Minimizing the Deployment Cost of UAVs for Delay-Sensitive Data Collection in IoT Networks

Wenzheng Xu, *Member, IEEE*, Tao Xiao, Junqi Zhang, Weifa Liang, *Senior Member, IEEE*, Zichuan Xu, *Member, IEEE*, Xuxun Liu, Xiaohua Jia, *Fellow, IEEE*, and Sajal K. Das, *Fellow, IEEE*

***Abstract*—In this paper, we study the deployment of Unmanned Aerial Vehicles (UAVs) to collect data from IoT devices, by finding a data collection tour for each UAV. To ensure the 'freshness' of the collected data, the total time spent in the tour of each UAV that consists of the UAV flying time and data collection time must be no greater than a given delay B, e.g., 20 minutes. In this paper, we consider a problem of deploying the minimum number of UAVs and finding their data collection tours, subject to the constraint that the total time spent in each tour of any UAV is no greater than B. Specifically, we study two variants of the problem: one is that a UAV needs to fly to the location of each IoT device to collect its data; the other is that a UAV is able to collect the data of an IoT device if the Euclidean distance between them is no greater than the wireless transmission range of the IoT device. For the first variant of the problem, we propose a novel 4-approximation algorithm, which improves the best approximation ratio $4\frac{4}{7}$ for it so far. For the second variant, we devise the very first constant factor approximation algorithm. We also evaluate the performance of the proposed algorithms via extensive experiment simulations. Experimental results show that the numbers of UAVs deployed by the proposed algorithms are from 11% to 19% less than those by existing algorithms on average.**

***Index Terms*—Mobile data collection, multiple UAV scheduling, minimum numbers of UAV deployments, minimum cycle cover with neighborhoods, approximation algorithms.**

Manuscript received April 2, 2021; revised September 23, 2021; accepted October 24, 2021; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor T. He. Date of publication November 5, 2021; date of current version April 18, 2022. The work of Wenzheng Xu was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61602330 and in part by the Sichuan Science and Technology Program under Grant 2018GZDZX0010 and Grant 2017GZDZX0003. The work of Weifa Liang was supported by the City University of Hong Kong under Project CityU 9380137/CS. The work of Xiaohua Jia was supported by the Research Grants Council of Hong Kong under Project CityU 11214316. The work of Sajal K. Das was supported in part by NSF Grants under Award CCF-1725755, Award CNS-1818942, Award SCC-1952045, and Award SaTC-2030624. *(Corresponding author: Zichuan Xu.)*

Wenzheng Xu, Tao Xiao, and Junqi Zhang are with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: wenzheng.xu3@gmail.com; xt980124@163.com; junqizhangscu@163.com).

Weifa Liang and Xiaohua Jia are with the Department of Computer Science, City University of Hong Kong, Hong Kong, China (e-mail: weifa.liang@cityu.edu.hk; csjia@cityu.edu.hk).

Zichuan Xu is with the School of Software, Dalian University of Technology, Dalian 116024, China (e-mail: z.xu@dlut.edu.cn).

Xuxun Liu is with the College of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: liuxuxun@scut.edu.cn).

Sajal K. Das is with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: sdas@mst.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNET.2021.3123606, provided by the authors.

Digital Object Identifier 10.1109/TNET.2021.3123606

I. INTRODUCTION

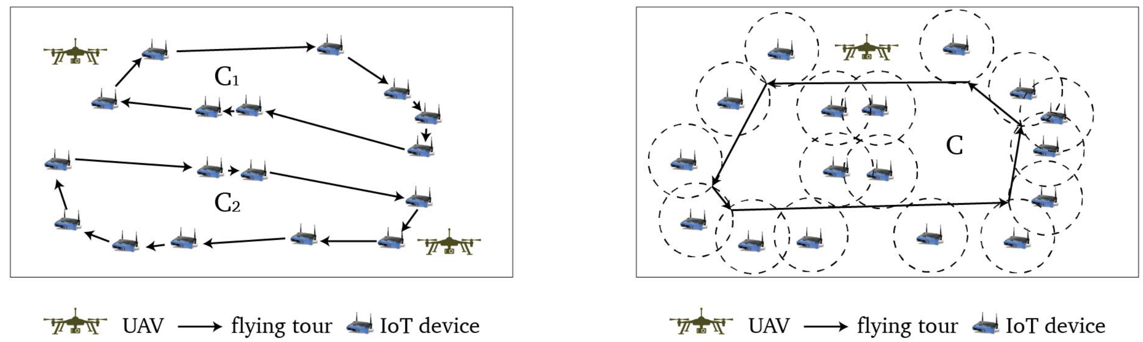
DUE to their flexibility and cost-efficiency, Unmanned Aerial Vehicles (UAVs) now are widely used in many applications including goods delivery, target tracking, emergency aid, charging wireless sensor networks [14], [20], [22], [27], [28], [31]–[33], [35], [40], [41], [43], and so on. On the other hand, millions of Internet of Thing (IoT) devices, such as various sensors and smart monitoring devices, have been deployed in many IoT networks in the past years for various applications.

In this paper, we study the data collection of IoT devices in a large-scale IoT network, e.g., ten square kilometers, where IoT devices are only sparsely deployed at some strategic locations to monitor important Points of Interest (PoIs) in the network. Due to the large scale of the network and limited energy supplies of IoT devices, sometimes it is unrealistic to allow the IoT devices to directly transmit or relay their sensing data to a base station via multihop relays.

We consider the deployment of multiple light-weight UAVs to collect data from IoT devices, where a UAV can fly to a location nearby an IoT device to collect its data, thereby saving the energy consumption of the IoT device. Fig. 1(a) shows that two UAVs are deployed to collect data of IoT devices along their data collection flying tours, respectively.

To ensure the 'freshness' of the collected data, a strict requirement is that the total time spent in the tour of each UAV, which consists of the UAV flying time and data collection time, should be no greater than a given delay B, e.g., 20 minutes [4]. Otherwise, the collected data is somewhat 'stale'. For example, consider networks in which IoT devices are deployed to monitor bushfires in a forest [18] or PM 2.5 pollution in a city [34], it is important to collect sensing data as timely as possible.

In this paper, we study a novel *minimum UAV deployment problem*, which is to determine the minimum number of UAVs to-be-deployed and find their the data collection tours, such that the data of each IoT device is collected by one of the UAVs, subject to that the total time spent by any UAV in its tour is no greater than a given delay B. Specifically, we consider two variants of the minimum UAV deployment problem: One is termed as the minimum UAV deployment problem *without neighborhoods*, in which a UAV needs to fly to the location of each IoT device to collect its data, see Fig. 1(a). In this case, the wireless transmission range of the IoT device is much shorter than the scale of the network, or the device cannot transmit data in a wireless way. One application example is that each IoT device is an RFID



(a) Data collection without neighborhoods by deploying two UAVs and their data collection tours

(b) Data collection with neighborhoods by deploying only one UAV and its data collection tour

Fig. 1. Data collection by UAVs in two scenarios.

tag. To collect the data of the tag, a UAV must be equipped with an RFID reader and the reader can read the data of the tag only when their distance is very short, e.g., a few meters [42]. Other applications include the scheduling of UAVs to monitor survivals in a disaster area [14], [32], and the deployment of UAVs to monitor traffic jams on congested cross roads in a smart city, where no IoT devices are deployed at all in such application scenarios.

The other is referred to as the minimum UAV deployment problem *with neighborhoods*, an IoT device is able to transmit its data to a UAV through wireless data transmission and the data then can be received by the UAV when the Euclidean distance between the device and the UAV is no greater than a given communication range, see Fig. 1(b). Unlike the wireless communication between two devices on the ground that wireless signals degrade very quickly due to various shadowing and scattering, the radio signals from a ground device to a UAV in the air, or vice visa, degrade much slower, due to less obstacles between them [1], [5]. Therefore, the wireless communication range between a ground IoT device and a UAV usually is much longer than the communication range between two ground IoT devices, e.g., 500 m vs. 50 m [1], [2], [5]. It then can be seen that by exploiting the long communication range between an IoT device and a UAV, the number of deployed UAVs may be significantly reduced. For example, Fig. 1(a) shows that two UAVs are deployed when a UAV must fly to the location of each IoT device to collect its data, while Fig. 1(b) demonstrates that only one UAV is deployed when taking the communication range into consideration.

The novelties of this paper are as follows. We propose a 4-approximation algorithm for the minimum UAV deployment problem without neighborhoods, improving the best approximation ratio $4\frac{4}{7}$ so far [37]. The techniques adopted in the algorithm are also different from existing ones including the one in [37]. Specifically, the algorithm in [37] first obtains some connected components by removing edges greater than a given edge weight threshold, then merges the minimum spanning trees of the connected components, and decomposes the merged trees into subtrees such that the weight of each decomposed subtree is no greater than $B/2$, finally obtains closed tours by doubling edges in the decomposed subtrees. In contrast, the proposed 4-approximation algorithm first obtains some connected components by removing edges greater than some edge weight threshold and the optimal value of the edge weight threshold needs to be searched, then finds minimum spanning trees of the connected components, respectively, and obtains closed tours by adding edges in the minimum weighted perfect matching of odd degree nodes in the trees, finally splits each tour into subtours such that the length of each subtour is no greater than B. We formally estimate non-trivial upper bounds on the total weight of the minimum spanning trees and the weight of the minimum weighted perfect matching in Sections IV-C.1 and IV-C.2, respectively. In addition, we devise the very first constant approximation algorithm for the minimum UAV deployment problem with neighborhoods. Experimental results show that much less numbers of UAVs are deployed when taking the wireless communication range (i.e., neighborhoods) of IoT devices into consideration.

The main contributions of this paper are summarized as follows. **(i)** We study a novel minimum UAV deployment problem, which is to determine the minimum number of UAVs to-be-deployed and find the data collection tour for each of the UAVs, subject to that the total time spent by each UAV per tour is no greater than a given delay B. **(ii)** For the first variant of the problem – the minimum UAV deployment problem without neighborhoods where a UAV needs to fly to the location of each IoT device for its data collection, we propose a 4-approximation algorithm, which improves the best approximation ratio $4\frac{4}{7}$ for the problem so far. **(iii)** For the second variant of the problem – the minimum UAV deployment problem with neighborhoods where a UAV can collect the data of an IoT device as long as their Euclidean distance is no greater than a given communication range, we devise the first constant factor approximation algorithm for it. **(iv)** Experimental results show that the numbers of UAVs deployed by the proposed algorithms are around from 11% to 19% less than those by existing algorithms on average.

The rest of the paper is organized as follows. Section II reviews related work. Section III introduces the network and data collection models, and defines the problems. Sections IV and V propose approximation algorithms for the minimum UAV deployment problem with and without neighborhoods, respectively. Section VI evaluates the performance of the proposed algorithms. Finally, Section VII concludes the paper.

II. Related Work

Some existing studies considered a scenario of dispatching a mobile sink to collect data from sensors in an IoT network. For example, Xu *et al.* [29] studied the problem of dispatching a mobile sink to collect data from sensors such that the lifetime of the network is maximized. Ren *et al.* [26] investigated the problem of using a mobile sink to collect data in a renewable sensor network deployed on a roadside, such that the amount of data collected from all sensors is maximized. On the other hand, there are also some studies focusing on deploying multiple mobile sinks to collect data from sensors. For example, Konstantopoulos *et al.* [17] studied the problem of dispatching multiple mobile sinks to collect sensor data to maximize the data throughput, while ensuring network connectivity and balancing energy consumption among sensors. They proposed algorithms to cluster sensors in a network, find cluster heads, and determine the rendezvous nodes, followed by dispatching mobile sinks to collect data from rendezvous nodes. However, due to various obstacles in the ground such as rocks, rivers and buildings, mobile sinks cannot move freely and they may not be able to reach to the locations of some sensors.

