Minimizing the Longest Tour Time Among a Fleet of UAVs for Disaster Area Surveillance

Qing Guo, Jian Peng, Wenzheng Xu, *Member, IEEE*, Weifa Liang, *Senior Member, IEEE*, Xiaohua Jia, *Fellow, IEEE*, Zichuan Xu, Yanbin Yang, and Minghui Wang

Abstract—In this paper, we study the employment of multiple Unmanned Aerial Vehicles (UAVs) to monitor Points of Interests (PoIs) in a disaster area, e.g., collapsed buildings after an earthquake, where the UAVs can take photos and videos for the people trapped at PoIs, because such valuable information is imperative to make rescue decisions. Unlike most existing studies that ignored the monitoring time of PoIs and simply minimized the longest flying distance among the UAVs, we observe that it takes time to monitor the PoIs. Then, it is possible that the flying distance of a UAV in its flying tour may not be too long, the tour however contains many densely-located PoIs. Therefore, it will take a very long time for the UAV to monitor the PoIs in its tour. In this paper, we first formulate a problem of finding flying tours for K given UAVs to collaboratively monitor PoIs in a disaster area, such that the maximum spent time of the K UAVs among their tours is minimized, where the spent time of a UAV in its tour consists of the flying time and the PoI monitoring time. We then propose a novel $5\frac{1}{3}$-approximation algorithm for the problem, improving the best approximation ratio 6 so far for the problem of minimizing the longest flying distance among the UAVs. In addition, we extend the proposed algorithm to the case that each UAV may not be able to monitor all PoIs assigned to it, due to its limited maximum flying time (e.g., 30 minutes), and the UAV must return to its depot to replace its battery. We finally evaluate the performance of the proposed algorithms via simulation environments, and experimental results show that the proposed algorithms are very promising. Especially, the maximum spent times of the K UAVs in their tours by the proposed algorithms are up to 30 percent shorter than those by existing algorithms. In addition, the empirical approximation ratios of the proposed algorithms are no more than 2.4, which are much smaller than their theoretical approximation ratios that are at least $5\frac{1}{3}$.

Index Terms—Disaster area monitoring with UAVs, flying tour scheduling, maximum tour time minimization, approximation algorithms

1 Introduction

According to a recent survey in [20], a variety of geophysical, meteorological, hydrological and climatological disasters, e.g., earthquakes, tsunamis or flooding, have incurred more than tens of thousands of fatalities and a high loss of hundreds of million dollars per year in the past decades. For example, the devastating bush fires have spread across Australia for months from 2019 to 2020. Until Jan. 8, 2020, millions of acres have burned, and the fires have killed at least 25 people, 5 hundred million animals, including 8,000 koalas [12]. When a disaster occurs, it is very imperative to obtain the most recent information for aiding search and rescue activities, so that survivals can be evacuated as quickly as possible. However, communication and transportation infrastructures may be damaged and not function any more after the disaster. Even worse, it may be very dangerous for rescuers to approach the disaster area.

Low-cost, flexible UAVs are emerging as a promising method for obtaining critical information about disaster areas, by taking photos and videos for PoIs (e.g., collapsed buildings, malls and schools) in the disaster area with their onboard lightweight cameras and/or thermal infrared imagers, and they send these invaluable information back to rescue stations [2], [9], [11], [17], [27]. For example, UAVs equipped with cameras were employed for conducting damage-assessment in the wake of natural disasters of Hurricanes Florence and Michael in 2018 [8].

It is very important to monitor PoIs in disaster areas as quickly as possible such that the critical information about the people trapped at PoIs can be collected in time for aiding rescue activities [16]. Therefore, scheduling UAVs to monitor PoIs in disaster areas has attracted lots of attentions. Most existing studies focused on balancing the flying distances of different UAVs to monitor PoIs so that the flying distance of a UAV will not be too much longer than those of other UAVs [14], [19], [24], [28], [36], [38]. For example, Scott *et al.* [24] found flying tours for multiple UAVs to cover PoIs in a geometrically complex area, so as to minimize the maximum flying distance among the UAVs. Modares *et al.* [19] recognized that the energy consumed by a UAV is related to not only its flying distance, but also the number of turns in its flying tour, and they employed multiple UAVs to monitor PoIs so that the maximum energy

- *Qing Guo, Jian Peng, Wenzheng Xu, Yanbing Yang, and Minghui Wang are with the College of Computer Science, Sichuan University, Chengdu 610065, P. R. China. E-mail: guoqing1@stu.scu.edu.cn, {jianpeng, yangyanbing, wangminghui}@scu.edu.cn, wenzheng.xu3@gmail.com.*
- *Weifa Liang is with the Research School of Computer Science, The Australian National University, Canberra, ACT 2601, Australia. E-mail: wliang@cs.anu.edu.au.*
- *Xiaohua Jia is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. E-mail: csjia@cityu.edu.hk.*
- *Zichuan Xu is with the School of Software, Dalian University of Technology, Dalian 116024, P. R. China. E-mail: z.xu@dlut.edu.cn.*

Manuscript received 2 June 2020; revised 12 Oct. 2020; accepted 11 Nov. 2020. Date of publication 16 Nov. 2020; date of current version 3 June 2022.
(Corresponding author: Wenzheng Xu.)
Digital Object Identifier no. 10.1109/TMC.2020.3038156

(a) Scheduling two UAVs to monitor PoIs so that their longest flying time/distance is minimized.

(b) Scheduling two UAVs to monitor PoIs so that the maximum tour time is minimized, where the time spent in a tour consists of the UAV flying time and PoI monitoring time.

