Bus Network Assisted Drone Scheduling for Sustainable Charging of Wireless Rechargeable Sensor Network[⋆]

Yong Jin[a,b], Jia Xu[a,*], Sixu Wu[a], Lijie Xu[a], Dejun Yang[c] and Kaijian Xia[d]

[a] *Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China*

[b] *School of Computer Science & Engineering, Changshu Institute of Technology, Changshu, Jiangsu 215500, China*

[c] *Colorado School of Mines, Golden, CO 80401*

[d] *The affiliated Changshu Hospital of Soochow University, Changshu, Jiangsu 215500, China*

ARTICLE INFO

Keywords:
Wireless rechargeable sensor network
Bus network
Drone scheduling
Traveling salesman path problem
Submodular orienteering problem

ABSTRACT

Wireless Rechargeable Sensor Network (*WRSN*) is largely used in monitoring of environment and traffic, video surveillance and medical care, etc., and helps to improve the quality of urban life. However, it is challenging to provide the sustainable energy for sensors deployed in buildings, soil or other places, where it is hard to harvest the energy from environment. To address this issue, we design a new wireless charging system, which levers the bus network assisted drone in urban areas. We formulate the drone scheduling problem based on this new wireless charging system to minimize the total time cost of drone subject to all sensors can be charged under the energy constraint of drone. Then, we propose an approximation algorithm *DSA* for the energy tightened drone scheduling problem. To make the tasks of *WRSN* sustainable, we further formulate the drone scheduling problem with deadlines of sensors, and present the approximation algorithm *DDSA* to find the drone schedule with the maximal number of sensors charged by the drone before deadlines. Through the extensive simulations, we demonstrate that *DSA* can reduce the total time cost by 84.83% compared with *Greedy Replenished Energy* algorithm, and uses at most 5.98 times of the total time cost of optimal solution on average. Then, we also demonstrate that *DDSA* can increase the survival rate of sensors by 51.95% compared with *Deadline Greedy Replenished Energy* algorithm, and can obtain 77.54% survival rate of optimal solution on average.

1. Introduction

Wireless Rechargeable Sensor Network (*WRSN*) plays an important role in urban life of smart city due to its advantage of sustainable power supply by the wireless charger network [1] and/or harvesting energy from environment [2], such as solar energy and wind energy. *WRSN* has been applied in many fields [3], such as long-term environmental monitoring [4] and vehicular traffic control application [5].

However, sensor networks deployed in inaccessible outdoor environment, such as precipitation analysis in mountains [6] and water quality monitoring [7], may incur higher cost of deployment and maintenance of wireless chargers. In addition, it is difficult to harvest energy from environment by using solar cells and/or wind energy collector in many sensing applications such as structural monitoring under bridges [8] and monitoring soil conditions [9].

The sensor can obtain energy from the wireless charger embedded drone or *Unmanned Aerial Vehicle* (*UAV*) [10], and store the energy in its capacity. [11] explored the feasibility of charging the sensors using drones that can wirelessly transfer energy to the sensors. [12] used the dedicated chargers carried by drones that can fly over the sensor network and transmit energy to the sensors using radio-frequency signals. The drone-enabled wireless charging for *WRSN* can sustainably replenish energy for the sensors deployed in inaccessible outdoor environment without the deployment and maintain of wireless charger network.

Due to the limited battery capacity, the drone has to return back to the ground charging stations to replenish energy for itself. This increases the energy consumption of the drone flight and decreases the charging efficiency. Due to the limited energy capacity, it is difficult for drone to charge the sensors deployed in a vast area. How to charge the drone efficiently is an interesting and significant problem, and has attracted a lot of attention. The latest research [13] proposed a solution of drone charging by riding buses to continuously collect and communicate video streams from a large number of *Points of Interests* (*PoI*s) in urban areas. [14] designed a new EV charging system, which levers the bus network in urban areas through the integration of *OnLine Electric Vehicle* (*OLEV*) system [15] and *Microwave Power Transfer* (*MPT*) system. By leveraging the bus network, the drone can not only replenish energy by riding on buses, but also extend the range of charging service. Meanwhile, the bus has a large capacity battery that can sustainably collect the energy from *OLEV* system or its fuel engine, and therefore, has sufficient energy to charge the drone. Moreover, the buses provide pervasive charging opportunities for the drone because of the high popularity and wide coverage of bus network in urban

[⋆] This work was supported in part by the NSFC under grants No. 61872193 and 62072254, Postgraduate Research & Practice Innovation Program of Jiangsu Province under grant No. SJKY190760, Natural Science Foundation of the Jiangsu Higher Education Institutions of China under grant No. 19KJB520020, and NSF under grant No. 1717315.

[*] Corresponding author

2018070263@njupt.edu.cn (Y. Jin); xujia@njupt.edu.cn, (J. Xu); 1019041115@njupt.edu.cn (S. Wu); ljxu@njupt.edu.cn (L. Xu); djyang@mines.edu (D. Yang); kjxia@suda.edu.cn (K. Xia)

ORCID(s):

areas.

However, the existing works studied drone-enabled wireless charging for *WRSN* [12] and drone scheduling using buses to charge the drone [13], separately. Actually, the energy between *WRSN*, drone and buses should be transferred efficiently to form a closed wireless charging system. Thus, we proposed a bus network assisted drone-enabled wireless charging system for *WRSN* in urban area. The introduction of bus network in the designed system not only accelerates the energy replenishment of drone, but also reduces the flying energy consumption of drone for charging *WRSN*. A toy example of our charging system is illustrated in Fig. 1. There are one drone, two buses and three sensors in the charging system. The drone can launch from any sensor and a set of fixed locations (termed landing points) on the bus routes. The buses have the regular schedules of themselves and can charge the drone when the drone rides the buses. The sensors can be charged by the drone. Then, the *WRSN* and the bus network together form a comprehensive network, which is consisted of sensors, landing points connecting road segments of bus routes, and flight segments between sensors and landing points. The drone rides the bus via the nearest landing point to the charged sensor for replenishing energy of itself from the bus and leaves the bus for charging the next sensor at some landing point when it has sufficient energy. Then the drone charges the sensor, and flies back to the nearest landing point to the charged sensor. Therefore, in this comprehensive network, the drone obtains the energy between any two landing points and consumes energy between any sensor and landing point.

In the designed system, the charging efficiency of drone and the sustainability of *WRSN* largely dependent on the drone scheduling. Unfortunately, to the best of our knowledge, there is no off-the-shelf bus network assisted drone scheduling for charging *WRSN*. We consider two drone scheduling scenarios according to the different requirements of sensing tasks. For the first scenario, we consider that the sensing tasks can tolerate some data loss, and allow the sensors to go to sleep for saving their energy. Thus, the drone scheduling is only constrained by the energy of drone in this case. In the second scenario, the sensing tasks require the continuous sensing data (such as vehicular traffic control application [5] and real-time environmental monitoring [16]), and therefore, the dead or dormancy of sensors will largely degrade the sensing quality. Thus, the drone scheduling is constrained by both energy of drone and deadlines of sensing tasks of sensors.

It is very challenging to schedule bus network assisted drone for sustainable charging of *WRSN*. First, it is impossible to obtain the travelling path of drone by solving the *Traveling Salesman Path Probelm* (*TSPP*) directly on the comprehensive network integrated by *WRSN* and bus network because our objective is to schedule drone for visiting the sensors only. Second, it is difficult to find the energy constrained shortest path of drone riding the buses from a sensor to the next sensor in the comprehensive network. This is because the drone may have the hybrid process of discharging and charging between any two sensors. Thus, the restricted shortest path algorithm [17] cannot be used to find our energy constrained shortest path straightforwardly because the restricted shortest path algorithm requires that the constrained metric should be non-negative. Moreover, the schedule of drone must ensure that each road/flight segment satisfies the energy constraint of drone, i.e., the residual energy of drone at the starting point of the road/flight segment is not less than the consumed energy passing through the segment. However, the residual energy of drone depends on the previously selected road/flight segments.

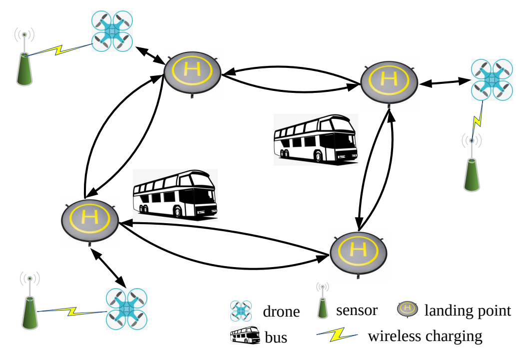


Figure 1: Bus network assisted drone scheduling for *WRSN*.

Our key contributions can be summarized as follows:

- We design a wireless charging system for *WRSN* through the bus network assisted drone in urban areas. To the best of our knowledge, we are the first to study the drone scheduling problem for such comprehensive wireless charging system.
- We formulate the problem of *Drone Scheduling with Bus network* (*DSB*) to minimize the time cost of drone for charging all sensors under the energy constraint of drone, and propose an approximation algorithm, *Drone Scheduling Algorithm* (*DSA*), to solve the energy tightened *DSB* problem.
- Considering the continuous sensing tasks of *WRSN*, we further formulate the *Deadline Drone Scheduling with Bus network* (*DDSB*) problem to maximize the number of charged sensors under the constraints of both energy of drone and deadlines of sensors, and we present an approximation algorithm, *Deadline Drone Scheduling Algorithm* (*DDSA*), to solve the energy tightened *DDSB* problem.
- We conduct extensive simulations and field experiments for the designed algorithms. The simulation results show that *DSA* can reduce the total time cost by 84.83% compared with *Greedy Replenished Energy* algorithm, and uses at most 5.98 times of the total time cost of optimal solution on average. Then, *DDSA* can increase the survival rate of sensors by 51.95% compared with *Deadline Greedy Replenished Energy* algorithm, and obtain 77.54% survival rate of optimal solution on average.

