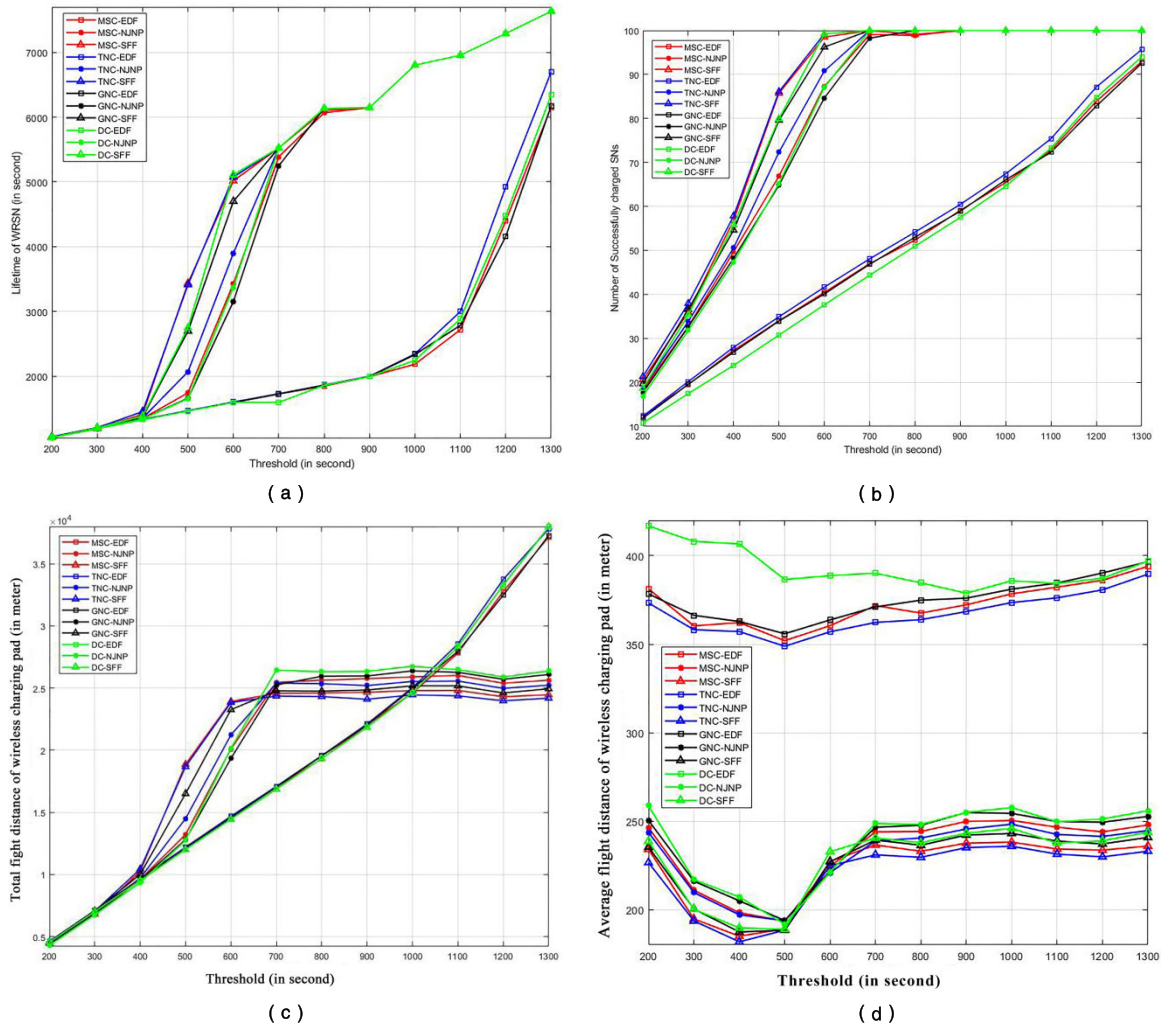
In this work, *lifetime* is defined as the difference between the time when system starts and the time when the first SN dies in a given WRSN. When the first dead SN appears, the simulation will stop. Lifetime is a very important metric for evaluating the performance of the proposed scheduling algorithm. The number of successfully charged SNs refers



**FIGURE 19.** Comparison of the number of successful charged SNs under various speeds of wireless charging drone in different workloads as follows, (a) heavy network workload, (b) medium network workload, (c) light network workload.

to the number of SNs that send charging requests and are charged by the drone during the lifetime of a given WRSN.





**FIGURE 20.** Comparison of performance under various energy thresholds of SNs, (a) lifetime of WRSN, (b) number of successful charged SNs, (c) total flight distance, (d) average flight distance.