Fig. 1. An illustration of dispatching two UAVs to monitor PoIs in a disaster area, where the monitoring time of each PoI is 10 seconds and the value next to each edge between two PoIs is the UAV flying time between them.

consumption of the UAVs in their flying tours is minimized, by proposing a Lin-Kernighan heuristic algorithm. Zhan *et al.* [38] investigated a problem of finding flying tours for multiple UAVs to collect data from sensors so that the maximum spent time by the UAVs is minimized, by devising a genetic algorithm. Kim *et al.* [14] proposed an 8-approximation algorithm for a problem of dispatching UAVs to collect information of PoIs in search and rescue applications, such that the longest flying time among the UAVs is minimized. Xu *et al.* [28] devised a 7-approximation algorithm for the min-max cycle cover problem of finding K rooted tours to cover nodes so that the longest length of the K tours is minimized, while Yu *et al.* [36] recently improved the approximation ratio to 6. Fig. 1a shows the flying tours of two UAVs delivered by existing algorithms so that their maximum flying distance/time is minimized.

We notice that it takes time to monitor each PoI, and the times spent for monitoring different PoIs may vary significantly, as UAVs need to monitor accurate information about every PoI and many factors will influence the monitoring time of a PoI, e.g., the number of people trapped at a PoI and obstacles nearby [16]. Although the flying distance of a UAV in its flying tour found by existing algorithms may not be long, the tour may contain many PoIs, while the monitoring of them takes time too. For example, it can be seen from Fig. 1a that the flying times of the two UAVs in tours C_1' and C_2' are approximately balanced, where the flying time of the UAV in tour C_1' is $15\ s \times 6 + 25\ s \times 3 = 165\ s$, and the flying time in tour C_2' is $40\ s \times 4 = 160\ s$. However, the flying tour C_1' in Fig. 1a contains 8 PoIs while C_2' contains only 3 PoIs. Then, the total time spent by the UAV in C_1' is much longer than that in C_2', i.e., $165\ s + 10\ s \times 8 = 245\ s > 160\ s + 10\ s \times 3 = 190\ s$, where the monitoring time of each PoI is assumed to be 10 seconds. However, we need to balance the times spent at different tours, so that the monitored information about PoIs can be sent to rescue teams as quickly as possible. Otherwise, the people trapped at a PoI in tour C_1' of Fig. 1a may wait for a long time before they are monitored by a UAV, thereby incurring more casualties.

Unlike most existing algorithms that ignored the monitoring time of PoIs, in this paper, we consider not only the flying time but also the monitoring time of PoIs in a tour. We study a scheduling problem of finding flying tours for multiple UAVs to collaboratively monitor PoIs in a disaster area, such that the maximum spent time of the UAVs in their tours is minimized, where the spent time of a UAV in its tour consists of both the flying time and the monitoring time of PoIs. For example, Fig. 1b shows that the spent times of the two UAVs in their tours are almost balanced, when we take the monitoring time of PoIs into consideration, where the spent time in tour C_1 is $(15\ s \times 6 + 65\ s) + 10\ s \times 6 = 215\ s$, and the spent time in tour C_2 is $(30 + 15 + 15 + 30 + 40 + 40) + 10 \times 5 = 220\ s$.

The UAV scheduling problem poses many challenges, including: (i) how to assign PoIs to different UAVs such that each UAV will not be assigned too many PoIs; (ii) how to find flying tours for the UAVs so that the maximum spent time of their tours is minimized; and (iii) how to ensure that each tour of a UAV contains a depot, so that the UAV departs from the depot and returns to the depot for replacing its battery, after it finishes the monitoring tasks of the PoIs assigned to it. In this paper, we address these challenges by proposing a novel approximation algorithm for the UAV scheduling problem.

The novelties of this paper lie in that we take not only the flying time of UAVs but also the monitoring time of PoIs into consideration. We propose a performance guaranteed $5\frac{1}{3}$-approximation algorithm for the UAV scheduling problem when ignoring the limited flying time of each UAV, such that the maximum spent time of UAVs in their flying tours is minimized, which improves the currently best approximation ratio 6 in [36].

The main contributions of this paper are summarized as follows.

- We formulate a multi-rooted monitoring time minimization problem without considering the limited flying time of each UAV, which is to find K rooted flying tours for K UAVs to monitor PoIs in a disaster area, such that the maximum spent time among the K tours is minimized, where the spent time in a tour consists of both the UAV flying time and the PoI monitoring time, and different tours contains different UAV depots.

- We propose a novel $5\frac{1}{3}$-approximation algorithm for the problem, which has a better approximation ratio than the state-of-the-art with a ratio of 6 in [36].
- We also extend the proposed algorithm to the case where the maximum flying time of each UAV is limited, and the UAV may not be able to monitor all PoIs assigned to it in a single tour, then the UAV needs to replace its battery and monitor the PoIs in multiple tours. Thus, the spent time of each tour by the UAV is no greater than its maximum flying time.
- We evaluate the performance of the proposed algorithms via simulation environments, and experimental results show that the proposed algorithms are very promising. Especially, the maximum spent times of the tours delivered by the proposed algorithms are up to 30 percent shorter than those by existing algorithms. In addition, the empirical approximation ratios of the proposed algorithms are no more than 2.4, which are much less than their theoretical approximation ratios that are at least $5\frac{1}{3}$.

The rest of this paper is organized as follows. Section 2 introduces the network model and defines the problems. Section 3 proposes an algorithm for a subproblem of the multi-rooted monitoring time minimization problem, which serves as a subroutine of the proposed approximation algorithm in Section 4 for the multi-rooted problem. Section 5 extends to the case with limited flying time constraint on each UAV. Section 6 evaluates the performance of the proposed algorithm through extensive simulation experiments. Section 7 reviews related work. Finally, Section 8 concludes this paper.

2 Preliminaries

In this section, we first introduce the network model, then define the problem precisely.

2.1 Network Model

In this paper, we study the employment of multiple UAVs equipped with cameras and/or thermal infrared imagers, e.g., DJI Cinema Color System ZENMUSE X7 [7], to monitor PoIs (e.g., collapsed buildings, malls and schools) in a disaster area [24], [25], [26], by taking photos/videos for them and sending these valuable information back to rescue stations.