There are several recent studies on the deployment of UAVs for data collection in an IoT network. For example, Zhan *et al.* [38] studied the problem of dispatching a UAV to collect data from sensors so as to minimize the maximum energy consumption among sensors, while ensuring that sensor data are reliably collected. Ebrahimi *et al.* [12] considered the problem of clustering densely-located sensors, constructing a data collection tree for each cluster, and finding a flying trajectory for a UAV to gather data from cluster heads, so that the UAV flying distance is minimized. Liang *et al.* [20] studied a problem of finding an optimized tour for a UAV such that the quality of photos taken during the tour is maximized, subject to energy capacity on a UAV. They proposed a novel approximation algorithm and a fast yet scalable heuristic algorithm for the problem. Zhan *et al.* [39] studied the problem of dispatching a UAV which starts from and ends at a given location so that the number of sensors with their data collected by the UAV within a given duration is maximized. Li *et al.* [19] considered the problem of deploying an energy constrained UAV to collect data in an IoT network in different data collection models: one is that the hovering coverage of different sensors do not overlap with each other, the other is that there are coverage overlapping. They also considered a partial data collection maximization problem. Unlike those mentioned studies, You and Zhang [35] considered a scenario that a UAV can change its altitude to collect data from different sensors, and studied the problem of deploying a UAV to collect sensor data, such that the minimum average data collection rate from all sensors is maximized, under a prescribed reliability constraint for each sensor. Unlike existing studies, in this paper we focus on dispatching the minimum number of UAVs for data collection in an IoT network, subject to that the maximum time spent by each UAV per tour is no greater than a given delay B.

We also note that there are several studies on the minimum cycle cover problem without neighborhoods, which is to find the minimum number of cycles to cover all nodes in a graph, such that the length of each cycle is no more than a given bound B, which are closely related to the work in this paper. For example, Arkin *et al.* [3] proposed the very first 6-approximation algorithm for the problem. Khani and Salavatipour [15] then devised a 5-approximation algorithm. Yu *et al.* [37] recently further improved the result by proposing two approximation algorithms for the problem: one with approximation ratio $4\frac{2}{3}$ and time complexity $O(n^3)$; the other with approximation ratio $4\frac{4}{7}$ and time complexity $O(n^5)$, respectively, where $4\frac{4}{7} < 4\frac{2}{3}$. In contrast, the proposed algorithm in this paper can deliver a 4-approximate solution in time $O(n^4)$. On the other hand, for the single-rooted minimum cycle cover problem with each cycle must contain a root node, Nagarajan and Ravi [23] proposed an $O(\log B)$-approximation algorithm, where B is the length constraint of each tour. Dai *et al.* [7] studied the problem of deploying the minimum number of charging vehicles to fully charge a set of energy-critical sensors, by utilizing the approximation algorithm in [23]. However, the cost of each obtained tour by the algorithms in [7] and [23] may exceed the length bound. In contrast, Zhang *et al.* [43] and Liang *et al.* [21] proposed approximation algorithms for the single-rooted minimum cycle cover problem, such that the length of each obtained tour is no greater than the length bound B.

The minimum UAV deployment problem without neighborhoods considered in this paper is closely related to the min-max cycle cover problem that is to find K closed tours to visit nodes in a graph such that the length of the longest tour among the found K tours is minimized, where K is a given positive integer. For the min-max cycle cover problem, Arkin *et al.* [3] proposed the first constant approximation algorithm with an approximation ratio of 8, and Khani and Salavatipour [15] later improved the approximation ratio to 6. Xu *et al.* [30] further improved the result by devising a $5\frac{1}{3}$-approximation algorithm. Yu and Liu [36] further reduced the approximation ratio to 5. When K is a small constant (e.g., $K = 5$), Guo *et al.* [14] proposed an improved $4\frac{1}{3}$-approximation algorithm to minimize the longest tour time among K UAVs for disaster area surveillance. It can be seen that, given an approximation algorithm for the min-max cycle cover problem, we can find the minimum number of UAVs deployed by invoking the algorithm multiple times, through increasing the value of K from 1 to n until the longest tour length is no greater than B, where n is the number of IoT devices. However, this method does not deliver a constant approximate solution to the minimum UAV deployment problem.

The study in this paper is also closely related to the studies on the Traveling Salesman Problem with Neighborhoods (TSPN) in a 2D Euclidean space that is to find a single shortest tour such that at least one location in each disk is visited. When the radii of all disks are identical, Dumitrescu and Mitchell [9] first proposed a 7.62-approximation algorithm and recently improved the ratio to 6.75 [10]. On the other hand, when the radii of different disks are different, Dumitrescu and Tóth [11] devised a constant factor approximation algorithm. In addition, Deng *et al.* [8] recently studied the problem of finding K closed tours to visit all disks in a 2D space, such that the length among the K found tours is minimized, for which they proposed approximation algorithms. Furthermore, Elbassioni *et al.* [13] devised a constant approximation algorithm for the TSPN problem with intersecting fat convex regions and the ratio of the largest radius of all regions to the smallest radius is upper bounded by a constant.

III. Preliminaries

A. Network Model

We consider an IoT application scenario where many IoT devices are deployed in an area to monitor important Points of Interest (PoIs) in the area. For example, IoT devices are used to monitor PM 2.5 pollution in a smart city [34] or monitor bushfires in a forest [18]. Assume that there are n devices $v_1, v_2, \ldots, v_n$ deployed at some strategic locations in the monitoring area, where n is a positive integer. Let V be the set of IoT devices, i.e., $V = \{v_1, v_2, \ldots, v_n\}$. Denote by $(x_i, y_i, 0)$ the coordinates of device v_i with $1 \leq i \leq n$. We assume that the coordinates of each device v_i are given, which can be obtained when the device is deployed.

Notice that a monitoring area may be very large. For example, in the case an IoT network is deployed for monitoring forest fires, the monitored forest area may be tens of square kilometers [18], and the IoT devices usually are sparsely deployed. To form a connected IoT network, a traditional solution is to deploy relay devices among the deployed devices. However, since the transmission range between two devices on the ground usually is only dozens of meters, a large number of relaying devices need to deploy in order to form a connected network, thereby incurring high deployment cost. In contrast, in this paper we consider the deployment of UAVs to collect data from IoT devices, because UAVs are flexible and no relaying devices need to be deployed, thus saving the deployment cost.

B. Data Collection Models

We consider two data collection scenarios. One is referred to as *the data collection without neighborhoods*. That is, a UAV needs to fly to the location of each IoT device to collect its data. One application example of this model is that each device is an RFID tag. To collect the data of the tag, a UAV must be equipped with an RFID reader to read the data of the tag when the Euclidean distance between them is very short, e.g., a few meters. The other is referred to as *the data collection with neighborhoods*. That is, each IoT device can send its data to a UAV wirelessly, and the UAV can collect the data from the device as long as their Euclidean distance is no greater than a given wireless transmission range. In the following, we introduce the data collection models for these two scenarios accordingly.

Assume that K UAVs are deployed to collect data from n IoT devices, where the value of K is unknown and will be determined later. The set V of the n IoT devices is partitioned into K disjoint subsets $V_1, V_2, \ldots, V_K$, where UAV k collects the data from the devices in V_k with $1 \leq k \leq K$. Let $V_k = \{v_1, v_2, \ldots, v_{n_k}\}$, where $n_k = |V_k|$. Denote by Δ_i the amount of data of device v_i in V to be collected. Assume that each UAV can collect data from v_i at a data rate b. Then, it takes $\rho(v_i) = \frac{\Delta_i}{b}$ time to collect all data of v_i. Also, denote by η the flying speed of each UAV, assuming that each UAV is equipped with a GPS module [25].

1) Data Collection Model in the Scenario Without Neighborhoods: We first introduce the data collection model in a scenario without neighborhoods, where a UAV must fly to the location of each device to collect its data. Assume that UAV k collects the data from devices in V_k in the order of $v_1, v_2, \ldots, v_{n_k}$, where $1 \leq k \leq K$. The data collection flying tour C_k of UAV k is defined as follows. UAV k first collects the data of device v_1 at the location of v_1, it then flies to collect the data of device v_2, and so on. After finishing the data collection from device v_{n_k}, UAV k returns to the location of v_1. That is, the data collection trajectory of UAV k is a *closed tour* $C_k = v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_{n_k} \rightarrow v_1$ with $1 \leq k \leq K$. The data collection trajectory of each UAV is a closed tour, since the UAV usually needs to periodically collect data from the deployed IoT devices, rather than only once. After collecting the data from each IoT device v_i, UAV k can forward the data to a base station immediately via 4G/5G communications. We assume that the base station is located at a place nearby the disaster area. Fig. 1(a) shows that 17 IoT devices are deployed in a monitoring area and two UAVs are dispatched to collect the data of the devices.

The total time spent by UAV k in its flying tour C_k consists of its flying time between IoT devices and the data collection time of devices in V_k. Denote by $t_f(v_i, v_{i+1})$ the flying time of UAV k between devices v_i and v_{i+1}, i.e., $t_f(v_i, v_{i+1}) = \frac{d(v_i, v_{i+1})}{\eta}$, where $d(v_i, v_{i+1})$ is the Euclidean distance between devices v_i and v_{i+1}, and η is the flying speed of UAV k. The flying time of UAV k in tour C_k then is $\sum_{i=1}^{n_k} t_f(v_i, v_{i+1})$, where $v_{n_k+1} = v_1$.

The total time $w(C_k)$ spent by UAV k in its tour C_k under the scenario without neighborhoods then is

$$w(C_k) = \sum_{i=1}^{n_k} t_f(v_i, v_{i+1}) + \sum_{i=1}^{n_k} \rho(v_i), \tag{1}$$

where $\rho(v_i)$ is the data collection time of device v_i.

It is important to collect data from devices in V_k as quickly as possible. Otherwise, the collected data will become 'stale', and lose its value. Denote by B a given delay, e.g., 20 minutes [4]. Then, the total spent time $w(C_k)$ of UAV k in its flying tour C_k for any k must be no greater than B, i.e., $w(C_k) \leq B$ with $1 \leq k \leq K$.

2) Data Collection Model in the Scenario With Neighborhoods: We then introduce the data collection model with neighborhoods, where a UAV can collect the data of each IoT device when their Euclidean distance is no greater than a given communication range R, e.g., $R = 600$ m [1], [5].

Assume that all UAVs fly at the same altitude h such that the coverage range of each UAV is maximized, where $h < R$ and the optimal altitude h can be obtained from the work in [1], e.g., $h = 300$ meters. Specifically, the air-to-ground wireless signal propagation consists of two main propagation components: the Line-of-Sight (LoS) propagation, and the non-LoS propagation with strong reflections and diffractions. It can be seen that, the higher the altitude is, the larger the LoS propagation loss is, but the smaller the non-LoS propagation loss is, due to higher LoS probability. Then, there is an optimal altitude h for the maximum coverage from the sky.

Denote by $D(v_i)$ the set of locations that a UAV can collect data from a device v_i at altitude h, i.e., $D(v_i) = \{(x, y, h) \mid (x - x_i)^2 + (y - y_i)^2 + (h - 0)^2 \le R^2\}$, where $(x_i, y_i, 0)$ are the coordinates of device v_i. It can be seen that $D(v_i)$ is a disk centered at point (x_i, y_i, h) with a radius $R_0 = \sqrt{R^2 - h^2}$ at altitude h.