The rest of the paper is organized as follows. We review the state-of-art research in Section 2. We present the system model, formulate the *DSB* problem and present an approximation algorithm for the energy tightened *DSB* problem in Section 3. We formulate the *DDSB* problem and propose the approximation algorithm for the energy tightened *DDSB* problem in Section 4. We conduct the simulations and field experiments in Section 5 and 6, respectively. We conclude this work in Section 7.

2. Related Work

The recent research on wireless charging for *WRSN* mainly aimed to solve the problems of wireless charger network deployment and management [18, 19], energy harvesting [20, 21], and route scheduling of drone and wireless charging vehicle [22, 23, 24].

Some works studied the deployment of wireless sensors and charging scheduling of wireless chargers. [18] solved the robust scheduling problem for wireless charger network by considering power jittering and rechargeable device drifting. [19] jointly optimized the node position and charging allocation to improve the charging utility.

A number of studies concluded that harvesting energy from the immediate surroundings of the deployed sensors can effectively extend the lifetime of *WRSN*. [20] reviewed the energy-harvesting *WRSN* for environmental monitoring applications and presented the technologies for harvesting energy from ambient sources. [21] built a self-sustainable network and guaranteed operation under any weather condition by integrating the multi-source energy harvesting and wireless charging.

However, the deployment and maintenance of wireless chargers may occupy lots of cost when *WRSN* is deployed in inaccessible environment. Moreover, it is difficult to replenish the energy for the sensors by harvesting energy from environment when sensors are deployed inside buildings, bridges, and soil, etc.

Some studies considered to charge sensors through the drone or wireless charging vehicle. [22] designed a multi-drone wireless charging scheme for *WRSN* and proposed the route association algorithm to maximize the overall charging coverage utility by jointly selecting the charging routes and associated nodes. [23] proposed the dynamic path generation scheme to arrange the travelling path of wireless charging vehicle for minimizing energy consumption of the vehicle while ensuring that no sensor runs out of energy. [24] proposed an intelligent routing strategy for a wireless charging vehicle, which can find a traveling path in sensor networks to minimize the energy consumption for both traveling and charging to sensors.

On the research of drone charging and scheduling, [25] proposed deployment strategies for consumer *UAV*s to maximize the stationary coverage of a target area and to guarantee the continuity of the service through replenishing energy at ground charging stations. [26] proposed a wireless power transfer-based opportunity-charging scheme to extend the flight range by providing harvested energy gathered from renewable resources to the drones. [27] improved the drone-in-flight wireless charging platform by applying the nonlinear parity-time-symmetric model. [28] proposed an optimal design of asymmetric coupling system for drone wireless charging to overcome the transmission power shortage caused by drone landing errors. [29] proposed an iterative auction-based algorithm for optimal charge scheduling among drones.

Our work is fundamentally different from the existing research. Our charging system is based on a comprehensive network by integrating the *WRSN* and bus network. To the best of our knowledge, we are the first to consider the bus network assisted drone as a mobile wireless charger for *WRSN*. We overcome the battery capacity limitation of drone by riding buses without the support of ground charging stations. Then, we find the energy constrained shortest path and compute the final feasible charging cycle of drone to charge all sensors. For the continuous sensing tasks, we find the energy constrained path to charge as many sensors as possible before their deadlines.

3. Drone Scheduling with Bus Network

3.1. System Model and Problem Definition

We consider a sparse *WRSN* consisting of a set of rechargeable sensors with fixed known positions. There are a bus network and a drone that is responsible for charging the sensors in an urban area. The drone cannot fly directly between any two sensors because of the limited battery capacity, and must ride the buses to charge itself before charging any sensor. We consider that the sensing tasks can tolerate some data loss, therefore, the sensors can go to sleep to save their energy and prolong their lifetime. The sensors can be awakened by replenishing energy from the drone. The drone rides the bus at the nearest landing point to a last charged sensor for replenishing energy from the bus and then leaves the bus for charging the next sensor at some landing point when it has sufficient energy. Then the drone charges the sensor, and flies back to the nearest landing point to the charged sensor. There are a set of predetermined locations, termed landing points, which can be viewed as the transfer locations for connecting *WRSN* and bus network. The drone must launch at a landing point to charge a sensor. Similarly, the drone also must land at a landing point to ride the bus for energy replenishment. The introduction of landing points largely reduce the difficulty of landing on a moving bus. The landing point also helps to reduce the energy consumption because when it is waiting the bus, the drone can land on the landing point rather than hovering. In practice, the bus stops can be redeveloped as landing points. Thus, a bus route consists of some consecutive landing points.

Without loss of generality, let S be the set of n sensors and V be the set of m landing points. Let β_k and φ_k be the moving speed and charging power of bus $b_k \in B$, respectively, where B is the set of buses. Let β_0 denote the flying speed of drone. Let t_s and t_d denote the ascending time and descending time of drone, respectively. Let c_0 and c_h denote the flying energy consumption per unit distance and hovering energy consumption per unit time, respectively. Let c_s and c_d denote the energy consumption of ascending and descending of drone, respectively. Let C_{max} denote the battery capacity of drone. Let φ_0 denote the charging power of drone. For convenience, we assume that the drone flies at a constant speed and the flying energy consumption is only related to the flying distance. The drone is initially located at a sensor, denoted by s_0. Note that s_0 can be any sensor in S.

Let $v(s_j)$ and $v(s_{j'})$ represent the nearest landing point to sensor $s_j, s_{j'} \in S$, respectively. Let $v_i \in V$ denote a landing point where the drone launches for charging sensor $s_{j'}$. Note that the drone cannot fly directly between any two sensors. Without loss of generality, the drone path from s_j to $s_{j'}$ can be represented as

$$p_{s_j,s_{j'}} = s_j \rightarrow v(s_j) \rightarrow ... \rightarrow v_i \rightarrow s_{j'} \tag{1}$$

To formalize the comprehensive charging system, we use the directed multigraph $G = (V \cup S, A)$ to represent the comprehensive network by integrating *WRSN* and bus network, where A represents the set of road segments that connect the adjacent landing points in bus routes and flight segments that connect the sensors and landing points. The weight on any edge $a \in A$ is time cost t_a, which represents the consumed time when passing through the edge. Let c_a be the energy cost of drone that represents the energy consumption when drone passes through edge a. Let e_a be the residual energy of drone after passing through starting point of edge a. Let $r(s_j)$ be the energy requirement of any sensor $s_j \in S$. Let $\parallel \langle v_i, v_{i'} \rangle \parallel$, $\parallel \langle v_i, s_{j'} \rangle \parallel$ and $\parallel \langle s_j, v(s_j) \rangle \parallel$ be the distance from v_i to $v_{i'}$, from v_i to $s_{j'}$ and from s_j to $v(s_j)$, respectively. Specifically, there are three types of edges in the directed multigraph G:

(1) $a = \langle v_i, v_{i'} \rangle_{b_k}$ indicates the edge from landing point v_i to landing point $v_{i'}$ taking bus b_k with weight t_a and energy cost c_a, where t_a is the traveling time of bus passing through edge a, and c_a is the minimum value between replenished energy when passing through edge a and maximum rechargeable capacity after passing through starting point of edge a. Then, t_a and c_a can be calculated by the following equations.

$$t_a = \frac{\parallel \langle v_i, v_{i'} \rangle \parallel}{\beta_k} \tag{2}$$

$$c_a = -\min\{\varphi_k t_a, C_{max} - e_a\} \tag{3}$$

(2) $a = \langle v_i, s_{j'} \rangle$ indicates the edge that the drone flies from landing point v_i to sensor $s_{j'}$ with weight t_a and energy cost c_a, where t_a is the total time of ascending, flying from v_i to $s_{j'}$ and charging for $s_{j'}$, and c_a is the total energy consumption of ascending, flying from v_i to $s_{j'}$, and hovering of drone, as well as energy requirement of $s_{j'}$. Then, t_a and c_a can be calculated by the following equations.

$$t_a = t_s + \frac{\parallel \langle v_i, s_{j'} \rangle \parallel}{\beta_0} + \frac{r(s_{j'})}{\varphi_0} \tag{4}$$

$$c_a = c_s + \parallel \langle v_i, s_{j'} \rangle \parallel c_0 + c_h \frac{r(s_{j'})}{\varphi_0} + r(s_{j'}) \tag{5}$$

(3) $a = \langle s_j, v(s_j) \rangle$ indicates the edge that the drone flies from sensor s_j to the nearest landing point $v(s_j)$ with weight t_a and c_a, where t_a is the total time of flying from s_j to $v(s_j)$ and descending of drone, and c_a is the total energy consumption of flying and descending of drone. Then, t_a and c_a can be calculated by the following equations.

$$t_a = \frac{\parallel \langle s_j, v(s_j) \rangle \parallel}{\beta_0} + t_d \tag{6}$$

$$c_a = \parallel \langle s_j, v(s_j) \rangle \parallel c_0 + c_d \tag{7}$$

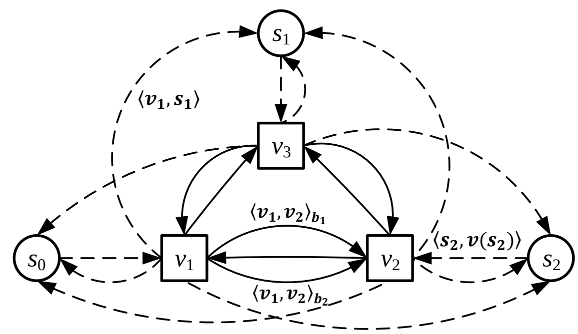


Figure 2: Illustration of directed multigraph G.