We treat a disaster area as a three dimensional euclidean space, in which there are n to-be-monitored PoIs $v_1, v_2, \ldots, v_n$, where each PoI represents a building, a school, a hospital, or a factory, and there may be many people trapped at each of the PoIs. Denote by (x_i, y_i, z_i) the coordinate of a PoI v_i with $1 \le i \le n$, where z_i is the altitude of v_i, as a UAV needs to monitor the PoI at a close hovering location, e.g., within a few meters. Notice that the disaster area may be very large. In order to collect the information about the n PoIs as quickly as possible, we assume that multiple UAVs are employed to collaboratively monitor the PoIs. Specially, assume that there are K UAVs available with $K \ge 1$, and each UAV k is located at a depot r_k initially. Let R be the set of the K depots of the UAVs, i.e., $R = \{r_1, r_2, \ldots, r_K\}$. Notice that some depots may be co-located.

We use a complete graph $G_r = (V \cup R, E_r)$ to model the UAV network, where V is the set of PoIs, i.e., $V = \{v_1, v_2, \ldots, v_n\}$, R is the set of UAV depots, and there is an edge in E_r between any two nodes in $V \cup R$.

To collaboratively monitor the n PoIs in V, we need to partition set V into K *disjoint* subsets $V_1, V_2, \ldots, V_K$, and UAV k monitors the PoIs in V_k with $1 \le k \le K$. It must be mentioned that no PoI will be monitored by two or more UAVs. For each UAV k, its flying tour can be represented as $C_k = r_k \rightarrow v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_{n_k} \rightarrow r_k$, which means that UAV k takes off from depot r_k, flies to the location of PoI v_1 and monitors v_1 by taking photos/videos, flies to monitor v_2, ..., monitors v_{n_k}, and finally returns to depot r_k, where $n_k = |V_k|$. For example, Fig. 1b shows two flying tours C_1 and C_2 of $K = 2$ UAVs.

The spent time of each flying tour C_k consists of the flying time between PoIs and the monitoring time of PoIs in V_k. Denote by $c(v_i, v_j)$ the flying time from PoIs v_i to v_j, which can be calculated as $c(v_i, v_j) = \frac{d(v_i, v_j)}{s}$, where $d(v_i, v_j)$ is the euclidean distance between v_i and v_j, and s is the maximum flying speed of a UAV. On the other hand, denote by $h(v_i)$ the monitoring time of a PoI v_i. The spent time of a UAV k in its flying tour C_k then is

$$w(C_k) = \sum_{i=0}^{n_k} c(v_i, v_{i+1}) + \sum_{i=1}^{n_k} h(v_i), \quad 1 \le k \le K, \tag{1}$$

where $v_0 = v_{n_k+1} = r_k$.

2.2 Problem Definition

It is imperative to quickly collect the information about the PoIs in the disaster area for aiding search and rescue operations. Therefore, in this paper, we consider a novel *multi-rooted monitoring time minimization problem*, which is to find flying tours $C_1, C_2, \ldots, C_K$ for K UAVs, such that the maximum tour time among the K tours, i.e., $\max_{k=1}^{K}\{w(C_k)\}$, is minimized. In other words, the tour times of the K UAVs must be approximately balanced. Otherwise, if the tour time of a UAV is much longer than those of other UAVs, the waiting time of some PoI before its monitoring by the UAV may be prohibitively long, some people trapped at the PoI may have already died when the PoI is monitored.

We formally define the problem as follows. A binary variable x_{ik} indicates whether a node v_i is contained in a flying tour C_k, i.e., $x_{ik} = 1$ if node v_i is contained in C_k; otherwise, $x_{ik} = 0$, where $1 \le i \le n + K$, nodes $v_{n+1}, v_{n+2}, \ldots, v_{n+K}$ represent depots $r_1, r_2, \ldots, r_K$, respectively, and $1 \le k \le K$. Another binary variable y_{ijk} is used to indicate whether an edge (v_i, v_j) from nodes v_i to v_j is contained in C_k, i.e., $y_{ijk} = 1$ if the edge (v_i, v_j) is in C_k; otherwise, $y_{ijk} = 0$, where $1 \le i, j \le n + K$ and $1 \le k \le K$. The problem can be formulated as

$$\min_{x_{ik}, y_{ijk}} \quad \max_{k=1}^{K}\{w(C_k)\}, \tag{2}$$

subject to

$$w(C_k) = \sum_{i=1}^{n} x_{ik} \cdot h(v_i) + \sum_{i=1}^{n+K} \sum_{j=1, j \ne i}^{n+K} y_{ijk} \cdot c(v_i, v_j) \qquad 1 \le k \le K \tag{3}$$

$$\sum_{j=1}^{n} y_{n+k,j,k} = \sum_{j=1}^{n} y_{j,n+k,k} = x_{n+k,k} = 1, \quad 1 \leq k \leq K \tag{4}$$

$$\sum_{j=1,j\neq i}^{n+K} y_{ijk} = \sum_{j=1,j\neq i}^{n+K} y_{jik} = x_{ik},\ 1 \leq i \leq n, 1 \leq k \leq K \tag{5}$$

$$\sum_{k=1}^{K} x_{ik} = 1, \qquad 1 \leq i \leq n \tag{6}$$

$$\sum_{v_i,v_j\in S} y_{ijk} \leq |S| - 1,\ 1 \leq k \leq K,\ \forall S \subseteq V \cup R \setminus \{r_k\}, \quad S \neq \emptyset \tag{7}$$

$$x_{ik} \in \{0,1\}, \qquad 1 \leq i \leq n+K,\ 1 \leq k \leq K \tag{8}$$

$$y_{ijk} \in \{0,1\}, \qquad 1 \leq i,\ j \leq n+K, 1 \leq k \leq K, \tag{9}$$

where Constraints (3) calculates the consumed time of UAV k in its flying tour C_k. Constraints (4) (5) (6) and (7) collaboratively ensure that tours $C_1, C_2, \ldots, C_K$ form a feasible flying schedule. Specifically, Constraints (4) requires that UAV k must take off from depot r_k (i.e., node v_{n+k}) and return to r_k after monitoring the PoIs in tour C_k. Constraints (5) indicates that there is exactly one incoming edge to v_i and one outgoing edge from v_i if v_i is contained in tour C_k. Constraints (6) implies that each PoI v_i should be visited once and only once. Constraints (7) indicates that the number of edges with their endpoints contained in any nonempty proper subset S of $V \cup (R \setminus \{r_k\})$ is no greater than $|S| - 1$, thus prevents solutions consisting of several disconnected closed tours.