Recall that we assumed that UAV k collects the data of devices in set V_k in the order of $v_1, v_2, \ldots, v_{n_k}$, where $V_k = \{v_1, v_2, \ldots, v_{n_k}\}$, $n_k = |V_k|$, and $1 \le k \le K$. The data collection flying tour C_k of UAV k in the scenario with neighborhoods is defined as follows. UAV k first collects data of device v_1 at a location p_1 in $D(v_1)$, it then flies to a location p_2 in $D(v_2)$ and collect the data of device v_2, and so on. After having collected the data from device v_{n_k} at a location p_{n_k} in $D(v_{n_k})$, the UAV finally returns to the starting location p_1. The flying tour C_k of UAV k can be represented as $C_k = p_1 \to p_2 \to \cdots \to p_{n_k} \to p_1$, where p_i is a location in the neighborhood $D(v_i)$ of device v_i with $1 \le i \le n_k$, and $1 \le k \le K$. Fig. 1(b) shows the flying tour of a UAV, where dotted circles represent the neighborhoods of devices. Similar to Eq. (1), the total time $w_N(C_k)$ spent by UAV k in its flying tour C_k is

$$w_N(C_k) = \sum_{i=1}^{n_k} t_f(p_i, p_{i+1}) + \sum_{i=1}^{n_k} \rho(v_i), \tag{2}$$

where $t_f(p_i, p_{i+1})$ is the UAV flying time between locations p_i and p_{i+1}, p_i and p_{i+1} are located in the neighborhoods $D(v_i)$ and $D(v_{i+1})$, respectively, and $\rho(v_i)$ is the data collection time of device v_i. Notice that the total spent time $w_N(C_k)$ of UAV k in its flying tour C_k must be no greater than the given delay B, i.e., $w_N(C_k) \le B$ with $1 \le k \le K$.

C. Problem Definitions

In this paper, we study a novel *minimum UAV deployment problem*, which is to minimize the number of deployed UAVs to collect data from all devices, subject to the constraint that the total time spent by each UAV in its tour is no greater than a given delay B. Specifically, we consider the problem under two different data collection models: the data collection with and without neighborhoods.

We first formulate the problem under the data collection model without neighborhoods. Given an IoT network $G = (V, E; t_f : E \mapsto \mathbb{R}^{\ge 0}, \rho : V \mapsto \mathbb{R}^{\ge 0})$ and a maximum data collection delay B, a UAV needs to fly to the location of each IoT device to collect its data. *The minimum UAV deployment problem without neighborhoods* in G is to determine the minimum number K of UAVs to be deployed, and to find the flying tours $C_1, C_2, \ldots, C_K$ for the K UAVs to collaboratively collect data from all devices in V, subject to the constraint that the total time $w(C_k)$ spent in each tour C_k with $1 \le k \le K$ is no greater than B. The minimum UAV deployment problem without neighborhoods is NP-hard [36].

The other variant of the problem under the data collection model with neighborhoods can be formulated similarly. Specifically, given an IoT network $G = (V, E; t_f : E \mapsto \mathbb{R}^{\ge 0}, \rho : V \mapsto \mathbb{R}^{\ge 0})$, a maximum data collection delay B, and a disk $D(v_i)$ of each device v_i which centers at the location of v_i with a radius R_0 at altitude h, the data of device v_i can be collected by a UAV when the UAV hovers at any location in $D(v_i)$. *The minimum UAV deployment problem with neighborhoods* in G is to determine the minimum number K of deployed UAVs and to find the flying tours $C_1, C_2, \ldots, C_K$ of the K UAVs to collect the data from all devices, subject to the constraint that the total time spent $w_N(C_k)$ by UAV k in its tour C_k is no greater than B for any k with $1 \le k \le K$.

Lemma 1: The minimum UAV deployment problem with neighborhoods is NP-hard.

Proof: The proof is given in Section 1 of the supplementary file. □

IV. Approximation Algorithm for the Minimum UAV Deployment Problem Without Neighborhoods

In this section, we deal with the minimum UAV deployment problem without neighborhoods, by proposing a 4-approximation algorithm for it.

A. Algorithm Framework

Given an IoT network $G = (V, E; t_f : E \mapsto \mathbb{R}^{\ge 0}, \rho : V \mapsto \mathbb{R}^{\ge 0})$, an auxiliary complete graph $G' = (V, E; w' : E \mapsto \mathbb{R}^{\ge 0})$ is constructed from G, and the weight of each edge (v_i, v_j) in G' is $w'(v_i, v_j) = t_f(v_i, v_j) + \frac{\rho(v_i) + \rho(v_j)}{2}$, where $t_f(v_i, v_j)$ is the flying time between devices v_i and v_j, $\rho(v_i)$ and $\rho(v_j)$ are the durations for collecting data from v_i and v_j, respectively. Following a similar analysis in the work [43], the optimal solutions to the minimum UAV deployment problem in G and G' are equal. Notice that the original graph G is both edge-weighted and node-weighted, while graph G' is only edge-weighted.

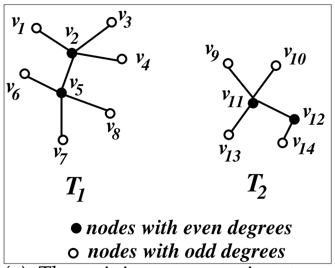
Let $\delta_i = \frac{B}{i}$ with $2 \le i \le n$, where δ_i is referred to as an *edge weight threshold*. The basic idea of the proposed algorithm is that, the algorithm finds a set $\mathcal{C}_i$ of tours that visit nodes in G' based on a given edge weight threshold δ_i, subject to the constraint that the cost of each found tour in $\mathcal{C}_i$ is no greater than B. The final solution $\mathcal{C}$ to the problem then is the set with the minimum number of tours, i.e., $|\mathcal{C}| = \min_{2 \le i \le n}\{|\mathcal{C}_i|\}$. The approximation algorithm for the minimum UAV deployment problem without neighborhoods is presented in `Algorithm 1`.

Algorithm 1 Algorithm for the Minimum UAV Deployment Problem Without Neighborhoods (**approAlgNoNei**)

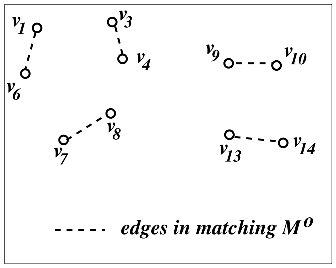
Input: an IoT network $G = (V, E; t_f : E \mapsto \mathbb{R}^{\ge 0}, \rho : V \mapsto \mathbb{R}^{\ge 0})$, and a maximum data collection delay B

Output: a set $\mathcal{C}$ of tours to visit all devices in V, such that the total spent time of each tour in $\mathcal{C}$ by a UAV is no greater than B.

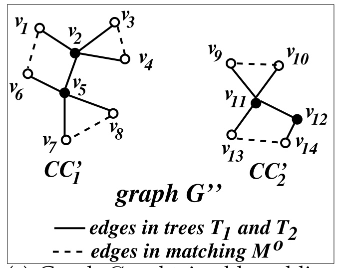
Construct a graph $G' = (V, E; w' : E \mapsto \mathbb{R}^{\ge 0})$ from G;
Construct a trivial solution $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$, where each tour C_i consists of only a single node v_i in V, and $n = |V|$;
for $i \leftarrow 2$ to n **do**
  Let $\delta_i \leftarrow \frac{B}{i}$; /* set an edge weight threshold */
  Find a set $\mathcal{C}_i$ of tours to visit all nodes in G' based on the edge weight threshold δ_i, subject to that the cost of each tour in $\mathcal{C}_i$ is no greater than B, by invoking `Algorithm 2`;
  if $|\mathcal{C}_i| < |\mathcal{C}|$ **then**
    Let $\mathcal{C} \leftarrow \mathcal{C}_i$; /* find a better solution */
  end if
end for



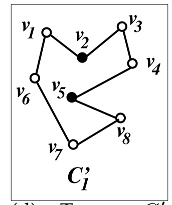
(a) The minimum spanning trees T_1 and T_2 of connected components CC_1 and CC_2, respectively



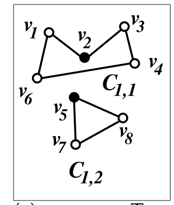
(b) Matching M^o



(c) Graph G_E obtained by adding edges in M^o to trees T_1 and T_2



(d) Tour C_1' derived from the Eulerian circuit of component CC_1' in G_E



(e) Two subtours $C_{1,1}$ and $C_{1,2}$ are derived from tour C_1'.

Fig. 2. An illustration of the execution of Algorithm 2.

B. Algorithm

We now show how to find a set $\mathcal{C}_i$ of tours that visit all devices in G', based on a given edge weight threshold δ_i. Recall that graph $G' = (V, E; w' : E \mapsto \mathbb{R}^{\geq 0})$ is constructed from G, where $w'(v_i, v_j) = t_f(v_i, v_j) + \frac{\rho(v_i)+\rho(v_j)}{2}$ for each edge (v_i, v_j) in E.

We first remove the edges with weights strictly greater than δ_i from G'. Assume that there are q connected components $CC_1, CC_2, \ldots, CC_q$ in the resulting graph after the edge removals, where $q \geq 1$ is a positive integer.

We then find a minimum spanning tree (MST) T_j in each connected component CC_j with $1 \leq j \leq q$, see Fig. 2(a). For each tree T_j, denote by V_j^o the set of *odd degree* nodes in T_j. Notice that the number of nodes in V_j^o is even. Let V^o be the set of odd degree nodes in the q trees $T_1, T_2, \ldots, T_q$, i.e., $V^o = \bigcup_{j=1}^q V_j^o$.

We thirdly construct a complete graph $G^o = (V^o, E^o; w^o : E^o \mapsto \mathbb{R}^{\geq 0})$, where there is an edge (u, v) in E^o for any two nodes u and v in V^o and the weight $w^o(u, v)$ of edge (u, v) in G^o is equal to its weight $w'(u, v)$ in G', i.e., $w^o(u, v) = w'(u, v)$. Since G^o is a complete graph and the number of nodes in G^o is even, we can find a minimum weighted perfect matching M^o in graph G^o, see Fig. 2(b).

We fourthly obtain a graph G_E by adding the edges in M^o to the q trees $T_1, T_2, \ldots, T_q$, i.e., $G_E = M^o \bigcup (\bigcup_{j=1}^q T_j)$, see Fig. 2(c). Assume that there are q' connected components $CC_1', CC_2', \ldots, CC_{q'}'$ in G_E. Notice that the two endpoints of some edge in M may lie in two different trees. Therefore, the number q' of connected components in G_E is no greater than q, i.e., $q' \leq q$. For each connected component CC_j' in G_E with $1 \leq j \leq q'$, it can be seen that the degree of each node in CC_j' is even. Then, there is a Eulerian circuit C_j^e in CC_j' [36]. A closed tour C_j' that visits each node in connected component CC_j' only once then can be obtained, by shortcutting duplicated nodes in C_j^e, see Fig. 2(d).

We finally split tour C_j' into, say n_j, subpaths $P_{j,1}, P_{j,2}, \ldots, P_{j,n_j}$ such that the cost of each subpath is no greater than $B/2$, and the number n_j of split subpaths is no more than $\lceil \frac{w'(C_j')}{B/2} \rceil$, i.e., $n_j \leq \lceil \frac{w'(C_j')}{B/2} \rceil$ [16]. Then, n_j subtours $C_{j,1}, C_{j,2}, \ldots, C_{j,n_j}$ can be derived from the n_j subpaths, where subtour $C_{j,l}$ is derived from subpath $P_{j,l}$ by connecting its two endpoints, where $1 \leq l \leq n_j$. It can be seen that the cost of each subtour $C_{j,l}$ is no more than twice the cost of subpath $P_{j,l}$, thus no more than B, i.e., $w'(C_{j,l}) \leq 2 \cdot w'(P_{j,l}) \leq 2 \cdot \frac{B}{2} = B$ with $1 \leq l \leq n_j$. Fig. 2(e) shows that two subtours $C_{1,1}$ and $C_{1,2}$ are derived from the tour C_1' in Fig. 2(d).

The detailed algorithm for finding a set $\mathcal{C}_i$ of tours that visit all nodes in G' based on a given edge weight δ_i is presented in Algorithm 2.