We use the example in Fig. 2 to illustrate the directed multigraph G. There are three sensors and three landing points along with two buses and a drone. We assume that $v(s_1) = v_1$, $v(s_2) = v_2$ and $v(s_3) = v_3$. There are two edges from v_1 to v_2, i.e., $\langle v_1, v_2 \rangle_{b_1}$ and $\langle v_1, v_2 \rangle_{b_2}$, which indicate the bus $b_1, b_2 \in B$ can drive from v_1 to v_2, respectively. The edge $\langle v_1, s_1 \rangle$ and $\langle s_2, v(s_2) \rangle$ indicates the drone flies from landing point v_1 to sensor s_1 and from sensor s_2 to the nearest landing point $v(s_2)$, respectively.

Let Ω denote all possible drone schedules that start from any given sensor s_0 and can charge all sensors exactly once under the energy constraint of drone. The objective of *DSB* problem is to find the drone schedule with minimum time cost. The *DSB* problem can be formulated as follows:

$$\begin{aligned} &\textbf{(P1)} \qquad \min_{T \in \Omega} \textstyle\sum_{a \in T} t_a \\ &s.\ t. \\ &\text{(P1-a)}\ c_a \le e_a, \forall a \in T \end{aligned} \tag{8}$$

Constraint (P1-a) ensures that the energy cost of each edge $a \in T$ is not larger than the residual energy of drone at the starting point of edge a.

We listed the frequently used notations in Tab. 1.

Table 1
Frequently Used Notations

Notation	Description
S, V	Set of sensors, set of landing points
n, m	Number of sensors, number of landing points
B, β_k, φ_k	Set of buses, moving speed of bus b_k, charging power of bus b_k
β_0	Flying speed of drone
t_s, t_d	Ascending time of drone, descending time of drone
c_0, c_h	Flying energy consumption per unit distance of drone, hovering energy consumption per unit time of drone
c_s, c_d	Ascending energy consumption of drone, descending energy consumption of drone
C_{max}, φ_0	Battery capacity of drone, charging power of drone
v_i	Landing point
$v(s_j)$	Nearest landing point to sensor s_j
$p_{s_j,s_{j'}}$	Drone path from sensor s_j to $s_{j'}$
G	Comprehensive network
A	Set of edges in G
t_a, c_a	Time cost of edge a, energy cost of edge a
$\langle v_i, v_{i'} \rangle_{b_k}$	Edge from landing point v_i to landing point $v_{i'}$ taking bus b_k
$r(s_j), \tau(s_j)$	Energy requirement of sensor s_j, deadline of sensor s_j
Ω	All possible drone schedules of *DSB* problem
$\bar{p}_{s_j,s_{j'}}$	Modified drone path from sensor s_j to sensor $s_{j'}$
Φ	All the paths from landing point $v(s_j)$ to landing point v_i on G
Γ_{s_j}	All possible directed Hamiltonian paths that start from sensor s_0 to sensor s_j
$S(\mathcal{T})$	Set of sensors charged by drone passing through cycle $\mathcal{T}$
Λ	All possible drone schedules of *DDSB* problem

3.2. Hardness and Design Rationale

As the following theorem shows, it is NP-hard to find the optimal solution for the *DSB* problem.

Theorem 1. *DSB problem is NP-hard.*

Proof. Consider the special case where G only contains sensors and the drone can fly between any two sensors by assuming that the energy of drone is infinite. Then, the problem is simplified to find a drone schedule starting from s_0 to charge all sensors exactly once with minimum time cost. This problem is the *TSPP* problem actually. Since the *TSPP* problem is a well-known NP-hard problem [30], the *DSB* problem is NP-hard □

The basic idea of our solution is to find the energy constrained drone paths with the minimum time cost between any two sensors. If these drone paths can be determined, we can construct a directed graph G' only consisting of sensors and the drone paths. Then the **P1** problem can be solved by calculating the *TSPP* of drone on G'.

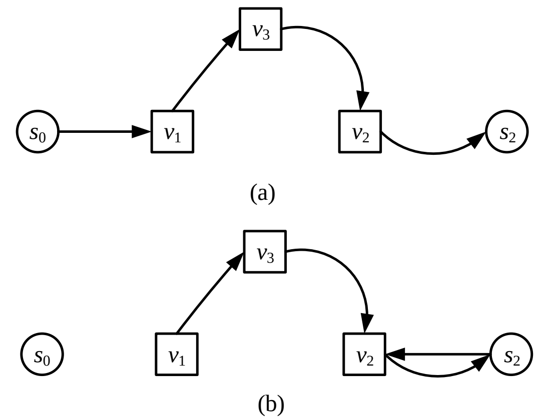


Figure 3: Illustration of transformation of drone path. (a) the original drone path $p_{s_0,s_2} = s_0 \rightarrow v_1 \rightarrow v_3 \rightarrow v_2 \rightarrow s_2$. (b) the modified drone path $\bar{p}_{s_0,s_2} = v_1 \rightarrow v_3 \rightarrow v_2 \rightarrow s_2 \rightarrow v_2$.

However, it is difficult to find the energy constrained shortest path (with the minimum time cost) because the drone may have the hybrid process of discharging and charging between any two sensors. This means that the restricted shortest path algorithm [17], which requires the non-negative constraint, cannot be applied straightforwardly.

To solve this problem, we transform the original drone path given in (1) from sensor s_j to $s_{j'}$ to the modified path $\bar{p}_{s_j,s_{j'}}$:

$$\bar{p}_{s_j,s_{j'}} = v(s_j) \rightarrow ... \rightarrow v_i \rightarrow s_{j'} \rightarrow v(s_{j'}) \tag{9}$$

Here, the drone passing through $\bar{p}_{s_j,s_{j'}}$ is charged first in the subpath $v(s_j) \rightarrow ... \rightarrow v_i$, and then discharge in the subpath $v_i \rightarrow s_{j'} \rightarrow v(s_{j'})$. So, $\bar{p}_{s_j,s_{j'}}$ can satisfy the energy constraint (P1-a) when the replenished energy from buses in the subpath $v(s_j) \rightarrow ... \rightarrow v_i$ is no less than the total energy consumption in the subpath $v_i \rightarrow s_{j'} \rightarrow v(s_{j'})$. The above transformation of drone path is illustrated in Fig. 3, where the original drone path is $p_{s_0,s_2} = s_0 \rightarrow v_1 \rightarrow v_3 \rightarrow v_2 \rightarrow s_2$ shown in Fig. 3 (a). The modified drone path is $\bar{p}_{s_0,s_2} = v_1 \rightarrow v_3 \rightarrow v_2 \rightarrow s_2 \rightarrow v_2$ shown in Fig. 3 (b).

Another challenge is that the initial energy of the drone path is uncertain. As illustrated in Fig. 3 (b), the energy of drone at landing point v_1 depends on the previously passed segments and cannot be calculated in advance. To address this issue, we assume that the residual energy of drone at the nearest landing point to the last charged sensor is 0, which is sufficient to ensure the energy feasibility. This assumption essentially tightens the energy constraints of **P1**. For the given landing point v_i and sensor $s_{j'}$, the total energy consumption can be computed in advance. So, the energy constrained shortest path problem can be solved by finding the shortest path $p_{v(s_j),v_i}$ from $v(s_j)$ to $v_i \in V$ with energy replenishment no less than a constant.

Let Φ be all the paths from $v(s_j)$ to v_i on G. Given any two sensors s_j and $s_{j'}$, and any landing point $v_i \in V$, the energy constrained shortest path problem can be formulated as follows:

$$\textbf{(P2)} \min_{p_{v(s_j),v_i} \in \Phi} \sum_{a \in p_{v(s_j),v_i}} t_a, s_j \in S, v_i \in V$$
$$s.\ t.$$
$$\text{(a)} \sum_{a \in p_{v(s_j),v_i}} |c_a| \geq c_{\langle v_i, s_{j'} \rangle} + c_{\langle s_{j'}, v(s_{j'}) \rangle}, \quad s_j, s_{j'} \in S, v_i \in V \tag{10}$$

Constraint (P2-a) ensures that the replenished energy of $p_{v(s_j),v_i}$ is no less than the total energy cost of $\langle v_i, s_{j'} \rangle$ and $\langle s_{j'}, v(s_{j'}) \rangle$.

Recall that if **P2** can be solved, the directed graph $G' = (S, A')$ can be constructed, where A' represents the set of the energy constrained shortest paths between any two sensors. The weight on any edge $a' \in A'$ is time cost $t_{a'}$, which indicates the consumed time when drone passes through the edge a'. Unfortunately, it is NP-hard to find the optimal solution for the **P2** problem [31]. Furthermore, to the best of our knowledge, there is no approximation algorithm to address the **P2** problem. Thus, we find the exact solution of the **P2** problem through $YenKSP$ algorithm [32] with probability .