In this paper, we also consider a *rootless* monitoring time minimization problem, where 'rootless' means that the flying tour C_k of each UAV k does not contain any depot r_k. The rationale behind is that, an algorithm for this rootless problem will be served as a *subroutine* of the proposed algorithm for the multi-rooted problem. Specifically, given n to-be-monitored PoIs $v_1, v_2, \ldots, v_n$ in V, *the rootless monitoring time minimization problem* is to find K flying tours $C_1, C_2, \ldots, C_K$ to monitor the n PoIs in V such that the maximum tour time among the K tours, i.e., $\max_{k=1}^{K}\{w(C_k)\}$, is minimized, where each tour does not contain any depot.

2.3 Approximation Ratio

It can be seen that the multi-rooted monitoring time minimization problem is NP-hard, as the well-known NP-hard Traveling Salesman Problem (TSP) is its special case with $K = 1$. Then, it is unlikely to find an exact solution to the problem within a polynomial time unless P=NP. In this paper, we propose a $5\frac{1}{3}$-approximation algorithm for the problem in a polynomial time, where the solution delivered by the proposed algorithm is no more than $5\frac{1}{3}$ times of the optimal solution to the problem.

Given a minimization problem, denote by OPT the value of an optimal solution to the problem. Also, denote by SOL the value of a feasible solution to the problem, which is delivered by a polynomial time algorithm A. Then, the approximation ratio of algorithm A is α if $SOL \leq \alpha \cdot OPT$, where $\alpha \geq 1$. It can be seen that, the smaller the approximation ratio α is, the better the solution SOL is.

3 Approximation Algorithm for the Rootless Monitoring Time Minimization Problem

In this section, we propose a $4\frac{1}{3}$-approximation algorithm for the rootless monitoring time minimization problem.

3.1 Basic Idea of the Proposed Algorithm

Given a network $G = (V, E; h : V \rightarrow \mathbb{Z}^{\geq 0}, c : E \rightarrow \mathbb{Z}^{\geq 0})$ with n to-be-monitored PoIs in V, assume that an optimal solution to the rootless problem consists of K tours $C_1^*, C_2^*, \ldots, C_K^*$. Denote by OPT the value of the optimal solution, i.e., $OPT = \max_{k=1}^{K}\{w(C_k^*)\}$.

The basic idea behind the proposed algorithm is that, given a guess B of OPT with $B \geq OPT$, it finds no more than K tours $C_1, C_2, \ldots, C_K$ with their maximum tour time $4\frac{1}{3}B$. It then finds a $4\frac{1}{3}$-approximation solution through a binary search for OPT.

In the following, we first assume that the K optimal tours $C_1^*, C_2^*, \ldots, C_K^*$ of the rootless problem are given. Under this assumption, we show that we are able to find no more than K tours $C_1, C_2, \ldots, C_K$ with their maximum tour time $4\frac{1}{3}B$ if $B \geq OPT$. We later remove this assumption and show that such K tours can still be found.

3.2 The Algorithm With Given K Optimal Tours

Given a network $G = (V, E; h : V \rightarrow \mathbb{Z}^{\geq 0}, c : E \rightarrow \mathbb{Z}^{\geq 0})$ and a guess B of OPT with $B \geq OPT$, the algorithm constructs an auxiliary graph $G' = (V, E; w' : E \rightarrow \mathbb{Z}^{\geq 0})$ from G, and the weight of each edge (v_i, v_j) in E is

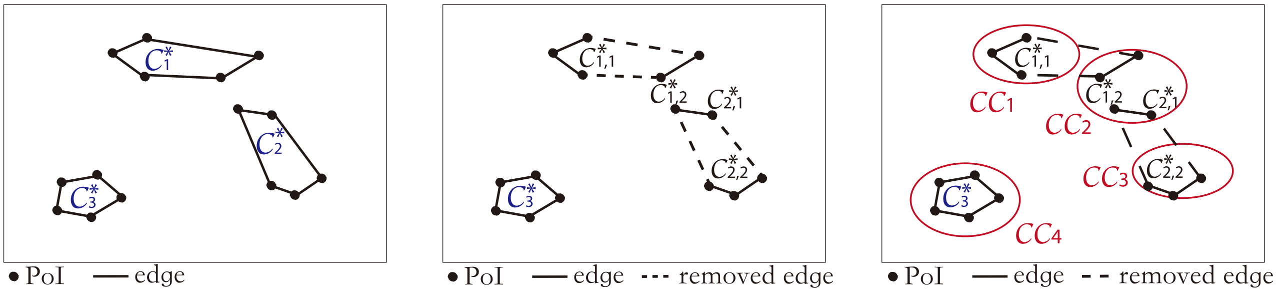
$$w'(v_i, v_j) = c(v_i, v_j) + \frac{h(v_i) + h(v_j)}{2}, \tag{10}$$

where $c(v_i, v_j)$ is the flying time between PoIs v_i and v_j in V, $h(v_i)$ and $h(v_j)$ are the monitoring times of v_i and v_j, respectively. We later show that the optimal values for the rootless monitoring time minimization problem in G and G' are equal. Then, an α-approximation solution to the problem in G' returns an α-approximation solution to the problem in G, where α is a constant with $\alpha \geq 1$.

Before we proceed, we introduce an important lemma, which is the cornerstone of the proposed algorithm.