One may notice that there are some similarities between the proposed algorithm and Christofides' algorithm for the TSP problem [6]. An alternative method for the problem of concern is to find closed tours in the q connected components $CC_1, CC_2, \ldots, CC_q$ by directly applying Christofides' algorithm to each of the q components, which first finds a minimum spanning tree T_j in each component CC_j, then obtains a minimum weighted perfect matching M_j of odd degree nodes in T_j, and finally obtains a Eulerian circuit by adding edges in M_j to tree T_j, where $1 \leq j \leq q$. There are two main differences between the proposed algorithm in this paper and the Christofides method. (i) The weighted sum of edges in matching M^o is no greater than the weighted sum of the edges in the q matchings $M_1, M_2, \ldots, M_q$, i.e., $w'(M^o) = \sum_{e \in M^o} w'(e) \leq \sum_{i=1}^q w'(M_j) = \sum_{e \in \bigcup_{j=q}^q M_j} w'(e)$, since the edges in $\bigcup_{j=q}^q M_j$ also form a perfect matching of nodes in V^o while M^o is the minimum weighted one. We estimate an upper bound on $w'(M^o)$ (see Ineq. (9) in Section IV-C.2). However, the upper bound may less than, rather than larger than, $\sum_{i=1}^q w'(M_j)$. (ii) Some edge in M^o may connect two odd degree nodes that lie in different minimum spanning trees, whereas every edge in matching M_j connects odd degree nodes only in tree T_j with $1 \leq j \leq q$. We show that less numbers of closed tours are delivered by the proposed algorithm with an example in Section 2 of the supplementary file.

C. Algorithm Analysis

Lemma 2: Given a complete graph $G' = (V, E)$ and an edge weight function $w' : E \mapsto \mathbb{R}^{\geq 0}$, assume that the edge weights in G' satisfy the triangle inequality. For any closed tour C in G' with $w'(C) \leq B$, there are no more than $i - 1$ edges in C with their edge weights strictly greater than $\frac{B}{i}$, where i is a given integer with $i \geq 1$.

Proof: We show the claim by distinguishing into two cases: (1) there are no more than $i - 1$ edges in C; and (2) there are no less than i edges in C. Case (1) where C contains no more than $i - 1$ edges, the lemma immediately follows.

Algorithm 2 Algorithm for Finding a Set of Tours That Visit All Nodes in G' Based on a Given Edge Weight Threshold δ_i

Input: A graph $G' = (V, E; w' : E \mapsto \mathbb{R}^{\geq 0})$, a maximum cost B of each UAV tour, and an edge weight threshold δ_i

Output: A set $\mathcal{C}_i$ of tours so that the cost of each tour in $\mathcal{C}_i$ is no greater than B

1: Remove the edges with weights greater than δ_i from G'. Assume that there are q connected components $CC_1, CC_2, \ldots, CC_q$ in the resulting graph after the edge removals;
2: Find an MST T_j in each connected component CC_j with $1 \leq j \leq q$;
3: Let V^o be the set of odd degree nodes in the q MSTs;
4: Construct a complete graph $G^o = (V^o, E^o; w^o : E^o \mapsto \mathbb{R}^{\geq 0})$;
5: Find a minimum weighted perfect matching M^o in G^o;
6: Graph G_E is constructed by adding the edges in M^o to the q trees $T_1, T_2, \ldots, T_q$, i.e., $G_E = M^o \bigcup (\bigcup_{j=1}^{q} T_j)$. Assume that there are $q'(\leq q)$ connected components $CC'_1, CC'_2, \ldots, CC'_{q'}$ in G_E;
7: Let $\mathcal{C}_i \leftarrow \emptyset$; /* the set of obtained tours */
8: **for** $j \leftarrow 1$ to q' **do**
9: Find a Eulerian circuit C^e_j in connected component CC'_j and obtain a tour C'_j visiting nodes in CC'_j by shortcutting duplicated nodes in C^e_j;
10: Split tour C'_j into n_j subtours $C_{j,1}, C_{j,2}, \ldots, C_{j,n_j}$ so that the cost of each subtour is no greater than B and $n_j \leq \lceil \frac{w'(C'_j)}{B/2} \rceil$;
11: Let $\mathcal{C}_i \leftarrow \mathcal{C}_i \cup \{C_{j,1}, C_{j,2}, \ldots, C_{j,n_j}\}$;
12: **end for**

Consider Case (2) where C contains no less than i edges. Suppose that there are at least i edges in C with edge weights greater than $\frac{B}{i}$. Then, the weighted sum $w'(C)$ of edges in C is larger than $i \cdot \frac{B}{i} = B$, which contradicts the assumption $w'(C) \leq B$. The lemma then follows. □

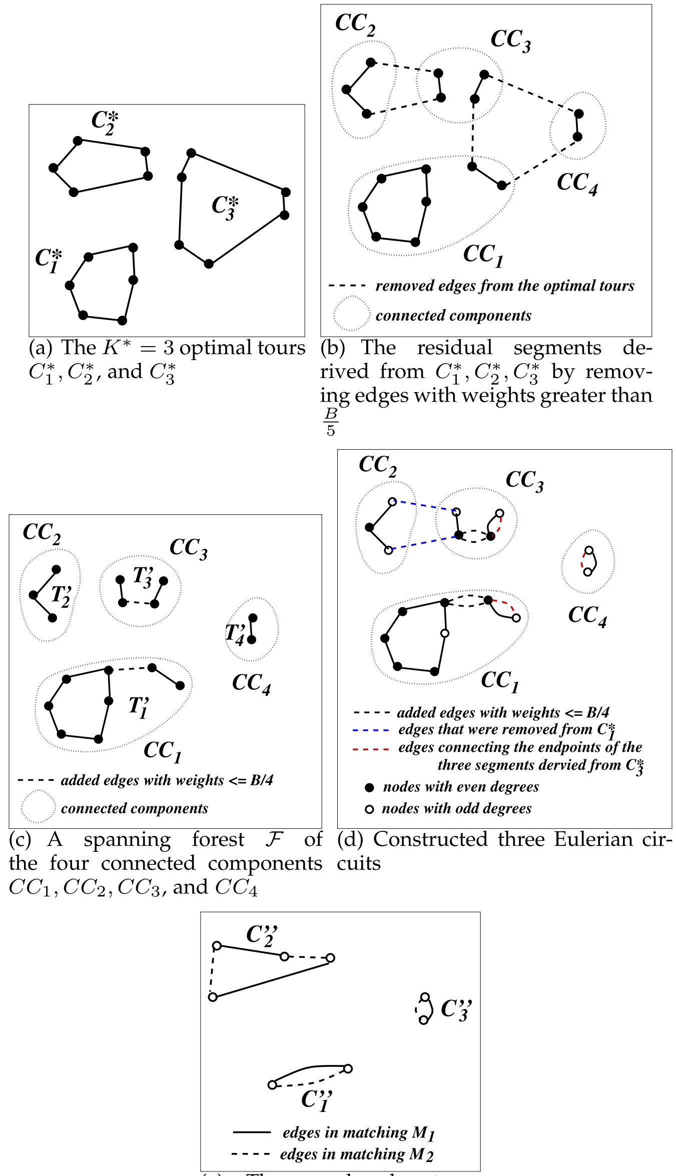
Following the similar argument as the one in [43], the values of the optimal solutions to the problem in G and G' are equal. We here only show that `Algorithm 1` delivers a 4-approximate solution to the problem in G', which also is a 4-approximate solution to the problem in G.

Assume that an optimal solution to the problem in G' consists of K^* tours $C^*_1, C^*_2, \ldots, C^*_{K^*}$. We estimate an upper bound on the number $|\mathcal{C}_i|$ of delivered tours by `Algorithm 2` with an edge weight threshold $\delta_i = \frac{B}{i}$, where $2 \leq i \leq n$. Following `Algorithm 2`, the number $|\mathcal{C}_i|$ of delivered tours is

$$
\begin{aligned}
&|\mathcal{C}_i| \\
&\leq \sum_{j=1}^{q'} \lceil \frac{w'(C'_j)}{B/2} \rceil \leq \frac{\sum_{j=1}^{q'} w'(C'_j)}{B/2} + q, \text{ as } q' \leq q \\
&= \frac{w'(G_E)}{B/2} + q, \text{ as } w'(C'_j) \leq w'(C^e_j),\ w'(G_E) = \sum_{j=1}^{q'} w'(C^e_j) \\
&= \frac{\sum_{j=1}^{q} w'(T_j) + w'(M^o)}{B/2} + q, \text{ as } G_E = M^o \bigcup (\bigcup_{j=1}^{q} T_j).
\end{aligned} \tag{3}
$$

In the following, we estimate the upper bounds of weights $\sum_{j=1}^{q} w'(T_j)$ and $w'(M^o)$, respectively.

1) Estimate an Upper Bound on $\sum_{j=1}^{q} w'(T_j)$: Following Lemma 2, each of the optimal K^* tours $C^*_1, C^*_2, \ldots, C^*_{K^*}$ contains no more than $i-1$ edges with weights greater than



(a) The $K^* = 3$ optimal tours C^*_1, C^*_2, and C^*_3

(b) The residual segments derived from C^*_1, C^*_2, C^*_3 by removing edges with weights greater than $\frac{B}{5}$

(c) A spanning forest $\mathcal{F}$ of the four connected components CC_1, CC_2, CC_3, and CC_4

(d) Constructed three Eulerian circuits

(e) Three closed tours C''_1, C''_2, C''_3 visiting nodes with odd degrees in the MSTs of components CC_1, CC_2, CC_3, CC_4.

Fig. 3. An illustration of estimating the upper bounds on $\sum_{j=1}^{q} w'(T_j)$ and $w'(M^o)$, when $\delta_i = \frac{B}{5}$.

$\frac{B}{i}$. We partition the K^* tours into i groups $\mathcal{C}^*_0, \mathcal{C}^*_1, \ldots, \mathcal{C}^*_{i-1}$, where a tour C^*_l is contained in a group $\mathcal{C}^*_j$ if C^*_l contains exactly j edges with weights greater than $\frac{B}{i}$, where $1 \leq l \leq K^*$. Let $k_j = |\mathcal{C}^*_j|$ with $0 \leq j \leq i-1$. Then, $\sum_{j=0}^{i-1} k_j = K^*$. For example, Fig. 3(a) shows $K^* = 3$ optimal tours C^*_1, C^*_2, C^*_3, where C^*_1 contains no edges with edge weights greater than $\frac{B}{i} = \frac{B}{5}$ with $i = 5$, C^*_2 contains two such edges, and C^*_3 contains three such edges. In this case, $k_0 = 1$, $k_1 = 0$, $k_2 = 1$, $k_3 = 1$, $k_4 = 0$, and $\sum_{j=0}^{5-1} k_j = 3 = K^*$.

We now estimate the upper bound on $\sum_{j=1}^{q} w'(T_j)$, where T_j is a minimum spanning tree of component CC_j at Step 1 of `Algorithm 2`. Following `Algorithm 2`, within each tour C^*_l in a group $\mathcal{C}^*_j$, j edges with edge weights greater than $\frac{B}{i}$ will be removed with $1 \leq j \leq i-1$. Then, there are j

segments $C^*_{l,1}, C^*_{l,2}, \ldots, C^*_{l,j}$ after the removals of the j edges from C^*_l. We can see that the total weight of the j segments is no more than $B - j \cdot \frac{B}{i}$, i.e.,

$$\sum_{s=1}^{j} w'(C^*_{l,s}) < B - j \cdot \frac{B}{i} = \frac{i-j}{i}B. \tag{4}$$

It also can be seen that the number of segments after the removals of edges with weights greater than $\frac{B}{i}$ from the K^* optimal tours is $n_{seg} = k_0 + \sum_{j=1}^{i-1} j \cdot k_j$, where $k_j = |\mathcal{C}^*_j|$ and each tour C^*_l in $\mathcal{C}^*_j$ are removed j edges. On the other hand, the total weight of the n_{seg} segments is no more than

$$k_0 \cdot B + \sum_{j=1}^{i-1} k_j \cdot \frac{i-j}{i}B, \text{ by Ineq. (4).} \tag{5}$$

We now construct a spanning forest of the q connected components $CC_1, CC_2, \ldots, CC_q$ at Step 1 of `Algorithm 2`. Since the n_{seg} segments are contained in the q connected components, we can obtain a spanning forest $\mathcal{F} = \{T'_1, T'_2, \ldots, T'_q\}$ of the q connected components, by adding $(n_{seg} - q)$ edges to the n_{seg} segments such that the weight of each added edge is no greater than $\frac{B}{i}$. Fig. 3(b) shows $q = 4$ connected components CC_1, CC_2, CC_3, CC_4 after the edge removals, and Fig. 3(c) shows that $q = 4$ spanning trees T'_1, T'_2, T'_3, T'_4 of connected components CC_1, CC_2, CC_3, CC_4 are obtained by adding $2 (= n_{seg} - q = 6 - 4)$ edges with edge weights no greater than $\frac{B}{4}$. Since T_j is a minimum spanning tree of connected component CC_j, we have

$$\begin{aligned}
&\sum_{j=1}^{q} w'(T_j) \\
&\quad \le w'(\mathcal{F}) = \sum_{j=1}^{q} w'(T'_j) \\
&\quad \le k_0 B + \sum_{j=1}^{i-1} k_j \frac{i-j}{i} B + (n_{seg} - q)\frac{B}{i}, \text{ by Eq. (5)} \\
&\quad = k_0 B + \sum_{j=1}^{i-1} k_j \frac{i-j}{i} B + (k_0 + \sum_{j=1}^{i-1} j k_j - q)\frac{B}{i} \\
&\quad = k_0 \frac{i+1}{i} B + \sum_{j=1}^{i-1} k_j B - \frac{qB}{i}.
\end{aligned} \tag{6}$$

2) Estimate an Upper Bound on $w'(M^o)$: We then estimate an upper bound on $w'(M^o)$, where M^o is a minimum weighted perfect matching of the nodes in V^o, and V^o is the odd degree nodes in the q trees $T_1, T_2, \ldots, T_q$.