If **P2** can be solved by an exact solution, the energy tightened **P1** problem on G' is equivalent to the *Asymmetric TSPP* (*ATSPP*) on G' [30]. Let Γ_{s_j} denote all possible directed Hamiltonian paths that start from s_0 to $s_j \in S$ and can charge all sensors exactly once. The objective of *ATSPP* is to find a directed Hamiltonian path from s_0 to s_j with minimum time cost on G'. The *ATSPP* on G' is defined as follows:

$$\textbf{(P3)} \min_{T_{s_j} \in \Gamma_{s_j}} \sum_{a' \in T_{s_j}} t_{a'}, s_j \in S \tag{11}$$

Then, the **P3** problem can be solved by $ATSPP\_Approx$ algorithm [30]. Finally, we find all the minimum time cost directed Hamiltonian paths from s_0 to each sensor $s_j \in S$, and return the minimum time cost path from the solution of **P3** as the final cycle T.

3.3. Algorithm Design and Analysis

In this subsection, we present the approximation algorithm *DSA* to solve the energy tightened **P1** problem.

The whole process of solving the energy tightened *DSB* problem is illustrated in Algorithm 1. Following the design rationale mentioned above, *DSA* consists two phases:

Phase 1: Directed Graph Construction

Let $P_{v(s_j),v_i}$ denote the K-shortest paths from $v(s_j)$ to $v_i \in V$. Let $\mathcal{P}_{s_j,s_{j'}}$ denote all energy constrained shortest paths from s_j to $s_{j'}$, where drone launches at each feasible $v_i \in V$ for charging sensor $s_{j'}$. For any two sensors $s_j, s_{j'} \in S$, we find the first K-shortest paths $P_{v(s_j),v_i}$ from $v(s_j)$ to any landing point $v_i \in V$ by $YenKSP(G, v(s_j), v_i, K)$ (Line 8). Then, we find the path $p_{v(s_j),v_i}$ with minimum time cost from $P_{v(s_j),v_i}$ (Line 9) and put it into set $\mathcal{P}_{s_j,s_{j'}}$ (Line 13) and terminate execution of while loop if $p_{v(s_j),v_i}$ is not empty, and update L otherwise (Line 11). Specifically, we use binary search to guess K over $[1, K_{max}]$ (Line 7), where K_{max} is a given positive integer. Then, the modified drone path $\bar{p}_{s_j,s_{j'}}$ is assembled by $p_{v(s_j),v_i}$, $\langle v_i, s_{j'} \rangle$ and $\langle s_{j'}, v(s_{j'}) \rangle$, where symbol $\uplus$ represents assembling the path (Line 14). Moreover, we add the tuple $(\langle s_j, s_{j'} \rangle, \sum_{a \in \bar{p}_{s_j,s_{j'}}} t_a)$ into graph G' (Line 15), where s_j and $s_{j'}$ are the vertexes in the edge of G', and the second term is the weight (time cost of the modified drone path $\bar{p}_{s_j,s_{j'}}$ on the edge $\langle s_j, s_{j'} \rangle$. Finally, the directed multigraph G is transformed into a directed graph G'.

Phase 2: ATSPP Calculation.

Let $\vec{T}$ denote all the directed Hamiltonian paths with minimum time cost from s_0 to any sensor $s_j \in S$, where $\vec{T}$ can be computed by $ATSPP\_Approx(G', s_0, s_j)$ (Lines 17-18), and return the minimum time cost path in the above paths as the final cycle T (Line 19).

Algorithm 1: ***Drone Scheduling Algorithm* (*DSA*)**

Input: G, K_{max}
Output: T;
`// Phase 1: Directed Graph Construction`
1 **foreach** $s_j \in S$ **do**
2   **foreach** $s_{j'} \in S, j' \neq j$ **do**
3     $\mathcal{P}_{s_j,s_{j'}} \leftarrow \emptyset$; $P_{s_j,s_{j'}} \leftarrow \emptyset$;
4   **foreach** $v_i \in V$ **do**
5       $L \leftarrow 1$; $F \leftarrow false$; $p_{v(s_j),v_i} \leftarrow \emptyset$;
6   **while** $L \leq K_{max}$ and $F = false$ **do**
7         $K \leftarrow \lfloor \frac{L+K_{max}}{2} \rfloor$;
8         $P_{s_j,s_{j'}} \leftarrow YenKSP(G, v(s_j), v_i, K)$;
9         $p_{s_j,s_{j'}} \leftarrow \arg\min_{p \in P_{s_j,s_{j'}}} \{\sum_{a \in p} t_a | \sum_{a \in p} |c_a| \geq c_{\langle v_i, s_{j'} \rangle} + c_{\langle s_{j'}, v(s_{j'}) \rangle}\}$;
10   **if** $p_{s_j,s_{j'}} = \emptyset$ **then**
11           $L \leftarrow K$;
12   **else**
13           $\mathcal{P}_{s_j,s_{j'}} \leftarrow \mathcal{P}_{s_j,s_{j'}} \cup \{p_{s_j,s_{j'}}\}$; $F \leftarrow true$;
14     $\bar{p}_{s_j,s_{j'}} \leftarrow \arg\min_{p_{v(s_j),v_i} \in \mathcal{P}_{s_j,s_{j'}}} \sum_{a \in p_{v(s_j),v_i}} t_a \uplus \{\langle v_i, s_{j'} \rangle\} \uplus \{\langle s_{j'}, v(s_{j'}) \rangle\}$;
15     $G' \leftarrow G' \cup \{(\langle s_j, s_{j'} \rangle, \sum_{a \in \bar{p}_{s_j,s_{j'}}} t_a\}$;
`// Phase 2: ATSPP Calculation`
16 $\vec{T} \leftarrow \emptyset$;
17 **foreach** $s_j \in S$ **do**
18   $T_{s_j} \leftarrow ATSPP\_Approx(G', s_0, s_j)$;
  $\vec{T} \leftarrow \vec{T} \cup \{T_{s_j}\}$;
19 $T \leftarrow \arg\min_{T' \in \vec{T}} \sum_{a' \in T'} t_{a'}$;

Theorem 2. *DSA is a pseudo-polynomial time and $\sqrt{n}$-approximation algorithm for the energy tightened DSB problem with probability* $2\frac{2^{\log K_{max}-1}}{\log K_{max}(K_{max}-1)}$.

Proof. In phase 1, $YenKSP$ is called $\log K_{max}$ times, and then finds all energy constrained shortest paths for any two sensors takes $O(Km^2n^2 \log K_{max}(|A|+m\log m))$ time, where $K \in [1, K_{max}]$ is a positive integer. In Phase 2, finding all the minimum time cost directed Hamiltonian paths from s_0 to any other sensor $s_j \in S$ takes $O(n^{4.5})$ time. Hence, *DSA* is a pseudo-polynomial time algorithm.

From Line 7, we have at most $\log K_{max}$ values of K, where the value of K in the q-th binary search can be calculated by

$$K_q = \frac{1}{2}(K_{q-1} + K_{max}), K_0 = 1, q = 1, ..., \log K_{max} \quad (12)$$

Then, we divide the interval $[1, K_{max}]$ into $\log K_{max}$ subintervals, i.e., $I_q = [K_{q-1}, K_q], q = 1, ..., \log K_{max}$. We define the event N_q to represent that we can find the optimal solution of **P2** problem in the subinterval I_q. Clearly, the above events are not compatible with each other, i.e., $N_q \cap N_{q'} = \emptyset, q \neq q', q, q' = 1, 2, ..., \log K_{max}$ and $\bigcup_{q=1}^{\log K_{max}} I_q = [1, K_{max}]$. Let M denote the event of finding the optimal solution of **P2** problem in interval $[1, K_{max}]$. So, according to the total probability theorem, the *DSA* can find the optimal solution of **P2** problem with the probability calculated by

$$\begin{aligned} P(M) &= \sum_{q=1}^{\log K_{max}} P(N_q)P(M|N_q) \\ &= \frac{1}{\log K_{max}(K_{max}-1)} \sum_{q=1}^{\log K_{max}} 2^q \\ &= 2\frac{2^{\log K_{max}-1}}{\log K_{max}(K_{max}-1)} \end{aligned} \quad (13)$$

Phase 2 adopts $ATSPP\_Approx$ to find the $\sqrt{n}$-approximation [30] Hamiltonian paths from s_0 to any other sensor in S on G'. Thus, the *DSA* is a $\sqrt{n}$-approximation algorithm for the energy tightened *DSB* problem with probability $2\frac{2^{\log K_{max}-1}}{\log K_{max}(K_{max}-1)}$. □

4. Deadline Drone Scheduling with Bus network

4.1. Problem Formulation

The continuous sensing application does not tolerate data loss. Therefore, the data quality will be deteriorated because of the data loss caused by the dead and dormancy of sensors. Due to the different energy consumption level of sensors, we consider that each sensor has a specified deadline. Generally, it is difficult to charge all sensors before their deadlines because of the limited flying speed and battery capacity of drone. A practical objective is to maximize the number of sensors charged by the drone before their deadlines. So, we are committed to make as many sensors as possible work sustainably under the constraints of both energy of drone and deadlines of sensors.