Lemma 1 [28]. *Given any closed tour C in graph G', assume that the weight $w'(C)$ of tour C is no more than B, i.e., $w'(C) \leq B$. Then, (i) the weight of each edge in C is no more than $\frac{B}{2}$; and (ii) there are no more than two edges in C with their edge weights strictly greater than $\frac{B}{3}$.*

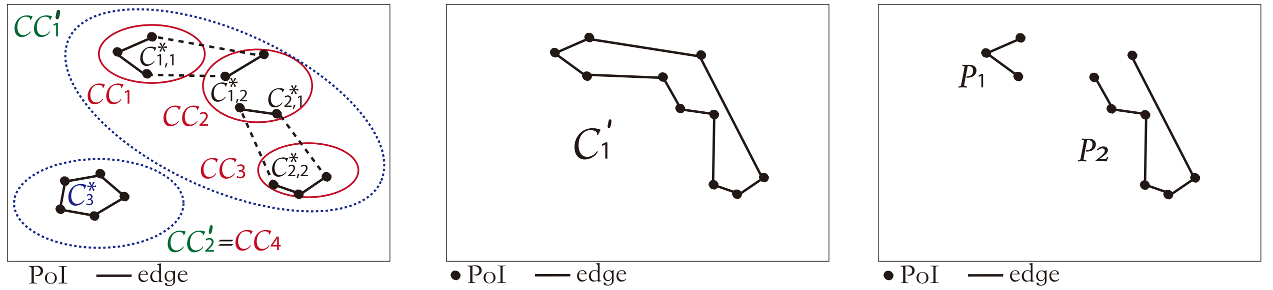
Recall that the K optimal tours $C_1^*, C_2^*, \ldots, C_K^*$ are given. For example, Fig. 2a shows $K = 3$ optimal tours. The algorithm first removes the edges with their weights strictly greater than $\frac{B}{3}$ from graph G'. Assume that there are p connected components $CC_1, CC_2, \ldots, CC_p$ in the residual graph after the edge removals, where p is a positive integer.



(a) The $K = 3$ optimal tours C_1^*, C_2^*, and C_3^*.

(b) The $K = 3$ optimal tours C_1^*, C_2^*, and C_3^* are split into segments $C_{1,1}^*$, $C_{1,2}^*$, $C_{2,1}^*$, $C_{2,2}^*$ and C_3^* after the edge removals with weights strictly greater than $\frac{B}{3}$.

(c) There are $p = 4$ components CC_1, CC_2, CC_3 and CC_4 in the residual graph after the edge removals with weights strictly greater than $\frac{B}{3}$ from graph G'.



(d) $p' = 2$ connected components CC_1' and CC_2' are merged from the $p = 4$ connected components CC_1, CC_2, CC_3, CC_4.

(e) A tour C_1' that covers all nodes in CC_1' is found, by invoking `Christofides' algorithm`.

(f) The tour C_1' is split into $k_1 = 2$ paths P_1 and P_2 with the maximum path weight $2\frac{1}{6}B$.

Fig. 2. An illustration of the algorithm for the rootless monitoring time minimization problem when the K optimal tours are given (notice that some edges in CC_1, CC_2, CC_3 and CC_4 are not shown).

Following Lemma 1, each optimal tour C_k^* is split into no more than two segments $C_{k,1}^*$ and $C_{k,2}^*$ after the edge removals, since there are no more than two edges in C_k^* with their weights strictly greater than $\frac{B}{3}$. For example, Fig. 2b shows that C_1^* is split into two segments $C_{1,1}^*$ and $C_{1,2}^*$ after the edge removals with weights strictly greater than $\frac{B}{3}$, C_2^* also is split into two segments $C_{2,1}^*$ and $C_{2,2}^*$, while no edges in C_3^* are removed. Assume that $C_{k,1}^*$ and $C_{k,2}^*$ lie in two connected components CC_i and CC_j, respectively. For example, Fig. 2c shows that segments $C_{1,1}^*$ and $C_{1,2}^*$ are in connected components CC_1 and CC_2, respectively, and segments $C_{2,1}^*$ and $C_{2,2}^*$ are in connected components CC_2 and CC_3, respectively. It must be mentioned that some edges in CC_1, CC_2, CC_3 and CC_4 are not shown in Fig. 2.

The algorithm then merges the p connected components $CC_1, CC_2, \ldots, CC_p$ with the knowledge of K optimal tours C_1^*, C_2^*, $\ldots$, C_K^*. For each optimal tour C_k^* with $1 \le k \le K$, if C_k^* was split into two segments $C_{k,1}^*$ and $C_{k,2}^*$ after the edge removals and they lie in two different connected components CC_i and CC_j, the algorithm merges the two connected components CC_i and CC_j with their nearest edge. For example, Fig. 2d shows that CC_1 and CC_2 are first merged, since the two segments $C_{1,1}^*$ and $C_{1,2}^*$ of C_1^* lie in CC_1 and CC_2, respectively, followed by merging CC_2 and CC_3 as $C_{2,1}^*$ and $C_{2,2}^*$ are in CC_2 and CC_3, respectively. Assume that, in the end, the p connected components $CC_1, CC_2, \ldots, CC_p$ are merged into p' $(\le p)$ connected components CC_1', CC_2', $\ldots$, $CC_{p'}'$. For example, Fig. 2d shows that the $p = 4$ connected components CC_1, CC_2, CC_3, CC_4 are merged into $p' = 2$ connected components CC_1' and CC_2'.

For each connected component CC_i' with $1 \le i \le p'$, the algorithm finds a closed tour C_i' for visiting nodes in CC_i', by invoking `Christofides' algorithm` [5], see Fig. 2e. The tour C_i' then is split into k_i paths $P_1, P_2, \ldots, P_{k_i}$ with the maximum path weight $2\frac{1}{6}B$, where $k_i = \lceil \frac{w'(C_i')}{2\frac{1}{6}B} \rceil$ [1], [10], [34], see Fig. 2f.