We construct two perfect matchings M_1 and M_2 in graph G^o as follows. Having the forest $\mathcal{F} = \{T'_1, T'_2, \ldots, T'_q\}$, we first duplicate the $(n_{seg} - q)$ edges with the weight of each duplicated edge no greater than $\frac{B}{i}$, see the edges in Fig. 3(d) plotted with black dotted lines. We then add back those removed edges in the optimal tours of the first $\lceil i/2 \rceil$ groups $\mathcal{C}^*_0, \mathcal{C}^*_1, \ldots \mathcal{C}^*_{\lceil i/2 \rceil - 1}$. It can be seen that now no edges in the optimal tours of the $\lceil i/2 \rceil$ groups are removed, see the edges in Fig. 3(d) plotted with blue dotted lines. Finally, consider an optimal tour C^*_l in a group $\mathcal{C}^*_j$ with $\lceil i/2 \rceil \le j \le i-1$, recall that there are j segments $C^*_{l,1}, C^*_{l,2}, \ldots, C^*_{l,j}$ after the removals of edges with edge weights greater than $\frac{B}{i}$. For each segment $C^*_{l,s}$ with $1 \le s \le j$, the structure of $C^*_{l,s}$ is a path. We obtain a closed tour from path $C^*_{l,s}$, by connecting the two endpoints of $C^*_{l,s}$, see the edges in Fig. 3(d) plotted with red dotted lines. Denote by G^e the resulting graph. Assume that G^e consists of q'' connected components $CC''_1, CC''_2, \ldots, CC''_{q''}$. It can be seen that each connected component CC''_l is a Eulerian graph, as the degree of each node in CC''_l is even, see Fig. 3(d). Also, it can be seen that the set V^o_j of odd degree nodes in each tree T_j is contained in some connected component CC''_l, and $|V^o_j|$ is even.

Consider a Eulerian circuit C^{eu}_l in each connected component CC''_l. We can obtain a closed tour C''_l that visits only nodes in V^o by shortcutting nodes not in V^o, see Fig. 3(e). We finally derive two perfect matchings M_1 and M_2 from the q'' tours $C''_1, C''_2, \ldots, C''_{q''}$, see Fig. 3(e). Since M^o is a minimum weighted perfect matching of nodes in V^o, we have $w'(M^o) \le w'(M_1)$ and $w'(M^o) \le w'(M_2)$. Then,

$$\begin{aligned}
&w'(M^o) \\
&\quad \le \frac{w'(M_1) + w'(M_2)}{2} \\
&\quad = \frac{\sum_{l=1}^{q''} w'(C''_l)}{2}, \text{ as } w'(M_1) + w'(M_2) = \sum_{l=1}^{q''} w'(C''_l) \\
&\quad \le \frac{\sum_{l=1}^{q''} w'(C^{eu}_l)}{2}, \text{ as } w'(C''_l) \le w'(C^{eu}_l) \\
&\quad = \frac{w'(G^e)}{2}, \text{ as } w'(G^e) = \sum_{l=1}^{q''} w'(C^{eu}_l).
\end{aligned} \tag{7}$$

Following the construction of graph G^e, it can be seen that G^e consists of (i) the optimal tours in the first $\lceil i/2 \rceil$ groups $\mathcal{C}^*_0, \mathcal{C}^*_1, \ldots \mathcal{C}^*_{\lceil i/2 \rceil - 1}$, as no edges are removed, where the weight of each optimal tour is no greater than B; (ii) the $(\sum_{j=\lceil i/2 \rceil}^{i-1} j k_j)$ segments derived from the optimal tours in the rest groups $\mathcal{C}^*_{\lceil i/2 \rceil}, \mathcal{C}^*_{\lceil i/2 \rceil + 1}, \ldots \mathcal{C}^*_{i-1}$, where the total weight of the segments is no greater than $\sum_{j=\lceil i/2 \rceil}^{i-1} k_j \frac{i-j}{i} B$ by Ineq. (4); (iii) the added edges between two endpoints of the $(\sum_{j=\lceil i/2 \rceil}^{i-1} j k_j)$ segments, where the total weight of the added edges is no greater than the total weight of the $(\sum_{j=\lceil i/2 \rceil}^{i-1} j k_j)$ segments, as edge weights in the graph satisfy the triangle inequality; and (iv) $2(n_{seg} - q)$ edges with the weight of each edge no greater than $\frac{B}{i}$. Then,

$$\begin{aligned}
&w'(G^e) \\
&\le \sum_{j=0}^{\lceil i/2 \rceil - 1} k_j B + \sum_{j=\lceil i/2 \rceil}^{i-1} k_j \frac{i-j}{i} B \\
&\quad + \sum_{j=\lceil i/2 \rceil}^{i-1} k_j \frac{i-j}{i} B + 2(n_{seg} - q)\frac{B}{i} \\
&= \sum_{j=0}^{\lceil i/2 \rceil - 1} k_j B + \sum_{j=\lceil i/2 \rceil}^{i-1} k_j \frac{i-j}{i} 2B + 2(k_0 + \sum_{j=1}^{i-1} j k_j - q)\frac{B}{i} \\
&= \frac{i+2}{i} k_0 B + \sum_{j=1}^{\lceil i/2 \rceil - 1} \frac{i+2j}{i} k_j B + \sum_{j=\lceil i/2 \rceil}^{i-1} 2 k_j B - \frac{2qB}{i}.
\end{aligned} \tag{8}$$

An upper bound of $w'(M^o)$ thus is

$$\begin{aligned} &w'(M^o) \\ &\le \frac{w'(G^e)}{2}, \text{ by Ineq. (7)} \\ &\le \frac{i+2}{2i}k_0B + \sum_{j=1}^{\lceil i/2\rceil -1}\frac{i+2j}{2i}k_jB + \sum_{j=\lceil i/2\rceil}^{i-1}k_jB - \frac{qB}{i}. \quad (9)\end{aligned}$$

3) Approximation Ratio Analysis:

Theorem 1: Given an IoT network $G = (V, E; t_f : E \mapsto \mathbb{R}^{\ge 0}, \rho : V \mapsto \mathbb{R}^{\ge 0})$ and a maximum data collection delay B, there is a 4-approximation algorithm, `Algorithm 1`, for the minimum UAV deployment problem without neighborhoods, which takes time $O(n^4)$, where $n = |V|$.

Proof: By combining Inequalities (6) and (9), we have

$$\sum_{j=1}^{q} w'(T_j) + w'(M^o) \le (1.5+2/i)k_0B + \sum_{j=1}^{i-1}2k_jB - \frac{2qB}{i}. \quad (10)$$

By combining Ineq. (3) and Ineq. (10), we upper bound the number of delivered tours in $\mathcal{C}_i$ as

$$|\mathcal{C}_i| \le (3+4/i)k_0 + 4\sum_{j=1}^{i-1}k_j - \frac{4q}{i} + q. \quad (11)$$

Consider the case where $i = 4$, we have $|\mathcal{C}_4| \le 4\sum_{j=0}^{3}k_j = 4K^*$, as $K^* = \sum_{j=0}^{3}k_j$.

Recall that the number of tours in $\mathcal{C}$ delivered by `Algorithm 1` is $|\mathcal{C}| = \min_{i=2}^{n}\{|\mathcal{C}_i|\} \le |\mathcal{C}_4| \le 4K^*$. This indicates that $\mathcal{C}$ is a 4-approximation solution.

We finally analyze the time complexity of `Algorithm 1`, which is dominated by invoking `Algorithm 2` no more than $n-1$ times. Note that the most time-consuming operation in `Algorithm 2` is the finding of a minimum weighted perfect matching M^o in G^o, which takes time $O(n^3)$. Therefore, the time complexity of `Algorithm 1` is $(n-1)\cdot O(n^3) = O(n^4)$. □

V. Approximation Algorithm for the Minimum UAV Deployment Problem With Neighborhoods

In this section, we consider the minimum UAV deployment problem with neighborhoods, where a UAV can collect the data of each IoT device when their Euclidean distance is within a given communication range, and we propose a novel approximation algorithm for the problem.

A. Algorithm Framework

Given an IoT network $G = (V, E; t_f : E \mapsto \mathbb{R}^{\ge 0}, \rho : V \mapsto \mathbb{R}^{\ge 0})$, neighborhoods of IoT devices in V, and a maximum data collection delay B, the algorithm framework for the problem with neighborhoods is similar to the one without neighborhoods in the previous section. That is, it finds a set $\mathcal{C}_i$ of tours that visit devices in G based on a given edge weight threshold $\delta_i = \frac{B}{i}$, subject to the constraint that the cost of each tour in $\mathcal{C}_i$ is no greater than B, where $2 \le i \le n$. The final solution $\mathcal{C}$ to the problem is the set with the minimum number of tours, i.e., $|\mathcal{C}| = \min_{2\le i\le n}\{|\mathcal{C}_i|\}$. The algorithm for the minimum UAV deployment problem with neighborhoods is presented in `Algorithm 3`.

Algorithm 3 Algorithm for the Minimum UAV Deployment Problem With Neighborhoods (**approAlgNei**)

Input: a network G, the neighborhood $D(v_i)$ of each device v_i in V, and a maximum data collection delay B

Output: a set $\mathcal{C}$ of tours to visit the neighborhoods of nodes in V, so that the total spent time of each tour in $\mathcal{C}$ is no more than B.

Construct a trivial solution $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$, where each tour C_i consists of only a single node v_i in V, and $n = |V|$;
for $i \leftarrow 2$ to n **do**
  Let $\delta_i \leftarrow \frac{B}{i}$; /* set an edge weight threshold */
  Find a set $\mathcal{C}_i$ of tours to visit neighborhoods of nodes in G based on δ_i, subject to that the cost of each tour in $\mathcal{C}_i$ is no greater than B, by invoking `Algorithm 4`;
  if $|\mathcal{C}_i| < |\mathcal{C}|$ **then**
    Let $\mathcal{C} \leftarrow \mathcal{C}_i$; /* find a better solution */
  end if
end for
return $\mathcal{C}$.

B. Algorithm

We now show how to find a set $\mathcal{C}_i$ of tours visiting neighborhoods of all devices in G, which is **different** from the one in Section IV-B for the problem without neighborhoods, as the data of an IoT device can be collected by a UAV as long as the Euclidean distance between them is no more than their communication range.