The average flight distance of drone is defined as the average flight distance of drone for completing a charging request, which is the distance a drone flies from the position of current charging SN to the position of the next SN in the charging schedule.

Next, we check the performance of the proposed algorithms by changing the speed of drone from 10 to 50 m/s, while the maximum flight distance is kept at  $d_{max} = 2000m$  and the energy threshold is kept at 300s. Fig. 18 and 19 show the corresponding simulation results.

Note that when the speed of the drone increases, both the lifetime of WRSN and the number of successfully charged SNs increase. This is because when the speed of drone is high, its flight time is short, allowing more SNs with charging requests to be served in time. While the lifetime and the number of successfully charged SNs of SFF is the highest among the three methods, these two metrics under EDF is the lowest. That is because a drone takes the least time in flight using SFF method and the most time using EDF method.

Apart from this, we can observe that the lifetime and the number of successfully charged SNs based on the TNC are the highest among the four methods, and those two metrics based on DC are the lowest. This is because the deployment of pads obtained by the DC scheme is more uniform and DC focuses less on leveraging density, which results in longer flight paths between pads. On the other hand, pads deployed by TNC scheme are closer to each other; in other words, the distance between pads is shorter. Therefore, the drone requires less flight time when the charging schedule is based on TNC.

We also run simulations for different energy thresholds of SNs. In these simulations, we keep the speed of drone as 20m/s, the maximum flight distance as 2000m. We vary the energy threshold from 200s to 1300s. For simulation purposes, energy of sensors is replaced by time (in seconds). Fig. 20(a)-(d) shows the performances obtained by twelve methods under different energy thresholds. From Fig. 20(a)-(d), we can observe that SFF slightly outperforms NJNP, and NJNP outperforms EDF because SFF saves more time in flight than other schemes. In Fig. 20(c), the total flight



distances of SFF and NJNP is higher than that of EDF in the range of 400s to 900s.

This is because the number of successfully charged SNs obtained by MSC\_NJNP and TNC\_NJNP is much more than that of MSC\_EDF and TNC\_EDF, as shown in Fig. 20(b), i.e. the total flight distance increases with the increase in number of successfully charged SNs.

## VII. CONCLUSION AND FUTURE WORK

In this work, we propose a novel WRSN model that charges low-power SNs using a drone with the help of pads. To overcome the shortcoming of limited flight distance of the drone, we formulate a problem to address the pad deployment issue. Algorithms MSC, TNC and GNC are designed to maximize efficiency of pad placement and take into account replenishment energy, time, flight distance and geometric distribution of nodes. In addition, to demonstrate the merits of the proposed algorithms, DC, a simple static deployment scheme of pads is also proposed. Simulation results show that the number of pads obtained by the three proposed algorithms is generally lower than that of DC. Then, based on the results obtained by MSC, TNC and GNC algorithms, we propose a SMHP charging scheduling scheme by considering the maximum flight distance. The proposed algorithm has been extensively simulated combined with charging schedules created using EDF, NJNP and SFF respectively. Simulation results show that SFF outperforms NJNP and EDF.

Although the proposed algorithms are primarily designed to address the pad deployment problem in the newly proposed WRSN model, more effective and optimal schemes for the pad deployment problem need to be designed as well. In the future, more attention will be paid to the characteristics of deployed pads. In addition, scenarios using one wirelessly-charged vehicles and multiple drones will also be taken into account to extrapolate the robustness of the proposed model under varying configurations and circumstances.