Let $\mathcal{T}$ denote the drone schedule on G starting from s_0 and charge other sensors under the constraints of both energy of drone and deadlines of sensors. To solve the drone scheduling problem considering the deadline, we find the drone schedule starting at s_0 to charge as many sensors as possible. Let $S(\mathcal{T}) \subseteq S$ denote the set of sensors charged by drone passing through $\mathcal{T}$. We define a function $f(S(\mathcal{T})) = |S(\mathcal{T})|$, which represents the number of sensors in $S(\mathcal{T})$. We have the following fact.

Fact 1. *The function f is a monotone submodular function.*

Let Λ denote all drone schedules on G starting from s_0 and charge other sensors under the constraints of both energy of drone and deadlines of sensors. Let $\tau(s)$ denote the deadline of sensor s, and $\mathcal{T}(s)$ denote the subpath of $\mathcal{T}$ from sensor s_0 to $s, s \in S$. The *DDSB* problem on G can be formulated as follows:

$$\begin{aligned} &\textbf{(P4)} \quad \max_{\mathcal{T}\in\Lambda} f(S(\mathcal{T})) \\ &s.\ t. \\ &\text{(P4-a)}\ c_a \leq e_a, \forall a \in \mathcal{T}; \\ &\text{(P4-b)}\ \textstyle\sum_{a\in\mathcal{T}(s)} t_a \leq \tau(s), \forall s \in S(\mathcal{T}) \end{aligned} \quad (14)$$

Constraint (P4-a) ensures that the energy cost of any edge $a \in \mathcal{T}$ is not larger than the residual energy of drone at the starting point of edge a. Constraint (P4-b) ensures that the total time of drone passing through $\mathcal{T}(s)$ is not larger than the deadline of any sensor $s \in S(\mathcal{T})$.

4.2. Algorithm Design and Analysis

First of all, as the following theorem shows, it is NP-hard to find the optimal solution for the *DDSB* problem.

Theorem 3. *DDSB problem is NP-hard.*

Proof. Consider the special case where G only contains sensors and the drone can fly between any two sensors by assuming that the energy of drone is infinite. Each sensor has a deadline. Let Λ^R denote all drone schedules on G starting at s_0 and charge other sensors before the deadlines of sensors. Let $\mathcal{T}^R(s)$ denote the subpath of $\mathcal{T}^R \in \Lambda^R$ from sensor s_0 to sensor $s, s \in S(\mathcal{T}^R)$. The objective of the relaxed *DDSB* problem is to find a path starting at s_0 that charges as many sensors as possible before their deadlines. Then, the relaxed *DDSB* problem can be formulated as follows:

$$\begin{aligned} &\textbf{(P5)} \quad \max_{\mathcal{T}^R\in\Lambda^R} f(S(\mathcal{T}^R)) \\ &s.t. \\ &\text{(P5-a)}\ \textstyle\sum_{a\in\mathcal{T}^R(s)} t_a \leq \tau(s), \forall s \in S(\mathcal{T}^R) \end{aligned} \quad (15)$$

Constraint (P5-a) ensures that the time cost of $\mathcal{T}^R(s)$ for any sensor $s \in S(\mathcal{T}^R)$ is not larger than the deadline of sensor s.

Then we give the instance of *Deadline-Traveling Salesman Problem* (*Deadline-TSP*): For a metric space $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ on n nodes, a starting node u_s and deadlines $\tau(u)$ for each vertex $u \in \mathcal{V}$, the objective of *Deadline TSP* [33] is to find a path starting at u_s that visits as many nodes as possible before their deadlines.

We can simply see that z is a solution of *Deadline TSP* if and only if z is a solution of **P5**.

Since the *Deadline TSP* is a well-known NP-hard problem [33], **P5** is NP-hard, and then the *DDSB* problem is NP-hard. □

To overcome the uncertainty of residual energy at each drone path. We still tighten the energy constraints of **P4** by assuming that the residual energy of drone at the nearest landing point to the last charged sensor is 0. So, the energy constrained shortest path with the minimum time cost is the optimal path between any two sensors. Then, we still use modified path given in (9) because of the hybrid process of discharging and charging between any two sensors. For the same reasons clarified in Section **3.2**, we use the Phase 1 of *DSA* to find the energy constrained drone paths with the minimum time cost between any two sensors and construct the directed graph $G' = (S, A')$ only consisting of sensors and the drone paths.

Then, we aim at solving **P5** on G'. Let $\mathcal{T}_{s_j}$ be a cycle from s_0 to s_j on G', where $\mathcal{T}_{s_j}$ can be represented as

$$\mathcal{T}_{s_j} = s(1) \rightarrow ... \rightarrow s(l) \rightarrow ... \rightarrow s(f(S(\mathcal{T}_{s_j}))), \quad l = 1, 2, ..., f(S(\mathcal{T}_{s_j})) \tag{16}$$

Each sensor $s \in S$ has a time window $[R(s), \tau(s)]$ during which it can be visited, where $R(s)$ is the release time of s. Let $t(s(l))$ be the time cost from s_0 to $s(l)$ on cycle $\mathcal{T}_{s_j}$. Given G', two sensors s_0, s_j, and a budget $\tau(s_j)$, the *Submodular Orienteering Problem-Time Windows* (*SOP-TW*) [34] is to find a cycle $\mathcal{T}_{s_j}$ that maximizes $f(S(\mathcal{T}_{s_j}))$ when the following conditions are satisfied: $t(s(l)) \leq \tau(s_j)$ and $t(s(l)) + t_{\langle s_l, s_{l+1}\rangle} \leq \tau(s_{l+1})$, $l = 1, 2, ..., f(S(\mathcal{T}_{s_j}))$.

Clearly, **P5** on G' is a special case of *SOP-TW* when we set the release time of all sensors as 0, and the reward function is a modular function (i.e., one for which the submodular inequality holds with equality). Since *SOP-TW* can be solved by RGA algorithm [34] within the approximation of $\log f(S(\mathcal{T}^\star))$, where $\mathcal{T}^\star$ is the optimal solution of energy tightened *DDSB* problem, **P5** (*Deadline TSP*) on G' can be solved by RGA algorithm within the approximation of $\log f(S(\mathcal{T}^\star))$.

We present the approximation algorithm *DDSA* to solve the energy tightened *DDSB* problem. The whole process of solving the energy tightened *DDSB* problem is illustrated in Algorithm 2, which consists of following phases.

Phase 1: Directed Graph Construction

This phase is the same to Phase 1 in *DSA*.

Phase 2: SOP-TW Calculation

We find the cycles charging as many sensors as possible from s_0 to each sensor $s_j \in S$ by $RGA(G', s_0, s_j, \tau(s_j))$ (Lines 3-4), and return the path with maximum charged sensors in the above cycles as the final cycle $\mathcal{T}$ (Line 5).

Theorem 4. *DDSA is a quasi-polynomial time and* $\log f(S(\mathcal{T}^\star))$*-approximation algorithm for the energy tightened DDSB problem with probability* $2\frac{2^{\log K_{max}-1}}{\log K_{max}(K_{max}-1)}$, *where* $\mathcal{T}^\star$ *is the optimal solution of energy tightened DDSB problem.*

Proof. Let τ_{max} denote the maximum deadline in all sensors, i.e., $\tau_{max} = \max_{s \in S}\{\tau(s)\}$. In Phase 2, finding all the cycles charging as many sensors as possible from s_0 to each sensor in S takes $O(n(n \log \tau_{max})^{O(\log n)})$ time. Thus, the *DDSA* is a quasi-polynomial algorithm.

Algorithm 2: ***Deadline Drone Scheduling Algorithm*** **(*DDSA*)**

Input: $G, K_{max}, \tau(s_j), s_j \in S$
Output: T;

```
// Phase 1: Directed Graph Construction
1 call the process of Lines 1-15 in DSA;
// Phase 2: SOP-TW Calculation
2 T⃗ ← ∅;
3 foreach s_j ∈ S do
4     T_{s_j} ← RGA(G', s_0, s_j, τ(s_j)); T⃗ ← T⃗ ∪ {T_{s_j}};
5 T ← arg max_{T' ∈ T⃗} f(S(T'));
```

Let $\mathcal{T}^\star$ be the optimal solution of energy tightened *DDSB* problem. Phase 2 adopts RGA to find the $\log f(S(\mathcal{T}^\star))$ approximation [34] cycle from s_0 to any sensor $s_j \in S, j \neq 0$ on G'. Thus, the *DDSA* can output the solution for energy tightened *DDSB* problem with number of sensors charged by drone no less than $\log f(S(\mathcal{T}^\star))$ with probability $2\frac{2^{\log K_{max}-1}}{\log K_{max}(K_{max}-1)}$. □

5. Numerical Experiments

In this section, we conduct extensive simulations to verify the performance of our proposed algorithms with different number of landing points, energy requirement, number of sensors and deadlines of sensors.

5.1. Simulation setup

We use the data of 'New York City Bus Data' [35], which includes the live data recorded from NYC Buses. This dataset is from the NYC MTA bus data stream service.

To compare the proposed algorithms with the benchmark algorithms, we select the bus routes from the dataset to create the large-scale transportation network. We compute the length of road/flight segments through Google map. Suppose that we deploy rechargeable sensors in Boston, New York. The parameters of drone are from Inspire 2 [36]. In our simulation, we evaluate the total time cost of drone, and survival rate of sensors. All the simulations were run on a cloud server ECS [37] with 8 core Intel Xeon Platinum 8269CY and 32 GB memory. The maximum hovering time of drone is about 0.45 *hours*. The other parameter settings of our simulations are listed in Tab. 2.