For any two nodes v_j and v_l in V, recall that their neighborhoods are $D(v_j)$ and $D(v_l)$, respectively. Denote by $c(D_j, D_l)$ *the minimum flying time* between the neighborhoods $D(v_j)$ and $D(v_l)$, which is defined as follows. If the two neighborhoods $D(v_j)$ and $D(v_l)$ overlap with each other, i.e., the Euclidean distance $d(v_j, v_l)$ between nodes v_j and v_l is no greater than $2R_0$, we define $c(D_j, D_l) = 0$, where R_0 is the disk radius of each neighborhood. On the other hand, if $D(v_j)$ and $D(v_l)$ do not overlap with each other, we define $c(D_j, D_l) = \frac{d(v_j,v_l)-2R_0}{\eta}$, where η is the flying speed of a UAV. That is,

$$c(D_j, D_l) = \begin{cases} 0, & \text{if } d(v_j, v_l) \le 2R_0 \\ \frac{d(v_j,v_l)-2R_0}{\eta}, & \text{if } d(v_j, v_l) > 2R_0 \end{cases} \quad (12)$$

We partition the IoT devices in V into several disjoint subsets as follows. We first construct an auxiliary graph $G' = (V, E; w' : E \mapsto \mathbb{R}^{\ge 0})$ from G, where the weight $w'(v_j, v_l)$ of each edge (v_j, v_l) in E is the minimum flying time $c(D_j, D_l)$ between neighborhoods $D(v_j)$ and $D(v_l)$, i.e., $w'(v_j, v_l) = c(D_j, D_l)$. We then obtain a graph $G'' = (V, E'')$ from G', by removing the edges with weights strictly greater than the given edge weight threshold $\delta_i = \frac{B}{i}$ from G'. Assume that there are q connected components $CC_1, CC_2, \ldots, CC_q$ in G'', where $q \ge 1$ is a positive integer. Accordingly, the set V of devices is partitioned into q disjoint subsets $V_1, V_2, \ldots, V_q$, where V_j is the set of nodes in connected component CC_j with $1 \le j \le q$. Let $V_j = \{v_1, v_2, \ldots, v_{n_j}\}$, where n_j is the number of nodes in V_j.

Having found the q disjoint subsets $V_1, V_2, \ldots, V_q$, we can find an approximate shortest tour C_j to visit all neighborhoods of nodes in each subset V_j, by invoking the best approximation algorithm so far for the *Traveling Salesman Problem with Neighborhoods (TSPN)* [10], where $1 \le j \le q$. Assume that $C_j = p_1 \to p_2 \to \cdots \to p_{n_j} \to p_1$, where p_l is a hovering location in the neighborhood $D(v_l)$ of a node v_l with $1 \leq l \leq n_j$.

Recall that $w_N(C_j)$ is the total time of a UAV spent in tour C_j, where $w_N(C_j) = \sum_{l=1}^{n_j} t_f(p_l, p_{l+1}) + \sum_{l=1}^{n_j} \rho(v_l)$, $t_f(p_l, p_{l+1})$ is the flying time between p_l and p_{l+1}, $\rho(v_l)$ is the duration of data collection from device v_l. Denote by r the flying time of a UAV for a distance of R_0, i.e., $r = \frac{R_0}{\eta}$, where R_0 is the radius of each neighborhood $D(v_l)$ and η is the UAV flying speed. Following the work in [10], we have $w_N(C_j) \leq 6.75 \cdot w_N(C_j^*) + 20.4 \cdot r$, where C_j^* is an optimal (i.e., shortest) tour for the TSPN problem.

We assign the weight $w'_N(p_l, p_{l+1})$ of each edge (p_l, p_{l+1}) in C_j as $w'_N(p_l, p_{l+1}) = t_f(p_l, p_{l+1}) + \frac{\rho(v_l)+\rho(v_{l+1})}{2}$. It can be seen that $w'_2(C_j) = \sum_{l=1}^{n_j} w'_2(p_l, p_{l+1}) = \sum_{l=1}^{n_j}(w(p_l, p_{l+1}) + \frac{\rho(v_l)+\rho(v_{l+1})}{2}) = \sum_{l=1}^{n_j} w(p_l, p_{l+1}) + \sum_{l=1}^{n_j} \rho(v_l) = w_2(C_j)$

We finally obtain, say s_j, subtours $C_{j,1}, C_{j,2}, \ldots, C_{j,s_j}$ from C_j by the tour splitting procedure in [16], such that the cost of each subtour $C_{j,l}$ is no greater than B, and the number s_j of subtours is no more than $\lceil \frac{w'_N(C_j)}{B/2} \rceil$, i.e., $s_j \leq \lceil \frac{w'_N(C_j)}{B/2} \rceil$.

The set $\mathcal{C}_i$ of obtained tours based on a given edge weight δ_i then is $\mathcal{C}_i = \bigcup_{j=1}^{q}(\bigcup_{l=1}^{b_j} C_{j,l})$. The detailed algorithm for finding a set $\mathcal{C}_i$ of tours that visit neighborhoods of nodes in G based on a given edge weight δ_i is presented in `Algorithm` 4.

Algorithm 4 Algorithm for Finding a Set of Tours That Visit All Neighborhoods of Nodes in G Based on a Given Edge Weight Threshold δ_i

Input: A graph $G = (V, E; t_f : E \mapsto \mathbb{R}^{\geq 0}, \rho : V \mapsto \mathbb{R}^{\geq 0})$, a maximum delay B, and an edge weight threshold δ_i

Output: A set $\mathcal{C}_i$ of tours so that the cost of each tour in $\mathcal{C}_i$ is no greater than B

1: Construct an auxiliary graph $G' = (V, E; w' : E \mapsto \mathbb{R}^{\geq 0})$ from G, where the weight $w'(v_j, v_l)$ of each edge (v_j, v_l) in E is the minimum flying time $c(D_j, D_l)$ between neighborhoods $D(v_j)$ and $D(v_l)$;
2: Graph $G'' = (V, E'')$ is derived from G' by removing the edges with weights greater than δ_i from G'. Assume that there are q connected components $CC_1, CC_2, \ldots, CC_q$ in G''. Denote by V_j the set of nodes in CC_j with $1 \leq j \leq q$.
3: Let $\mathcal{C}_i \leftarrow \emptyset$; /* the set of obtained tours */
4: **for** $j \leftarrow 1$ to q **do**
5: Find an approximate tour C_j to visit all neighborhoods of nodes in V_j, by invoking the algorithm in [10] for the TSPN problem;
6: Split tour C_j into, say s_j, subtours $C_{j,1}, C_{j,2}, \ldots, C_{j,s_j}$ so that the cost of each subtour is no greater than B and $s_j \leq \lceil \frac{w'_N(C_j)}{B/2} \rceil$;
7: Let $\mathcal{C}_i \leftarrow \mathcal{C}_i \cup \{C_{j,1}, C_{j,2}, \ldots, C_{j,s_j}\}$;
8: **end for**
9: **return** $\mathcal{C}_i$.

C. Algorithm Analysis

Assume that an optimal solution to the problem in G' consists of K^* tours $C_1^*, C_2^*, \ldots, C_{K^*}^*$. Following Lemma 2, each of the optimal K^* tours contains no more than $i-1$ edges with weights greater than $\frac{B}{i}$. We partition the K^* optimal tours into i groups $\mathcal{C}_0^*, \mathcal{C}_1^*, \ldots, \mathcal{C}_{i-1}^*$, where a tour C_l^* is contained in a group $\mathcal{C}_j^*$ if C_l^* contains exactly j edges with weights greater than $\frac{B}{i}$, where $1 \leq l \leq K^*$. Let $k_j = |\mathcal{C}_j^*|$ with $0 \leq j \leq i-1$. Then, $\sum_{j=0}^{i-1} k_j = K^*$.

Following `Algorithm` 4, there are q connected components $CC_1, CC_2, \ldots, CC_q$ in G'' after the removals of the edges with weights strictly greater than δ_i from G'. Denote by C_j^* the optimal closed tour for visiting the neighborhoods of nodes in connected component CC_j with $1 \leq j \leq q$.

We first estimate an upper bound on the total weight $\sum_{j=1}^{q} w'_N(C_j^*)$ of the q tours $C_1^*, C_2^*, \ldots, C_q^*$ by the following lemma.

Lemma 3: The total weight $\sum_{j=1}^{q} w'_N(C_j^*)$ of the q tours $C_1^*, C_2^*, \ldots, C_q^*$ with a given edge weight threshold $\delta_i = \frac{B}{i}$ is upper bounded by $k_0\frac{i+2}{i}B + 2\sum_{j=1}^{i-1} k_j B - \frac{2qB}{i} + 8r(k_0 + \sum_{j=1}^{i-1} j \cdot k_j - q)$, where k_j is the number of optimal tours in group $\mathcal{C}_j^*$ with $0 \leq j \leq i-1$, B is the maximum cost of each tour, r is the flying time for a distance of R_0 by a UAV, and R_0 is the radius of each neighborhood.

Proof: The proof body can be seen in Section 4 of the supplementary file. □

We now estimate the upper bound on the number of tours delivered by `Algorithm` 3 as follows.

Theorem 2: Given an IoT network $G = (V, E; t_f : E \mapsto \mathbb{R}^{\geq 0}, \rho : V \mapsto \mathbb{R}^{\geq 0})$, a maximum data collection delay B, and the radius R_0 of each neighborhood, there is an approximation algorithm, `Algorithm` 3, for the minimum UAV deployment problem with neighborhoods, such that the number of delivered tours is no greater than $(27+108\lambda) \cdot K^* - (67.2\lambda + 12.5)$, where K^* is the minimum number of tours, $\lambda = \frac{r}{B}$, and r is the UAV flying time for a distance of R_0, which usually is a very small constant.

Proof: The proof body can be seen in Section 5 of the supplementary file.

Remark: Notice that the value of λ usually is small. For example, a DJI Phantom 4 Pro UAV can fly at a speed of η=10 m/s [24]. Assume that the radius of each neighborhood is $R_0 = 500$ meters [1] and the maximum data collection delay B is 30 minutes. Then, the value of λ is

$$\lambda = \frac{r}{B} = \frac{\frac{R_0}{\eta}}{B} = \frac{\frac{500\ m}{10\ m/s}}{30\ min} = \frac{1}{36}. \tag{13}$$

In this case, the number $|\mathcal{C}|$ of closed tours delivered by `Algorithm 3` can be upper bounded as

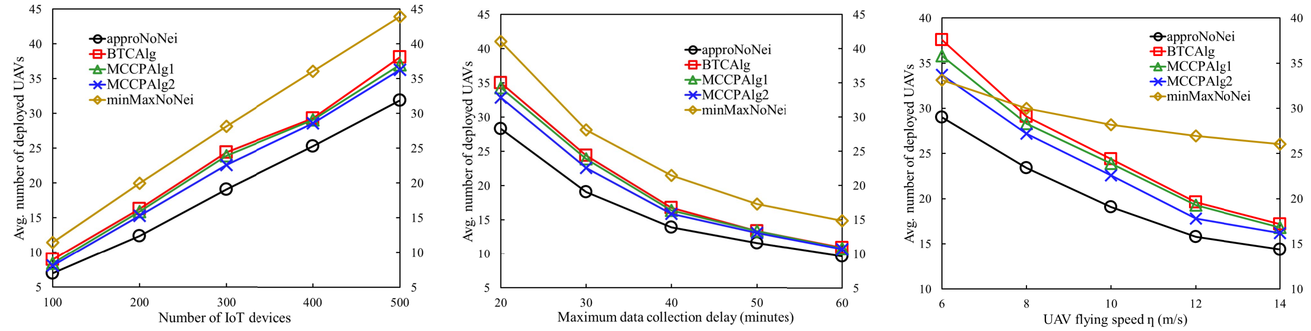
$$\begin{aligned} |\mathcal{C}| &\leq (27 + 108\lambda)K^* - (67.2\lambda + 12.5) \\ &\leq 30K^* - 14.36, \text{ where } \lambda = \frac{1}{36}. \end{aligned} \tag{14}$$

We thus have that $|\mathcal{C}| \leq 30K^* - 15$, as $|\mathcal{C}|$ is a positive integer. □

VI. Performance Evaluation

A. Simulation Environment

We consider an IoT network area in a 5 km×5 km×1 km three-dimensional Euclidean space. There are from 100 to 500 IoT devices randomly deployed on the ground of a monitoring area. The amount of data Δ_i of each IoT device v_i is randomly chosen from an interval from 5 MB to 10 MB [38]. The data transmission rate b is 1 Mb/s [26]. The flying speed η of each UAV is 10 m/s [24]. The maximum data collection delay B ranges from 20 minutes to one hour. On the other hand, when the data of each IoT device can be collected by a UAV with wireless transmissions, we assume that all UAVs



(a) Vary the IoT network size n from 100 to 500, when $B = 30$ minutes and $\eta = 10$ m/s

(b) Vary the maximum data collection delay B from 20 minutes to one hour, when $n = 300$ and $\eta = 10\ m/s$

(c) Vary the UAV flying speed η from 6 m/s to 14 m/s, when $n = 300$ and $B = 30$ minutes

Fig. 4. Performance of different algorithms for the minimum UAV deployment problem *without neighborhoods*.

hover at an altitude $h = 300$ m and the transmission range R ranges from 400 meters to 600 meters [1]. Then, the radius of each neighborhood is $R_0 = \sqrt{R^2 - h^2}$ meters.