## REFERENCES

- [1] L. M. Borges, F. J. Velez, and A. S. Lebres, "Survey on the characterization and classification of wireless sensor network applications," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1860–1890, Apr. 2014.
- [2] M. Zhao, J. Li, and Y. Yang, "A framework of joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 12, pp. 2689–2705, Dec. 2014.
- [3] L. Xie, Y. Shi, Y. Thomas Hou, W. Lou, H. D. Sherali, and S. F. Midkiff, "Multi-node wireless energy charging in sensor networks," *IEEE/ACM Trans. Netw.*, vol. 23, no. 2, pp. 437–450, Apr. 2015.
- [4] L. Fu, L. He, P. Cheng, Y. Gu, J. Pan, and J. Chen, "ESync: Energy synchronized mobile charging in rechargeable wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7415–7431, Sep. 2016.
- [5] 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.
- [6] H. Chae, J. Park, H. Song, Y. Kim, and H. Jeong, "The IoT based automate landing system of a drone for the round-the-clock surveillance solution," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatronics (AIM)*, Busan, South Korea, Jul. 2015, pp. 1575–1580.
- [7] M. Simic, C. Bil, and V. Vojisavljevic, "Investigation in wireless power transmission for UAV charging," *Procedia Comput. Sci.*, vol. 60, pp. 1846–1855, Jan. 2015.
- [8] C. H. Choi, H. J. Jang, S. G. Lim, H. C. Lim, S. H. Cho, and I. Gaponov, "Automatic wireless drone charging station creating essential environment for continuous drone operation," in *Proc. Int. Conf. Control, Autom. Inf. Sci. (ICCAIS)*, Ansan, South Korea, Oct. 2016, pp. 132–136.
- [9] C. Wang and Z. Ma, "Design of wireless power transfer device for UAV," in *Proc. IEEE Int. Conf. Mechatronics Autom.*, Harbin, China, Aug. 2016, pp. 2449–2454.
- [10] T. Campi, F. Dionisi, S. Cruciani, V. De Santis, M. Feliziani, and F. Maradei, "Magnetic field levels in drones equipped with wireless power transfer technology," in *Proc. Asia-Pacific Int. Symp. Electromagn. Compat. (APEMC)*, Shenzhen, China, May 2016, pp. 544–547.
- [11] T. Campi, S. Cruciani, M. Feliziani, and F. Maradei, "High efficiency and lightweight wireless charging system for drone batteries," in *Proc. AEIT Int. Annu. Conf.*, Cagliari, Italy, Sep. 2017, pp. 1–6.
- [12] S. Aldaher, P. D. Mitcheson, J. M. Artega, G. Kkelis, and D. C. Yates, "Light-weight wireless power transfer for mid-air charging of drones," in *Proc. 11th Eur. Conf. Antennas Propag. (EUCAP)*, Paris, France, Mar. 2017, pp. 336–340.
- [13] T. M. Mostafa, A. Muharam, and R. Hattori, "Wireless battery charging system for drones via capacitive power transfer," in *Proc. IEEE PELS Workshop Emerg. Technol., Wireless Power Transf. (WoW)*, Chongqing, China, May 2017, pp. 1–6.
- [14] T. Long, M. Ozger, O. Cetinkaya, and O. B. Akan, "Energy neutral Internet of drones," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 22–28, Jan. 2018.
- [15] S.-C. Choi, I.-Y. Ahn, J.-H. Park, and J. Kim, "Towards real-time data delivery in oneM2M platform for UAV management system," in *Proc. Int. Conf. Electron., Inf., Commun. (ICEIC)*, Auckland, New Zealand, Jan. 2019, pp. 1–3.
- [16] J. A. Stankovic, *Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms*, vol. 460. Berlin, Germany: Springer, 2012.
- [17] L. He, Y. Gu, J. Pan, and T. Zhu, "On-demand charging in wireless sensor networks: Theories and applications," in *Proc. IEEE 10th Int. Conf. Mobile Ad-Hoc Sensor Syst.*, Hangzhou, China, Oct. 2013, pp. 28–36.
- [18] L. Fu, P. Cheng, Y. Gu, J. Chen, and T. He, "Optimal charging in wireless rechargeable sensor networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 1, pp. 278–291, Jan. 2016.
- [19] S. Guo, C. Wang, and Y. Yang, "Joint mobile data gathering and energy provisioning in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 12, pp. 2836–2852, Dec. 2014.
- [20] Y. Shu, H. Yousefi, P. Cheng, J. Chen, Y. J. Gu, T. He, and K. G. Shin, "Near-optimal velocity control for mobile charging in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 7, pp. 1699–1713, Jul. 2016.
- [21] H. Liu, Q. Deng, S. Tian, X. Peng, and T. Pei, "Recharging schedule for mitigating data loss in wireless rechargeable sensor network," *Sensors*, vol. 18, no. 7, p. 2223, Jul. 2018.
- [22] X. Liu, A. Liu, T. Wang, K. Ota, M. Dong, Y. Liu, and Z. Cai, "Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks," *J. Parallel Distrib. Comput.*, vol. 135, pp. 140–155, Jan. 2020.
- [23] X. Liu, T. Wang, W. Jia, A. Liu, and K. Chi, "Quick convex hull-based rendezvous planning for delay-harsh mobile data gathering in disjoint sensor networks," *IEEE Trans. Syst., Man, Cybern. Syst.*, to be published, doi: 10.1109/TSMC.2019.2938790.
- [24] L. He, "Evaluating the on-demand mobile charging in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 9, pp. 1861–1875, Sep. 2014.
- [25] C. Lin, B. Xue, Z. Wang, D. Han, J. Deng, and G. Wu, "DWDP: A double warning thresholds with double preemptive scheduling scheme for wireless rechargeable sensor networks," in *Proc. IEEE 17th Int. Conf. High Perform. Comput. Commun.*, New York, NY, USA, Aug. 2015, pp. 503–508.
- [26] C. Lin, Z. Wang, D. Han, Y. Wu, C. W. Yu, and G. Wu, "TADP: Enabling temporal and distastical priority scheduling for on-demand charging architecture in wireless rechargeable sensor networks," *J. Syst. Archit.*, vol. 70, pp. 26–38, Oct. 2016.
- [27] C. Lin, D. Han, J. Deng, and G. Wu, "P<sup>2</sup>S: A primary and passer-by scheduling algorithm for on-demand charging architecture in wireless rechargeable sensor networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 8047–8058, Sep. 2017.
- [28] C. Lin, Y. Zhou, H. Song, C. W. Yu, and G. Wu, "OPPC: An optimal path planning charging scheme based on schedulability evaluation for WRSNs," *ACM Trans. Embedded Comput. Syst.*, vol. 17, no. 1, pp. 1–25, Oct. 2017.