It can be seen that $\sum_{i=1}^{p'} k_i$ paths are obtained from the p' connected components, where the weight of each path is no greater than $2\frac{1}{6}B$. We later show that the number of obtained paths is no more than K, i.e., $\sum_{i=1}^{p'} k_i \le K$. Then, no more than K tours C_1, C_2, $\ldots$, C_K can be obtained, where C_j is derived from path P_j by connecting its two endnodes. It can be seen that the weight $w'(C_j)$ of each obtained tour C_j is no more than $2 \cdot 2\frac{1}{6}B = 4\frac{1}{3}B$, since edge weights in graph G' satisfy the triangle inequality.

3.3 The Algorithm Without the Knowledge of the Optimal Tours

In the previous subsection, we assumed that the optimal K tours C_1^*, C_2^*, $\ldots$, C_K^* are given. Under this assumption, we showed that we can find no more than K tours C_1, C_2, $\ldots$, C_K with their maximum tour time $4\frac{1}{3}B$, if the guess B is at least the optimal value OPT, i.e., $B \ge OPT$. We now remove this assumption, and show that we are still able to find such K tours $C_1, C_2, \ldots, C_K$ with their maximum tour time $4\frac{1}{3}B$ if $B \ge OPT$.

Given a network $G = (V, E; h : V \rightarrow \mathbb{Z}^{\ge 0}, c : E \rightarrow \mathbb{Z}^{\ge 0})$, similar to the algorithm with given optimal tours, we first construct an auxiliary complete graph $G' = (V, E; w' : E \rightarrow \mathbb{Z}^{\ge 0})$ from G. We then remove the edges with their edge weights strictly greater than $\frac{B}{3}$ from graph G'. Also, assume that there are p connected components $CC_1, CC_2, \ldots, CC_p$ in the residual graph after the edge removals, where p is a positive integer, see Fig. 2(c). However, there is a problem about how to merge the p connected components, since the optimal K tours C_1^*, C_2^*, ..., C_K^* now are not given.

Algorithm 1. Algorithm for the Rootless Monitoring Time Minimization Problem

Input: A network $G = (V, E; h : V \rightarrow \mathbb{Z}^{\geq 0}, c : E \rightarrow \mathbb{Z}^{\geq 0})$, the monitoring time $h(v_i)$ of each PoI v_i, the UAV flying speed s.
Output: a set $\mathcal{C}$ of K flying tours $C_1, C_2, \ldots, C_K$.

```
Construct an auxiliary graph G' = (V, E; w' : E → Z^{≥0}) from G, where w'(v_i, v_j) = c(v_i, v_j) + (h(v_i)+h(v_j))/2;
/* B_l and B_u are the lower and upper bounds on the optimal value OPT, respectively */
Let B_l = 1, B_u = w'(C), where w'(C) is the weight of the tour C that visits all nodes in G', by invoking Christofides' algorithm;
Let C ← ∅; /* the set of obtained K tours.*/
while B_l + 1 < B_u do
  Let B = ⌊(B_l+B_u)/2⌋; /* B is a guess of OPT. */
  Remove the edges with weights greater than B/3 in graph G'. Assume that there are p connected components CC_1, CC_2, ..., CC_p in the residual graph after the edge removals;
  Let C_B ← ∅; /* The set of obtained tours with guess B */
  for j ← 1 to p^{2K-1} do
    /* the jth combination */
    Merge the p connected components CC_1, CC_2, ..., CC_p into p_j connected components CC'_1, CC'_2, ..., CC'_{p_j} for the jth combination, and obtain a set C_j of rootless tours;
    if |C_j| ≤ K then
      Let C_B ← C_j;
    end if
  end for
  if |C_B| ≤ K and C_B ≠ ∅ then
    Let B_u ← B; /* the guess B is larger than OPT */
    Let C ← C_B;
  else
    Let B_l ← B; /* the guess B is smaller than OPT */
  end if
end while
return C.
```

We detour the problem by considering all combinations derived from the K optimal tours. For each optimal tour C_k^*, since it is split into at most two segments $C_{k,1}^*$ and $C_{k,2}^*$ after the edge removals, $C_{k,1}^*$ (or $C_{k,2}^*$) is in one of the p connected components CC_1, CC_2, ..., CC_p. Without loss of generality, we assume that the first segment $C_{1,1}^*$ of the optimal tour C_1^* is in CC_1. Therefore, there are no more than p^{2K-1} different combinations, and the combination derived from the K optimal tours C_1^*, C_2^*, ..., C_K^* must be one of them. Also, since each optimal tour C_k^* is split into no more than two segments $C_{k,1}^*$ and $C_{k,2}^*$, the number p of the connected components is no more than $2K$, i.e., $p \leq 2K$. Then, we have $p^{2K-1} \leq (2K)^{2K-1}$. Notice that the number K of UAVs in real applications for disaster monitoring usually is very small, e.g., $K = 5$ [3], [14]. Then, K can be considered as a constant.

For the jth combination of the p^{2K-1} combinations, similarly to the algorithm with given optimal tours, we can merge the connected components CC_1, CC_2, ..., CC_p, e.g., see Fig. 2d, and find k_j tours $C_1, C_2, \ldots, C_{k_j}$, where k_j is the number of obtained tours, and $1 \leq j \leq p^{2K-1}$.

It can be seen that the minimum number $\min_{1 \leq j \leq p^{2K-1}}\{k_j\}$ is no greater than K, since the combination derived from the optimal K tours is one of the p^{2K-1} combinations and no more than K tours are delivered from the optimal combination by Lemma 3.