To evaluate the performance of the proposed algorithm `approAlgNoNei` for the minimum UAV deployment problem without neighborhoods, we consider four existing benchmarks. (i) Algorithm `BTCAlg` [15] proposed a 5-approximation algorithm for the problem. (ii) Algorithm `MCCPAlg1` [37] delivered an improved $4\frac{2}{3}$-approximate solution to the problem. (iii) Algorithm `MCCPAlg2` [37] found a $4\frac{4}{7}$-approximate solution, by better refining the algorithm `MCCPAlg1`. Notice that $4\frac{4}{7} < 4\frac{2}{3}$. (iv) Algorithm `minMaxNoNei` [36] finds a given number K of closed tours to visit the n nodes such that the longest length among the K found tours is minimized. We find the minimum number K of UAVs deployed by invoking algorithm `minMaxNoNei` multiple times through increasing the value of K from 1 to n until the longest tour length is no greater than B, where n is the number of IoT devices.

On the other hand, to evaluate the performance of the proposed algorithm `approAlgNei` for the minimum UAV deployment problem with neighborhoods, in addition to the four algorithms `BTCAlg`, `MCCPAlg1`, `MCCPAlg2`, and `minMaxNoNei` that ignore neighborhoods, we compare with another benchmark algorithm `minMaxNei` [8] that takes the neighborhoods into consideration, which is to find a given number K of closed tours to visit the neighborhoods of n nodes such that the longest length among the K found tours is minimized, where at least one location in each neighborhood must be visited by a tour.

Each value in the figures is the average of the results by applying each algorithm to 100 different network topologies with the same network size.

B. Algorithm Performance for the Minimum UAV Deployment Problem Without Neighborhoods

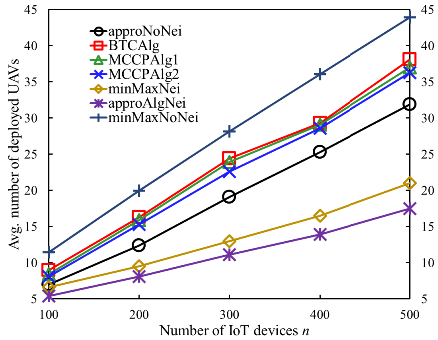
We first evaluate the performance of different algorithms by varying the number of IoT devices n from 100 to 500, while fixing the maximum delay B at 30 minutes and the UAV flying speed η at 10 m/s. Fig. 4(a) shows that the number of UAVs deployed by each algorithm on average increases with the growth of the number of IoT devices n, as more UAVs need to be deployed in a large network. Fig. 4(a) also demonstrates that the average number of UAVs by the proposed algorithm `approAlgNoNei` is around from 11% to 19% less than those by the three benchmark algorithms `BTCAlg`, `MCCPAlg1` and `MCCPAlg2`. For example, the average numbers of UAVs by algorithms `approAlgNoNei`, `BTCAlg`, `MCCPAlg1` and `MCCPAlg2` are about 19.1, 24.4, 23.9, and 22.6, respectively, when there are $n = 300$ IoT devices in the network.

We then study the performance of different algorithms by varying the maximum data collection delay B from 20 minutes to one hour, when $n = 300$ and $\eta = 10$ m/s. Fig. 4(b) plots the performance of algorithms `approAlgNoNei`, `BTCAlg`, `MCCPAlg1` and `MCCPAlg2`, from which it can be seen that the average number of UAVs deployed by each algorithm decreases quickly with the increase of the maximum data collection delay B. Fig. 4(b) also shows that the average number of UAVs by algorithm `approAlgNoNei` is around from 9.5% to 15.5% less than those by algorithms `BTCAlg`, `MCCPAlg1` and `MCCPAlg2`.

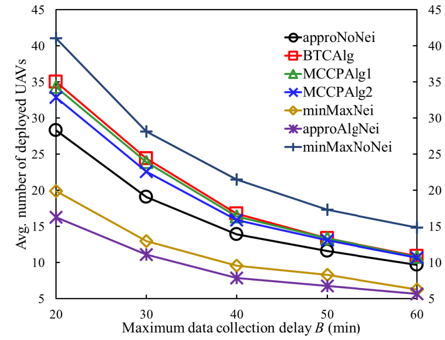
We finally investigate the performance of the four mentioned algorithms by varying the UAV flying speed η from 6 m/s to 14 m/s, when $B = 30$ minutes and $n = 300$. Fig. 4(c) demonstrates that less number of UAVs are deployed in the solution delivered by each algorithm when each UAV flies at a higher speed. Fig. 4(c) further shows that the average number of UAVs by algorithm `approAlgNoNei` is about from 11% to 15.5% less than those by algorithms `BTCAlg`, `MCCPAlg1` and `MCCPAlg2`.

C. Algorithm Performance for the Minimum UAV Deployment Problem With Neighborhoods

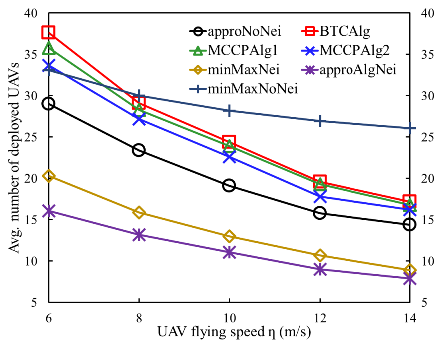
We now evaluate the performance of different algorithms for the minimum UAV deployment problem with neighborhoods. Fig. 5(a) plots the performance of different algorithms by varying the number of IoT devices n from 100 to 500, when $B = 30$ minutes, $R = 500$ m and $\eta = 10$ m/s. It can be seen that the numbers of UAVs by both algorithms `approAlgNei` and `minMaxNei` are much smaller than those by the four algorithms `approAlgNoNei`, `BTCAlg`, `MCCPAlg1` and `MCCPAlg2`, as the first two algorithms take the transmission range of each IoT device into consideration, i.e., a UAV can collect the data of the IoT device when their Euclidean distance is no greater than the transmission range, whereas the rest of the four algorithms do not consider the



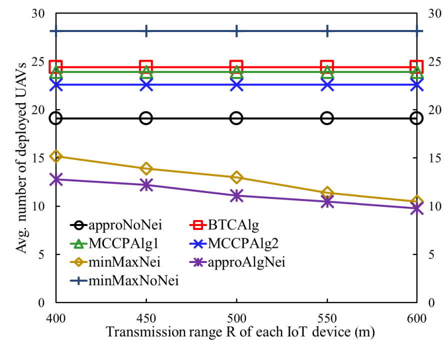
(a) Vary the IoT network size n from 100 to 500, when $B = 30$ minutes and $\eta = 10$ m/s



(b) Vary the maximum data collection delay B from 20 minutes to one hour, when $n = 300$ and $\eta = 10$ m/s



(c) Vary the UAV flying speed η from 6 m/s to 14 m/s, when $n = 300$ and $B = 30$ min



(d) Vary the transmission range R of each IoT device from 400 m to 600 m, when $n = 300$, $B = 30$ minutes and $\eta = 10$ m/s

Fig. 5. Performance of different algorithms for the minimum UAV deployment problem *with neighborhoods*.

transmission range and a UAV needs to fly to the location of the IoT device to collect its data. In addition, Fig. 5(a) shows that the average number of UAVs deployed by the proposed algorithm `approAlgNei` is around from 14.5% to 18% less than that by algorithm `minMaxNei`. For example, the average numbers of UAVs by algorithms `approAlgNei` and `minMaxNei` are around 5.5 and 6.6, respectively, when there are $n = 100$ IoT devices.

Fig. 5(b) demonstrates the performance of different algorithms by varying the maximum data collection delay B from 20 minutes to one hour, from which it can be seen that the average number of UAVs by each algorithm decreases quickly with the growth of B. Fig. 5(b) also shows that the number of UAVs by the proposed algorithm `approAlgNei` is around from 9.5% to 18% less than that by algorithm `minMaxNei`, and at least 40% less than those by the other four algorithms `approAlgNoNei`, `BTCAlg`, `MCCPAlg1` and `MCCPAlg2`.

Fig. 5(c) compares the performance of different algorithms by varying the UAV flying speed η from 6 m/s to 14 m/s, when $B = 30$ minutes, $n = 300$ and $R = 500$ meters. Fig. 5(c) shows that the average number of UAVs by algorithm `approAlgNei` is at least 11% less than those by the other five algorithms.

Fig. 5(d) shows the performance of different algorithms by varying the transmission range R of each IoT device from 400 meters to 600 meters, when $B = 30$ minutes, $n = 300$ and $\eta = 10$ m/s. It can be seen that the numbers of UAVs by both algorithms `approAlgNei` and `minMaxNei` decreases with the growth of the transmission range R, as the flying distance of each UAV is shorter with a larger transmission range R and the saved UAV flying time can be used for collecting data from more IoT devices, thereby deploying less UAVs. In contrast, Fig. 5(d) demonstrates that the numbers of UAVs by the other four algorithms `approAlgNoNei` `BTCAlg`, `MCCPAlg1` and `MCCPAlg2` do not change with the increase on the transmission range, since the four algorithms do not consider the transmission range and each UAV flies to the location of each IoT device to collect its data. Finally, Fig. 5(d) plots that the average number of UAVs by the proposed algorithm `approAlgNei` is around from 6.5% to 15.5% less than that by algorithm `minMaxNei`.

VII. Conclusion

In this paper, we first formulated the minimum UAV deployment problem, which is to determine the minimum number K of UAVs and to find the data collection tours for the K UAVs to collect data from IoT devices in an IoT network, subject to that the total time spent by each UAV per tour is no greater than a given delay. We then proposed a 4-approximation algorithm for the problem without neighborhoods, which improves the best approximation ratio $4\frac{4}{7}$ for the problem so far. We also devised the very first constant approximation algorithm for the problem with neighborhoods. We finally evaluated the performance of the proposed algorithms via simulation experiments, and experimental results showed that the number of deployed UAVs by the proposed algorithms are around 11% to 19% less than those by existing algorithms on average.

REFERENCES

[1] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[3] E. M. Arkin, R. Hassin, and A. Levin, "Approximations for minimum and min-max vehicle routing problems," *J. Algorithms*, vol. 59, no. 1, pp. 1–18, Apr. 2006.

[4] A. M. Bedewy, Y. Sun, and N. B. Shroff, "The age of information in multihop networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1248–1257, Jun. 2019.

[5] S. Chandrasekharan *et al.*, "Designing and implementing future aerial communication networks," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 26–34, May 2016.