5.2. Benchmarks

Since there are no existing algorithms for bus network assisted drone scheduling, we develop four benchmark algorithms for comparison,

(1) *GRE*. The *Greedy Replenished Energy* (*GRE*) algorithm consists of Directed Graph Construction Phase and AT-SPP Calculation Phase. In Phase 1, *GRE* finds the maximum replenished energy paths on G for any two sensors greedily [38]. Phase 2 is the same to that of *DSA*.

Table 2
Parameter settings

Parameter	Value
n	500
m	22
$\|B\|$	23
C_{max}	97.58 Wh
β_0	94 km/h
t_s, t_d	0.02 $hours$
c_s, c_d	0.54 Wh
c_0	1.07 Wh/km
φ_k	80 kW
φ_0	0.04 kW
$r(s_j)$	[5, 20] Wh
$\tau(s_j)$	[2, 12] $hours$

(2) *DGRE*. The *Deadline Greedy Replenished Energy* (*DGRE*) algorithm consists of Directed Graph Construction Phase and SOP-TW Calculation Phase. In Phase 1, the *DGRE* finds the maximum replenished energy paths on G for any two sensors greedily [38]. Phase 2 is the same to that of *DDSA*.

(3) *OPT*. The optimal solution for energy tightened *DSB* problem. *OPT* enumerates all the paths for any two sensors on G and selects the energy constrained path with the minimum time cost to construct G'. Then, *OPT* enumerates all the directed Hamiltonian paths starting from s_0 and charging all sensors on G', and then returns the minimum time cost path as the final solution.

(4) *DOPT*. The optimal solution for energy tightened *DDSB* problem. *DOPT* enumerates all the paths for any two sensors on G and selects the energy constrained shortest path with the minimum time cost to construct G'. Then, *DOPT* enumerates all the paths starting from s_0 and charging other sensors before deadlines on G', and then returns the path with maximum charged sensors as the final solution.

5.3. Performance evaluation

In this subsection, we evaluate the performance of *GRE, DSA, DGRE* and *DDSA* in the large-scale transportation network shown in Fig. 4. Tab. 3 gives the schedules of bus lines in the large-scale transportation network including bus ID, sub route, route length, and average speed of bus. The above information are calculated based on the data records from 0:00-23:59 on December 1-6, 2017 in Boston.

Impact of number of landing points. We vary number of landing points from 10 to 22. Fig. 5 shows that *DSA* reduces 96.35% of total time cost of *GRE* on average, and *DDSA* increases 76.49% of survival rate of sensors of *DGRE* on average. This indicates that the proposed algorithms significantly outperform *GRE* and *DGRE*, respectively. This is because the output of *DSA* and *DDSA* are paths with the minimum time cost satisfying the energy constraint of drone, and are better than the paths obtained by *GRE* and *DGRE*, respectively.

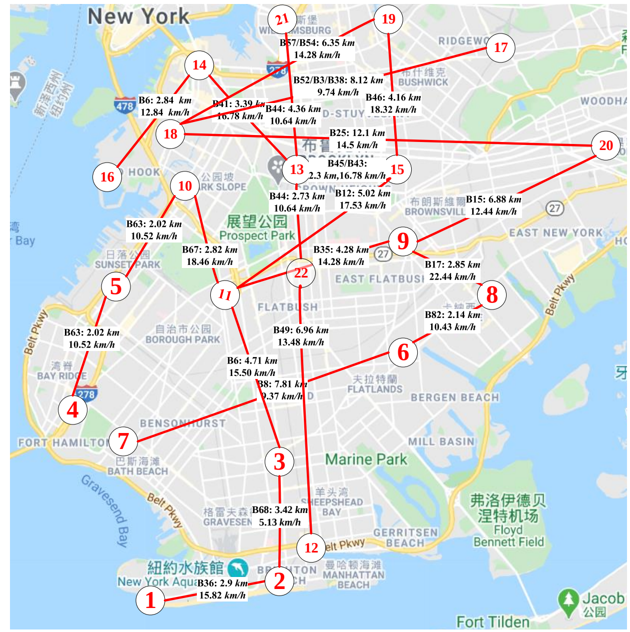


Figure 4: Large-scale transportation network. The white nodes represent landing points and the red lines represent road segments. There are three parameters on each road segment, i.e., bus ID, length of road segment and average speed of bus passing through the road segment.

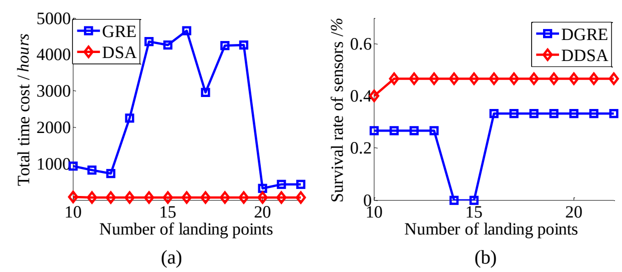


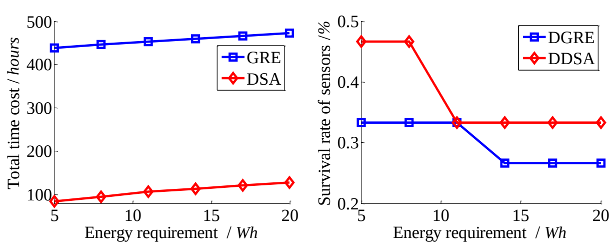
Figure 5: Impact of number of landing points: (a) Total time cost v.s. number of landing points. (b) Survival rate of sensors v.s. number of landing points.

Impact of energy requirement. We vary energy requirement of sensors from 5 Wh to 20 Wh. Fig. 6 shows that *DSA* reduces 76.31% of total time cost of *GRE* on average, and *DDSA* increases 25.92% of survival rate of sensors of *DGRE* on average. Note that the total time cost of *GRE* and *DSA* increase with energy requirement. This is because the more energy requirement and the more landing points passed through by the drone for charging sensors. In most cases, the time cost of paths computed by *DSA* and *DDSA* is less than that of paths computed by *GRE* and *DGRE*, respectively. Thus, drone scheduled by *DDSA* have more time to charge more sensors than drone scheduled by *DGRE*.

Impact of number of sensors. We vary number of sensors from 10 to 500. Fig. 7 shows that *DSA* reduces 81.84% of total time cost of *GRE* on average, and *DDSA* increases

Table 3
Schedule of bus lines in our experiments

Bus ID	Sub route	Route length (km)	Average speed (km/h)
B_{67}	$\{10 \rightarrow 11\}$	2.82	18.46
B_{44}	$\{21 \rightarrow 13 \rightarrow 22\}$	7.09	10.64
B_{41}	$\{13 \rightarrow 14\}$	3.39	16.78
B_{12}	$\{11 \rightarrow 15\}$	5.07	17.53
B_{35}	$\{9 \rightarrow 11\}$	4.28	14.28
B_{6}	$\{14 \rightarrow 16\}$	2.84	12.84
B_{17}	$\{8 \rightarrow 9\}$	2.85	22.44
B_{82}	$\{6 \rightarrow 8\}$	2.14	10.43
B_{52}	$\{17 \rightarrow 18\}$	8.12	9.74
B_{46}	$\{15 \rightarrow 19\}$	4.10	18.32
B_{68}	$\{2 \rightarrow 3\}$	3.42	5.13
B_{15}	$\{9 \rightarrow 20\}$	6.88	12.44
B_{8}	$\{6 \rightarrow 7\}$	7.81	9.37
B_{3}	$\{17 \rightarrow 18\}$	8.12	9.74
B_{25}	$\{18 \rightarrow 20\}$	12.10	14.50
B_{57}	$\{18 \rightarrow 19\}$	6.35	14.28
B_{45}	$\{13 \rightarrow 15\}$	2.3	16.78
B_{63}	$\{4 \rightarrow 5\}$	2.02	10.52
B_{49}	$\{12 \rightarrow 24\}$	6.96	13.48
B_{54}	$\{18 \rightarrow 19\}$	6.96	13.48
B_{36}	$\{1 \rightarrow 2\}$	2.9	15.82
B_{43}	$\{13 \rightarrow 15\}$	2.3	16.78
B_{38}	$\{17 \rightarrow 18\}$	8.12	9.74



(a) (b)

Figure 6: Impact of energy requirement: (a) Total time cost v.s. energy requirement. (b) Survival rate of sensors v.s. energy requirement.

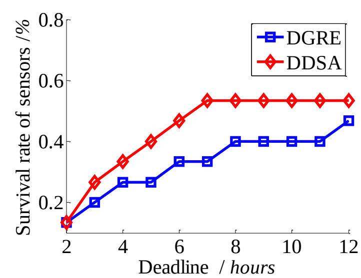
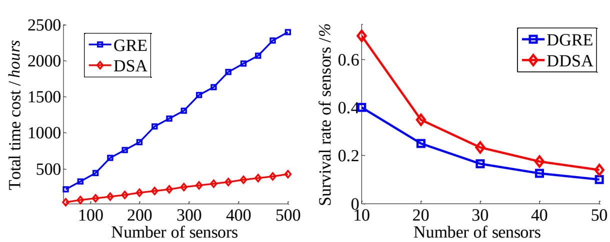


Figure 8: Survival rate of sensors v.s. deadline.



(a) (b)

Figure 7: Impact of number of sensors: (a) Total time cost v.s. number of sensors. (b) Survival rate of sensors v.s. number of sensors.