[29] 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, pp. 123–134, Jul. 2018.

[30] Y. Wang, Y. Dong, S. Li, H. Wu, and M. Cui, "CRCM: A new combined data gathering and energy charging model for WRSN," *Symmetry*, vol. 10, no. 8, p. 319, Aug. 2018.

[31] C. Xu, R.-H. Cheng, and T.-K. Wu, "Wireless rechargeable sensor networks with separable charger array," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 4, Apr. 2018, Art. no. 155014771876899.

[32] D. Zorbas and C. Douligeris, "Computing optimal drone positions to wirelessly recharge IoT devices," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Honolulu, HI, USA, Apr. 2018, pp. 628–633.

[33] A. Bürkle, F. Segor, and M. Kollmann, "Towards autonomous micro UAV swarms," *J. Intell. Robotic Syst.*, vol. 61, nos. 1–4, pp. 339–353, Jan. 2011.

[34] D. Lichtenstein, "Planar formulae and their uses," *SIAM J. Comput.*, vol. 11, no. 2, pp. 329–343, May 1982.

[35] M. Charikar, S. Guha, É. Tardos, and D. B. Shmoys, "A constant-factor approximation algorithm for the  $k$ -median problem," *J. Comput. Syst. Sci.*, vol. 65, no. 1, pp. 129–149, Aug. 2002.

[36] D. S. Hochbaum and D. B. Shmoys, "A best possible heuristic for the  $k$ -center problem," *Math. Oper. Res.*, vol. 10, no. 2, pp. 180–184, May 1985.



**JINGJING CHEN** was born in Longyan, China, in 1982. She received the B.S. and M.S. degrees in electronic and information engineering from Central China Normal University, in 2005 and 2007, respectively. She is currently pursuing the Ph.D. degree in engineering science with Chung Hua University, Hsinchu, Taiwan. Since 2010, she has been a Lecturer with Longyan University. Her current research interests include wireless sensor networks and wireless communication.



**CHANG WU YU** was born in Taoyuan, Taiwan, in 1964. He received the B.S. degree from Soochow University, in 1985, the M.S. degree from National Tsing Hua University, in 1989, and the Ph.D. degree from National Taiwan University, in 1993, all in computer science. He is currently a Professor with the Department of Computer Science and Information Engineering, Chung Hua University, Taiwan. He has published more than 80 articles, including the IEEE JOURNAL

ON SELECTED AREAS IN COMMUNICATIONS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, *Theoretical Computer Science*, *Wireless Networks (ACM/Springer)*, *ACM Transactions on Embedded Systems*, and *Ad Hoc Networks (Elsevier)*. His current research interests include fundamental theories, algorithms, and applications in wireless networks, wireless sensor networks, and distributed computing by exploiting graph techniques. He received best paper awards from the 2008 ACM International Conference on Sensor, Ad Hoc, and Mesh Networks, the Ninth Workshop on Wireless, Ad Hoc, and Sensor Networks (WASN 2013), and the 2004 and 2007 Mobile Computing Workshop. He serves on the editor boards of several international journals, including *Ad Hoc & Sensor Wireless Networks: An International Journal*. Moreover, he serves as a guest editor for several international journals.



**WEN OUYANG** was born in Hsinchu, Taiwan. She received the Ph.D. degree in computer science from The University of Texas at Dallas, in 1993. From then on, she worked as a Member of Scientific Staff and a Senior Member of Scientific Staff with Bell Northern Research, later became Nortel Networks, where she was involved in large-scale telecommunication systems research and development. She is currently an Associate Professor with the Department of Computer Science and Information Engineering, Chung Hua University, Taiwan. Her research interests

include fault tolerance, routing schemes, tracking, and energy shortage for wireless sensor networks. She also works in topics involving software verification and validation techniques.

...