[6] N. Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem," Graduate School Ind. Admin., Carnegie-Mellon Univ., Pittsburgh, PA, USA, Res. Rep. 388, 1976.

[7] H. Dai, X. Wu, G. Chen, L. Xu, and S. Lin, "Minimizing the number of mobile chargers for large-scale wireless rechargeable sensor networks," *Comput. Commun.*, vol. 46, pp. 54–65, Jun. 2014.

[8] L. Deng *et al.*, "Approximation algorithms for the min-max cycle cover problem with neighborhoods," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1845–1858, Aug. 2020.

[9] A. Dumitrescu and J. S. B. Mitchell, "Approximation algorithms for TSP with neighborhoods in the plane," *J. Algorithms*, vol. 48, no. 1, pp. 135–159, Aug. 2003.

[10] A. Dumitrescu and C. D. Tóth, "The traveling salesman problem for lines, balls, and planes," *ACM Trans. Algorithms*, vol. 12, no. 3, pp. 1–29, Jun. 2016.

[11] A. Dumitrescu and C. D. Tóth, "Constant-factor approximation for TSP with disks," in *A Journey Through Discrete Mathematics*. Cham, Switzerland: Springer, 2017, pp. 375–390.

[12] D. Ebrahimi, S. Sharafeddine, P.-H. Ho, and C. Assi, "UAV-aided projection-based compressive data gathering in wireless sensor networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1893–1905, Apr. 2019.

[13] K. Elbassioni, A. V. Fishkin, and R. Sitters, "Approximation algorithms for the Euclidean traveling salesman problem with discrete and continuous neighborhoods," *Int. J. Comput. Geometry Appl.*, vol. 19, no. 2, pp. 173–193, Apr. 2009.

[14] Q. Guo *et al.*, "Minimizing the longest tour time among a fleet of UAVs for disaster area surveillance," *IEEE Trans. Mobile Comput.*, early access, Nov. 16, 2020, doi: 10.1109/TMC.2020.3038156.

[15] M. R. Khani and M. R. Salavatipour, "Improved approximation algorithms for the min-max tree cover and bounded tree cover problems," *Algorithmica*, vol. 69, no. 2, pp. 443–460, Jun. 2014.

[16] S. Khuller, M. Purohit, and K. K. Sarpatwar, "Analyzing the optimal neighborhood: Algorithms for partial and budgeted connected dominating set problems," *SIAM J. Discrete Math.*, vol. 34, no. 1, pp. 251–270, Jan. 2020.

[17] C. Konstantopoulos, G. Pantziou, D. Gavalas, A. Mpitziopoulos, and B. Mamalis, "A rendezvous-based approach enabling energy-efficient sensory data collection with mobile sinks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 5, pp. 809–817, May 2012.

[18] M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 3, no. 1, pp. 45–54, Mar. 2013.

[19] Y. Li, W. Liang, W. Xu, and X. Jia, "Data collection of IoT devices using an energy-constrained UAV," in *Proc. 34th IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, Jun. 2020, pp. 644–653.

[20] Y. Liang *et al.*, "Nonredundant information collection in rescue applications via an energy-constrained UAV," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2945–2958, Apr. 2019.

[21] W. Liang, W. Xu, X. Ren, X. Jia, and X. Lin, "Maintaining large-scale rechargeable sensor networks perpetually via multiple mobile charging vehicles," *ACM Trans. Sensor Netw.*, vol. 12, no. 2, pp. 1–26, May 2016.

[22] Y. Lin, T. Wang, and S. Wang, "UAV-assisted emergency communications: An extended multi-armed bandit perspective," *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 938–941, May 2019.

[23] V. Nagarajan and R. Ravi, "Approximation algorithms for distance constrained vehicle routing problems," *Networks*, vol. 59, no. 2, pp. 209–214, Mar. 2012.

[24] (2019). *Phantom 4 Specification*. [Online]. Available: https://www.dji.com/phantom-4-adv/info

[25] (2021). *Matrice M300 RTK Specification*. [Online]. Available: https://www.dji.com/cn/matrice-300/specs

[26] X. Ren, W. Liang, and W. Xu, "Data collection maximization in renewable sensor networks via time-slot scheduling," *IEEE Trans. Comput.*, vol. 64, no. 7, pp. 1870–1883, Jul. 2015.

[27] K. T. San, E. Y. Lee, and Y. S. Chang, "The delivery assignment solution for swarms of UAVs dealing with multi-dimensional chromosome representation of genetic algorithm," in *Proc. IEEE 7th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2016, pp. 1–7.

[28] C. Wang, J. Li, Y. Yang, and F. Ye, "Combining solar energy harvesting with wireless charging for hybrid wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 3, pp. 560–576, Mar. 2018.

[29] X. Xu, J. Luo, and Q. Zhang, "Delay tolerant event collection in sensor networks with mobile sink," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[30] W. Xu, W. Liang, and X. Lin, "Approximation algorithms for min-max cycle cover problems," *IEEE Trans. Comput.*, vol. 64, no. 3, pp. 600–613, Mar. 2015.

[31] W. Xu *et al.*, "Approximation algorithms for the team orienteering problem," in *Proc. IEEE Conf. Comput. Commun.*, Jul. 2020, pp. 1389–1398.

[32] W. Xu *et al.*, "Approximation algorithms for the generalized team orienteering problem and its applications," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 176–189, Feb. 2021.

[33] W. Xu, W. Liang, X. Jia, H. Kan, Y. Xu, and X. Zhang, "Minimizing the maximum charging delay of multiple mobile chargers under the multi-node energy charging scheme," *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 1846–1861, May 2021.

[34] Y. Yang, Z. Zheng, K. Bian, L. Song, and Z. Han, "Real-time profiling of fine-grained air quality index distribution using UAV sensing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 186–198, Feb. 2018.

[35] C. You and R. Zhang, "3D trajectory optimization in Rician fading for UAV-enabled data harvesting," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 3192–3207, Jun. 2019.

[36] W. Yu and Z. Liu, "Improved approximation algorithms for some min-max and minimum cycle cover problems," *Theor. Comput. Sci.*, vol. 654, pp. 45–58, Nov. 2016.

[37] W. Yu, Z. Liu, and X. Bao, "New approximation algorithms for the minimum cycle cover problem," *Theor. Comput. Sci.*, vol. 793, pp. 44–58, Nov. 2019.

[38] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in UAV enabled wireless sensor network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 328–331, Jun. 2018.

[39] C. Zhan, Y. Zeng, and R. Zhang, "Trajectory design for distributed estimation in UAV-enabled wireless sensor network," *IEEE Trans. Veh. Techn.*, vol. 67, no. 10, pp. 10155–10159, Oct. 2018.

[40] J. Zhang *et al.*, "Minimizing the number of deployed UAVs for delay-bounded data collection of IoT devices," in *Proc. IEEE Conf. Comput. Commun.*, May 2021, pp. 1–10.

[41] W. Zhang, K. Song, X. Rong, and Y. Li, "Coarse-to-fine UAV target tracking with deep reinforcement learning," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1522–1530, Oct. 2019.

[42] J. Zhang *et al.*, "Robust RFID based 6-DoF localization for unmanned aerial vehicles," *IEEE Access*, vol. 7, pp. 77348–77361, 2019.

[43] Q. Zhang, W. Xu, W. Liang, J. Peng, T. Liu, and T. Wang, "An improved algorithm for dispatching the minimum number of electric charging vehicles for wireless sensor networks," *Wireless Netw.*, vol. 25, no. 3, pp. 2689–2705, Dec. 2019.

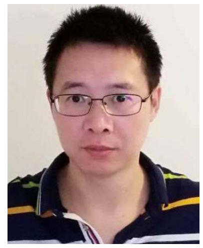
Wenzheng Xu (Member, IEEE) received the B.Sc., M.E., and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2008, 2010, and 2015, respectively. He was a Visitor at both The Australian National University and The Chinese University of Hong Kong. He is currently an Associate Professor at Sichuan University. His research interests include wireless *ad-hoc* and sensor networks, mobile computing, approximation algorithms, combinatorial optimization, online social networks, and graph theory.

Tao Xiao received the B.Sc. degree in communication engineering from the Chongqing University of Posts and Telecommunications, China, in 2020. He is currently pursuing the master's degree in electronic and information engineering with Sichuan University. His current research interest is UAV scheduling.

Junqi Zhang received the B.Sc. and M.E. degrees in computer science from Sichuan University, China, in 2018 and 2021, respectively. His research interests include UAV scheduling and wireless sensor networks.

Weifa Liang (Senior Member, IEEE) received the B.Sc. degree in computer science from Wuhan University, China, in 1984, the M.E. degree in computer science from the University of Science and Technology of China in 1989, and the Ph.D. degree in computer science from The Australian National University in 1998. He is currently a Professor with the Department of Computer Science, City University of Hong Kong. Prior to the current position, he was a Professor at The Australian National University. His research interests include design and analysis of energy efficient routing protocols for wireless *ad-hoc* and sensor networks, the Internet of Things, edge and cloud computing, network function virtualization and software-defined networking, design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory. He serves as an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS.

Zichuan Xu (Member, IEEE) received the B.Sc. and M.E. degrees in computer science from the Dalian University of Technology, China, in 2008 and 2011, and the Ph.D. degree in computer science from The Australian National University in 2016. He was a Research Associate at University College London. He is currently an Associate Professor with the School of Software, Dalian University of Technology. His research interests include cloud computing, software-defined networking, wireless sensor networks, algorithmic game theory, and optimization problems.



Xuxun Liu received the Ph.D. degree in communication and information systems from Wuhan University, Wuhan, China, in 2007. He was a Visiting Professor of electrical engineering and computer sciences at the University of California, Berkeley, USA, in 2017. He is currently an Associate Professor with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China. He has authored or coauthored over 40 scientific articles in international journals and conference proceedings. His current research interests include wireless sensor networks, vehicle-to-vehicle communications, mobile computing, and edge computing. He was a recipient of the National Innovation Award of Industry-University-Research Collaboration in 2017. He serves as the workshop chair, the publication chair, or a TPC member for a number of conferences. He serves as an Editorial Board Member for *Wireless Networks* (Springer) and an Associate Editor for IEEE ACCESS.

Xiaohua Jia (Fellow, IEEE) received the B.Sc. and M.Eng. degrees from the University of Science and Technology of China in 1984 and 1987, respectively, and the D.Sc. degree in information science from The University of Tokyo in 1991. He is currently a Chair Professor with the Department of Computer Science, City University of Hong Kong. His research interests include cloud computing and distributed systems, computer networks, wireless sensor networks, and mobile wireless networks. He is the General Chair of ACM MobiHoc 2008, the TPC Co-Chair of IEEE MASS 2009, the Area Chair of IEEE INFOCOM 2010, the TPC Co-Chair of IEEE GLOBECOM 2010 and Ad Hoc and Sensor Networking Symposium, and the Panel Co-Chair of IEEE INFOCOM 2011. He was an Editor of IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS from 2006 to 2009. He is an Editor of *World Wide Web*, *Wireless Networks*, *Journal of Combinatorial Optimization*, and so on.

Sajal K. Das (Fellow, IEEE) is currently the Chair of the Computer Science Department and the Daniel St. Clair Endowed Chair at the Missouri University of Science and Technology, USA. He has directed numerous funded projects in these areas totaling over $15M and published extensively with more than 600 research articles in high quality journals and refereed conference proceedings. His current research interests include theory and practice of wireless sensor networks, big data, cyber-physical systems, smart healthcare, distributed and cloud computing, security and privacy, biological and social networks, applied graph theory, and game theory. He is a Co-Founder of the IEEE PerCom Conference, IEEE WoWMoM Conference, and ICDCN Conference. He served on numerous conference committees as the general chair, the program chair, or a program committee member. He serves as the Founding Editor-in-Chief for the *Pervasive and Mobile Computing* journal and an Associate Editor for IEEE TRANSACTIONS ON MOBILE COMPUTING and *ACM Transactions on Sensor Networks*.