53.43% of survival rate of sensors of *DGRE* on average. Fig. 7 (a) and (b) show that total time cost for *DSA* and *GRE* increase and survival rate of sensors for *DDSA* and *DGRE* decrease with number of sensors, respectively. This is because the more sensors, the longer travel time of drone, and the fewer sensors that can be charged before the deadlines.

Impact of deadline. We vary deadlines of sensors from 2 *hours* to 12 *hours*. Fig. 8 shows that *DDSA* increases 33.33% of survival rate of sensors of *DGRE* on average. This is because *DGRE* can not find the feasible route for drone to charge 66.67% of sensors when their deadlines are less than 7 *hours* in the large-scale transportation network. However, drone passing through the feasible route computed by *DDSA* can charge at least 20% of these sensors.

Overall, *DSA* and *DDSA* can significantly decrease the total time cost and increase the survival rate of sensors through the designed algorithms, respectively.

Figure 9: Transportation network in Xianlin campus of NJUPT.

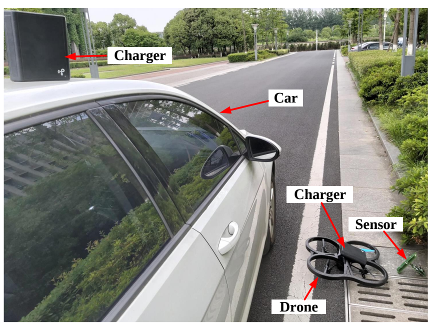


Figure 10: Testbed: a car, a rechargeable sensor, a drone and two chargers.

6. Field Experiment

In this subsection, we further evaluate the performance of *OPT, DSA, DOPT* and *DDSA* in the transportation network in Xianlin campus of NJUPT as shown in Fig. 9. Fig. 10 gives the test-bed, which consists of one drone carried one TX91501 power transmitter [39], three cars carried the TX91501 chargers as buses, 12 sensors deployed in Xianlin campus of NJUPT and 12 landing points.

First, Fig. 11 shows that the total time cost of *DSA* is 7.6 times that of *OPT* on average, and survival rate of sensors of *DDSA* is 80.00% of that of *DOPT* on average when number of landing points is from 7 to 10. Note that the performance gaps between *DSA* and *OPT*, *DDSA* and *DOPT* are small since *DSA* and *DDSA* have the guaranteed approximation. Then, Fig. 12 shows that the total time cost of *DSA* is 4.33 times of that of *OPT* on average, and survival rate of sensors of *DDSA* is 72.73% of that of *DOPT* on average when energy requirement is from 5 Wh to 20 Wh. Moreover, Fig. 13 shows that the total time cost of *DSA* is 6 times of that of *OPT* on average, and survival rate of sensors of *DDSA* is 88.54% of that of *DOPT* on average when number of sensors is from 5 to 12. Furthermore, Fig. 14 shows that survival rate of sensors of *DDSA* is 68.89% of that of *DOPT* on average when deadline is from 2 *hours* to 10 *hours*.

Running time. We vary number of sensors from 5 to 10. Fig. 15 shows that the running time of *OPT, DSA, DOPT* and *DDSA* grow linearly with the number of sensors.

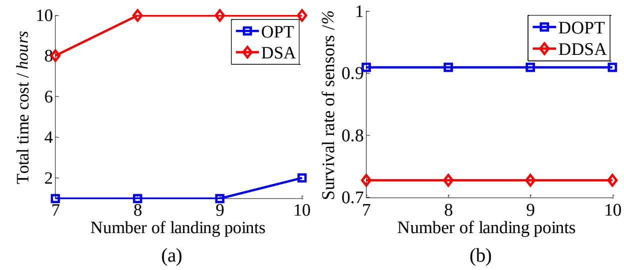


Figure 11: Impact of number of landing points: (a) Total time cost v.s. number of landing points. (b) Survival rate of sensors v.s. number of landing points.

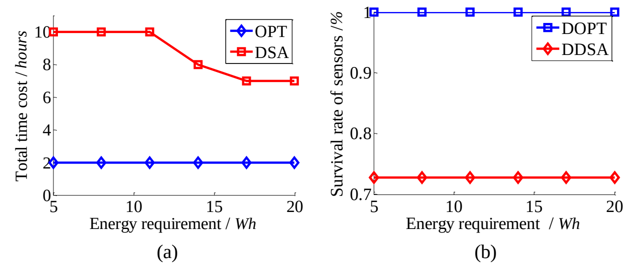


Figure 12: Impact of energy requirement: (a) Total time cost v.s. energy requirement. (b) Survival rate of sensors v.s. energy requirement.

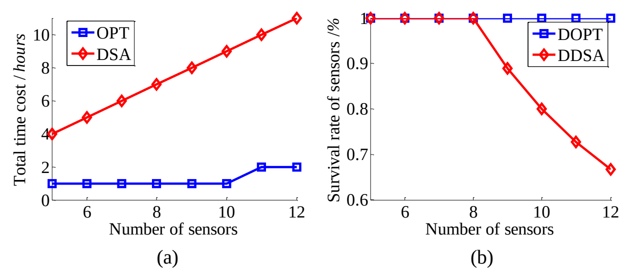


Figure 13: Impact of number of sensors: (a) Total time cost v.s. number of sensors. (b) Survival rate of sensors v.s. number of sensors.

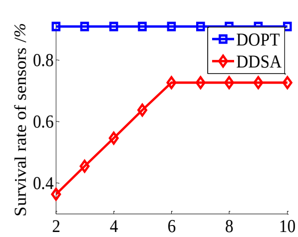


Figure 14: Survival rate of sensors v.s. deadline.

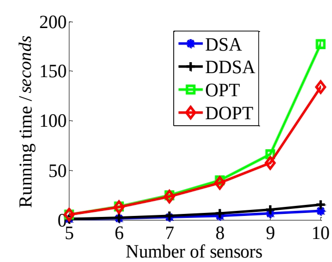


Figure 15: Running time v.s. number of sensors.

The running time of *OPT* and *DOPT* are 14.50 *seconds* and 14.04 *seconds* on average, respectively. Whereas, *DSA* and *DDSA* can complete the bus network assisted drone scheduling in 1.63 *seconds* and 2.45 *seconds* on average, respectively. Note that *DSA* and *DDSA* only use 11.23% and 17.42% of running time of *OPT* and *DOPT* on average, respectively.

So, the proposed algorithms are much more suitable to the large-scale transportation network.

7. Conclusion

In this article, we have designed the drone-enabled unique wireless charging system for sensors supported by the bus network in urban areas. The bus are sustainably charged by the *OLEV* system or its fuel engine, and has sufficient energy to charge the drone. The sensors are charged by the drone. We have formulated *DSB* problem to minimize the total time cost of drone subject to all sensors can be charged exactly once by the drone under the energy constraint of drone, and proposed an approximation algorithm to solve the energy tightened *DSB* problem. To consider the deadlines of sensors, we further formulate the *DDSB* problem to maximize the number of sensors charged by the drone under the constraints of both energy of drone and deadlines of sensors, and proposed an approximation algorithm to solve the energy tightened *DDSB* problem. The theoretical analysis, numerical simulations and field experiments have confirmed the efficiency and effectiveness of the proposed algorithms. The simulation results show that *DSA* can reduce the total time cost by 84.83% compared with *GRE* algorithm, and uses at most 5.98 times of the total time cost of optimal solution on average. Then, the results show that *DDSA* can increase the survival rate of sensors by 51.95% compared with *DGRE* algorithm, and can obtain 77.54% survival rate of optimal solution on average.

References

[1] L. Xie, Y. Shi, Y. Hou, and W. Lou. Wireless power transfer and applications to sensor networks. *IEEE Wireless Communications*, 20(4):140–145, 2013.

[2] S. Sudevalayam and P. Kulkarni. Energy harvesting sensor nodes: Survey and implications. *IEEE Communications Surveys & Tutorials*, 13(3):443–461, 2011.

[3] S. He, J. Chen, F. Jiang, D. Yau, G. Xing, and Y. Sun. Energy provisioning in wireless rechargeable sensor networks. In *Proc. IEEE INFOCOM*, pages 2006–2014, 2011.

[4] Mihai T. Lazarescu. Design of a wsn platform for long-term environmental monitoring for iot applications. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 3(1):45–54, 2013.

[5] J. PedroMuñoz-Gea, P. Manzanares-Lopez, J. Malgosa-Sanahuja, and J. Garcia-Haro. Design and implementation of a P2P communication infrastructure for WSN-based vehicular traffic control applications. *Journal of Systems Architecture*, 59(10):923–930, 2013.

[6] Z. Zhang, S. Glaser, R. Bales, Martha H. Conklin, R. Rice, and D. Marks. Insights into mountain precipitation and snowpack from a basin-scale wireless-sensor network. *Water Resources Research*, 53(8):6626–6641, 2017.

[7] A. Faustine, A. Mvuma, H. Mongi, M. Gabriel, and S. Kucel. Wireless sensor networks for water quality monitoring and control within lake victoria basin: Prototype development. *Wireless Sensor Network*, 6(12):281–290, 2014.

[8] X. Hu, B. Wang, and H. Ji. A wireless sensor network based structural health monitoring system for highway bridges. *Computer-Aided Civil and Infrastructure Engineering*, 28(3):193 – 209, 2013.