We note that the number of combinations p^{2K-1} may be large, and we propose a novel strategy to reduce the number of potential combinations. Assume that the first segment $C_{k,1}^*$ of an optimal tour C_k^* is in a connected component CC_i. Denote by $w'(CC_i, CC_j)$ the nearest distance between connected components CC_i and CC_j, i.e., $w'(CC_i, CC_j) = \min_{\forall u \in CC_i, \forall v \in CC_j}\{w'(u, v)\}$, where the nodes u and v are in CC_i and CC_j, respectively. Let $N(CC_i)$ be the set of connected components such that the nearest distance between each CC_j in $N(CC_i)$ and CC_i is no more than $\frac{B}{2}$, i.e., $N(CC_i) = \{CC_j \mid 1 \leq j \leq p,\ w'(CC_i, CC_j) \leq \frac{B}{2}\}$. Then, it can be seen that the second segment $C_{k,2}^*$ of the optimal tour C_k^* must be in one of the connected components in $N(CC_i)$, as there are no edges in C_k^* with their weights larger than $\frac{B}{2}$, following Lemma 1. Notice that the number $|N(CC_i)|$ of connected components $N(CC_i)$ may be much smaller than the total number p of the connected components. By doing so, we reduce the number of the combinations. In addition, our later experimental results show that the running time of the proposed algorithm by adopting this strategy is very short.

The algorithm for the rootless monitoring time minimization problem is presented in Algorithm 1.

3.4 Algorithm Analysis

We start with the following lemma.

Lemma 2. *Given a network $G = (V, E; h : V \rightarrow \mathbb{Z}^{\geq 0}, c : E \rightarrow \mathbb{Z}^{\geq 0})$, an auxiliary graph $G' = (V, E; w' : E \rightarrow \mathbb{Z}^{\geq 0})$ is constructed from G, where the weight of each edge (v_i, v_j) in G' is $w'(v_i, v_j) = c(v_i, v_j) + \frac{h(v_i)+h(v_j)}{2}$, $c(v_i, v_j)$ is the flying time between PoIs v_i and v_j, $h(v_i)$ and $h(v_j)$ are the monitoring times of PoIs v_i and v_j, respectively. Then, the optimal values of the rootless monitoring time minimization problem in G and G' are equal.*

Proof. The proof is contained in the supplementary materials file, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2020.3038156. □

In the following, we distinguish our discussion into two cases: (i) the optimal K tours are given; and (ii) the optimal K tours are not given.

3.4.1 Algorithm Analysis With Given Optimal K Tours

Lemma 3. *Given a network $G = (V, E; h : V \rightarrow \mathbb{Z}^{\geq 0}, c : E \rightarrow \mathbb{Z}^{\geq 0})$ and K optimal tours C_1^*, C_2^*, ..., C_K^*, assume that $B \geq OPT$, where $OPT = \max_{k=1}^{K}\{w(C_k^*)\}$. There is an approximation algorithm for the rootless monitoring time minimization problem, which delivers no more than K tours $C_1, C_2, \ldots, C_K$ with their maximum tour time $4\frac{1}{3}B$.*

Proof. The proof is contained in the supplementary materials file, available online. □

3.4.2 Algorithm Analysis Without Given Optimal K Tours

Theorem 1. *Given a network $G=(V,E;h:V\rightarrow\mathbb{Z}^{\geq 0},c:E\rightarrow\mathbb{Z}^{\geq 0})$, there is a $4\frac{1}{3}$-approximation algorithm, `Algorithm 1`, for the rootless monitoring time minimization problem with time complexity of $O(n^4)$, where n is the number of nodes in V.*

Proof. We first show that the maximum tour time of the obtained tours $C_1, C_2, \ldots, C_K$ is $4\frac{1}{3}\cdot OPT$, where OPT is the optimal value. Following Lemma 3, if `Algorithm 1` finds at least $K+1$ tours with their maximum tour weight $4\frac{1}{3}B$ for a guess B of OPT, we then conclude that $B < OPT$. Otherwise ($B\geq OPT$), the algorithm can find no more than K tours with their maximum tour weight $4\frac{1}{3}B$ by Lemma 3.

Following `Algorithm 1`, we know that $B_l+1=B_u$ when the binary search in the algorithm terminates. Then, the algorithm delivers no more than K tours with their maximum tour weight at most $4\frac{1}{3}B$ when setting $B=B_u$, while the algorithm finds at least $K+1$ tours with their maximum tour weight at most $4\frac{1}{3}B$ when setting $B=B_l$. Then, we know that $B_l < OPT$.

Since $B_l < OPT$ and $B_l+1=B_u$, we conclude that $B_l+1=B_u\leq OPT$, as OPT is an integer. On the other hand, we know that the number of obtained tours with their maximum tour time $4\frac{1}{3}B$ is no more than K when $B=B_u$. That is, `Algorithm 1` delivers no more than K tours $C_1, C_2, \ldots, C_K$ with their maximum tour time $4\frac{1}{3}B_u\leq 4\frac{1}{3}OPT$.

We then analyze the time complexity of `Algorithm 1`. The construction of the auxiliary graph G' takes $O(n^2)$, where n is the number of PoIs in V. The number of binary searches in `Algorithm 1` is $O(\log w'(C))$, where $w'(C)$ is the weight of a tour C that visits all nodes in G' by invoking `Christofides' algorithm` [5]. We assume that the value of the maximum edge weight in G' is upper bounded by $O(2^n)$, i.e., $\max_{(u,v)\in G'}\{w'(u,v)\}=O(2^n)$. Notice that this is a very loose assumption. For example, in a large monitoring area, the maximum flying time between two PoIs (i.e., the maximum edge weight) usually is no greater than one hour (i.e., 3,600 seconds), which is no more than 2^n even when there are only 12 PoIs, i.e., $3,600\leq 2^{12}\leq 2^n$ if $n\geq 12$. Then, $O(\log w'(C))=O(\log(n\cdot 2^n))=O(n)$.