[9] M. Anisi, G. Abdul-Salaam, and A. Abdullah. A survey of wireless sensor network approaches and their energy consumption for monitoring farm fields in precision a griculture. *Precision Agriculture*, 16(2):216–238, 2015.

[10] M. Eiskamp, B. Griffin, J. Johnson, and E. Basha. *Unmanned Aerial Vehicle-Based Wireless Charging of Sensor Networks*. 2016.

[11] D. Zorbas and C. Douligeris. Computing optimal drone positions to wirelessly recharge iot devices. In *Proc. IEEE INFOCOM*, pages 628–633, 2018.

[12] C. Caillouet, T. Razafindralambo, and D. Zorbas. Optimal placement of drones for fast sensor energy replenishment using wireless power transfer. In *Proc. 2019 Wireless Days (WD)*, pages 1–6, 2019.

[13] A. Trotta, F. Andreagiovanni, M. Felice, E. Natalizio, and K. Chowdhury. When UAVs ride a bus: Towards energy-efficient city-scale video surveillance. In *Proc. IEEE INFOCOM*, pages 1043–1051, 2018.

[14] Y. Jin, J. Xu, S. Wu, L. Xu, and D. Yang. Enabling the wireless charging via bus network: Route scheduling for electric vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2020:1–13, 2020.

[15] S. Choi, B. Gu, S. Jeong, and C. Rim. Advances in wireless power transfer systems for roadway-powered electric vehicles. *IEEE J. Emerg. Sel. Topics Power Electron.*, 3(1):18–36, 2015.

[16] A. Somasundara, A. Ramamoorthy, and B. Srivastava. Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In *Proc. IEEE RTSS*, pages 1–10, 2004.

[17] R. Hassin. Approximation schemes for the restricted shortest path problem. *Math. Oper. Res.*, 17(1):36–42, 1992.

[18] X. Wang, H. Dai, H. Huang, Y. Liu, G. Chen, and W. Dou. Robust scheduling for wireless charger networks. In *Proc. IEEE INFOCOM*, 2019.

[19] T. Wu, P. Yang, H. Dai, W. Xu, and Xu M. Charging oriented sensor placement and flexible scheduling in rechargeable wsn. In *Proc. IEEE INFOCOM*, 2019.

[20] K. Adu-Manu, N. Adam, C. Tapparello, H. Ayatollahi, and W. Heinzelman. Energy-harvesting wireless sensor networks (EH-WSNs): A review. *ACM Transactions on Sensor Networks*, 14(2):1–50, 2018.

[21] P. Zhou, C. Wang, and Y. Yang. Self-sustainable sensor networks with multi-source energy harvesting and wireless charging. In *Proc. INFOCOM*, pages 1828–1836, 2019.

[22] T. Wu, P. Yang, P. Dai, P. Li, and X. Rao. Near optimal bounded route association for drone-enabled rechargeable WSNs. *Computer Networks*, 145:107–117, 2018.

[23] T. Chen, H. Wei, Y. Cheng, W. Shih, and H. Chen. An efficient routing algorithm to optimize the lifetime of sensor network using wireless charging vehicle. In *Proc. IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, pages 1–5, 2014.

[24] S. Chen, Y. Chang, T. Chen, Y Cheng, H. Wei, T. Hsu, and W. Shih. Prolong lifetime of dynamic sensor network by an intelligent wireless charging vehicle. In *Proc. IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pages 501–502, 2015.

[25] T. Angelo, F. Di, M. Federico, K. Chowdhury, and L. Bononi. Joint coverage, connectivity, and charging strategies for distributed uav networks. *IEEE Transactions on Robotics*, 34(4):883–900, 2018.

[26] T. Long, M. Ozger, O. Cetinkaya, and O. Akan. Energy neutral internet of drones. *IEEE Communications Magazine*, 56(1):22–28, 2018.

[27] J. Zhou, B. Zhang, W. Xiao, D. Qiu, and Y. Chen. Nonlinear paritytime-symmetric model for constant efficiency wireless power transfer: Application to a drone-in-flight wireless charging platform. *IEEE Transactions on Industrial Electronics*, 66(5):4097–4107, 2019.

[28] H. Wang, Q. Li, Z. Lin, C. Cai, W. Wang, and T. Meng. Optimization design of drone wireless charging system based on asymmetric coupling. In *Proc. the International Conference on Artificial Intelligence and Computer Science*, pages 258–264, 2019.

[29] V. Hassij, V. Saxen, and V. Chamol. Scheduling drone charging for multi-drone network based on consensus time-stamp and game theory. *Computer Communications*, 149:51–61, 2020.

[30] F. Lam and A. Newman. Traveling salesman path problems. *Mathematical Programming*, 113:39–59, 2008.

[31] R. Garey and S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. 1979.

[32] J. Yen. An algorithm for finding shortest routes from all source nodes to a given destination in general networks. *Quarterly of Applied Mathematics.*, 27(4):526–530, 1970.

[33] N. Bansal, A. Blum, S. Chawla, and A. Meyerson. Approximation algorithms for deadline-TSP and vehicle routing with time-windows. In *Proc. STOC*, pages 166–174, 2004.

[34] C. Chekuri and M. Pál. A recursive greedy algorithm for walks in directed graphs. In *Proc. 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 1–9, 2005.

[35] *New York City Bus Data*, 2020. https://www.kaggle.com/stoney71/new-york-city-transport-tatistics/.

[36] *Inspire 2*, 2020. https://www.dji.com/cn.

[37] *Aliyun ECS Cloud Server*, 2020. https://www.aliyun.com/product/ecs.

[38] H. Cormen, E. Leiserson, L. Rivest, and C. Stein. *Introduction To Algorithms*. 2001.

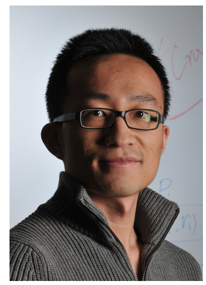
[39] *Powercast*, 2020. http: //www.powercastco.com/products/.

Yong Jin received the M.S. degree from Nanjing Tech University, Nanjing, China, in 2009. He is currently pursuing the Ph.D. degree with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, China. He is an associate professor in the School of Computer Science & Engineering, Changshu Institute of Technology, Changshu, China. His current research interests include wireless charging, intelligent transportation system, 5G, etc.

Jia Xu (M'15) received the M.S. degree in School of Information and Engineering from Yangzhou University, Jiangsu, China, in 2006 and the PhD. Degree in School of Computer Science and Engineering from Nanjing University of Science and Technology, Jiangsu, China, in 2010. He is currently a professor in the School of Computer Science at Nanjing University of Posts and Telecommunications. He was a visiting Scholar in the Department of Electrical Engineering & Computer Science at Colorado School of Mines from Nov. 2014 to May. 2015. His main research interests include crowdsourcing, edge computing and wireless sensor networks. Prof. Xu has served as the PC Co-Chair of SciSec 2019, Organizing Chair of ISKE 2017, TPC member of Globecom, ICC, MASS, ICNC, EDGE. He currently serves as the Publicity Co-Chair of SciSec 2021.

Sixu Wu received the bachelor's degree in School of Computer Science from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2019. He is currently working toward the master's degree in the same university. His research interests are mainly in the areas of the mobile crowd sensing, and wireless charger network.

Lijie Xu received his Ph.D. degree in the Department of Computer Science and Technology from Nanjing University, Nanjing, in 2014. He was a research assistant in the Department of Computing at the Hong Kong Polytechnic University, Hong Kong, from 2011 to 2012. He is currently an associate professor in the School of Computer Science at Nanjing University of Posts and Telecommunications, Nanjing. His research interests are mainly in the areas of wireless sensor networks, ad-hoc networks, mobile and distributed computing, and graph theory algorithms.



Dejun Yang (SM'19) received the B.S. degree in Computer Science from Peking University, Beijing, China, in 2007 and PhD degree in Computer Science from Arizona State University, Tempe, AZ, USA, in 2013. He is currently the Ben L. Fryrear Assistant Professor of Computer Science in the Department of Electrical Engineering & Computer Science at Colorado School of Mines. His main research interests include economic and optimization approaches to networks, crowdsourcing, smart grid, big data, and cloud computing. Prof. Yang has served as a Technical Program Committee Memberfor many conferences, including the IEEE International Conference on Computer Communications (INFOCOM), the IEEE International Conference on Communications (ICC), and the IEEE Global Communications Conference (GLOBECOM). He has received Best Paper Awards at the IEEE GLOBECOM (2015), the IEEE International Conference on Mobile Ad hoc and Sensor Sys-tems (2011), and the IEEE ICC (2011 and 2012), as well as a Best Paper Award Runner-up at the IEEE International Conference on Network Protocols (2010).

Kaijian Xia (M'16), received the M.S. degree in School of Information and Engineering from Jiangnan University, Wuxi, Jiangsu, China, in 2009 and the PhD. degree at School of Information and Control Engineering from China University of Mining and Technology, Xuzhou, Jiangsu, China, in 2020. Currently, he is a senior engineer of computer science with the affiliated Changshu Hospital of Soochow University, Changshu, Jiangsu, China, and an associate professor of medical information with Xuzhou Medical University. His research interests include intelligent medical information, deep learning, biomedical image analysis, bio-inspired computing, pattern recognition and transfer learning. He has served as the TPC Vice-Chair for CyberLife 2019 and currently serves the Editor in Chief for International Journal of Health Systems and Translational Medicine, an Associate Editor for Journal of Medical Imaging and Health Informatics.