In each binary search, it takes $O(n^2)$ time to obtain the p connected components $CC_1, CC_2, \ldots, CC_p$ by removing the edges with their edge weights greater than $\frac{B}{3}$ from graph G'. For the p connected components $CC_1, CC_2, \ldots, CC_p$, there are at most $(2K)^{2K-1}$ combinations, as $p^{2K-1}\leq(2K)^{2K-1}$, where K is the number of UAVs. With each combination, we obtain a tour C'_i that visits the nodes in connected component CC'_i by `Christofides' algorithm`, which takes $O(n^3)$ time [5]. Then, the time complexity of `Algorithm 1` is $O(n^2)+O(\log w'(C))(O(n^2)+(2K)^{2K-1}O(n^3))=O(n)O((2K)^{2K-1}n^3)=O(n^4)$, as K is a given constant. The theorem then follows. □

4 Approximation Algorithm for the Multi-Rooted Monitoring Time Minimization Problem

In this section, we deal with the multi-rooted monitoring time minimization problem, by proposing a $5\frac{1}{3}$-approximation algorithm.

4.1 Basic Idea

Recall that, given a UAV tour network $G_r=(V\cup R, E_r)$ and K UAVs, the multi-rooted monitoring time minimization problem is to find K flying tours $C^r_1, C^r_2, \ldots, C^r_K$ for K UAVs to visit the PoIs in V, such that the maximum tour time among the K tours is minimized, where different tours contain different depots in R.

Denote by OPT^r the optimal value for the multi-rooted problem, which is no less than the optimal value OPT for the rootless problem, i.e., $OPT^r\geq OPT$, since a feasible solution to the rootless problem can be obtained by short-cutting the depots in the optimal solution to the multi-rooted problem.

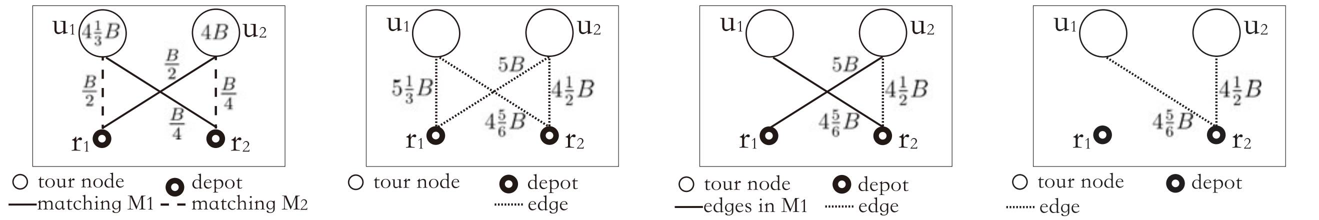
The basic idea behind the proposed algorithm is that, given a guess B of OPT^r with $B\geq OPT^r$, it first finds K' ($\leq K$) *rootless* tours C_1, C_2, ..., $C_{K'}$, such that the weight of each tour is no more than $4\frac{1}{3}B$, by invoking the procedure from lines 7 to 15 in `Algorithm 1` (see Theorem 1), as $B\geq OPT^r\geq OPT$. Then, the algorithm transforms the K' rootless tours C_1, C_2, ..., $C_{K'}$ to K'' ($\leq K'\leq K$) rootless tours C'_1, C'_2, ..., $C'_{K''}$, so that the weight of each tour is also no greater than $4\frac{1}{3}B$, by applying the algorithm in [36]. In addition, each rootless tour C'_k can be matched to a depot r_j in R in an auxiliary graph, and the weight between C'_k and r_j is no greater than $\frac{B}{2}$, where $1\leq k\leq K''$. The algorithm finally obtains K'' rooted tour C^r_1, C^r_2, ..., $C^r_{K''}$, by connecting each C'_k to its matched depot r_j. We will show that the weight of each rooted tour C^r_k is no more than $5\frac{1}{3}B$. We elaborate the algorithm as follows.

4.2 Algorithm

Given a node-weighted and edge-weighted UAV network $G_r=(V\cup R, E_r)$, similar to Step 1 in `Algorithm 1`, an only edge-weighted graph $G'_r=(V\cup R, E_r; w':E_r\rightarrow\mathbb{Z}^{\geq 0})$ is first constructed from G_r, where the weight of each edge (v_i, v_j) in E_r is $w'(v_i,v_j)=c(v_i,v_j)+\frac{h(v_i)+h(v_j)}{2}$, $c(v_i,v_j)$ is the flying time between nodes v_i and v_j in $V\cup R$, $h(v_i)$ and $h(v_j)$ are the monitoring times of v_i and v_j, respectively. Especially, the monitoring time $h(r_j)$ of each depot r_j in R is 0, i.e., $h(r_j)=0$, with $1\leq j\leq K$. Following Lemma 2, the optimal values for the multi-rooted problem in G_r and G'_r are equal.

Recall that OPT^r and OPT are the optimal values for the multi-rooted and rootless problems in G'_r and G', respectively. Given a guess B of OPT^r with $B\geq OPT^r$, the algorithm finds K' ($\leq K$) rootless tours C_1, C_2, ..., $C_{K'}$ with their maximum tour time $4\frac{1}{3}B$, by invoking the procedure from lines 7 to 15 in `Algorithm 1` by Theorem 1, as $B\geq OPT^r\geq OPT$. The algorithm then transforms the K' rootless tours C_1, C_2, ..., $C_{K'}$ into K'' ($\leq K'\leq K$) rootless tours C'_1, C'_2, ..., $C'_{K''}$, such that the weight of each tour is still no more than $4\frac{1}{3}B$, by applying the algorithm in [36], see Lemma 4. There is an interesting property about the derived K'' tours $C'_1, C'_2, \ldots, C'_{K''}$, which is stated as follows.

Denote by $w'(C'_k, r_j)$ the minimum edge weight $w'(v, r_j)$ between a PoI v in C'_k and r_j, i.e., $w'(C'_k, r_j)=\min_{\forall v\in C'_k}\{w'(v,r_j)\}$. A bipartite graph $G_u=(U\cup R, E_u)$ is constructed, where each node u_k in set U represents a tour C'_k with $1\leq k\leq K''$, R is the set of the K depots, and there is an edge (u_k, r_j) in E_u if the weight $w'(C'_k, r_j)$ between C'_k and r_j is no more than $\frac{B}{2}$. For example, Fig. 3a shows such a bipartite graph G_u with



(a) An example of a bipartite graph G_u with $K''=2$ tour nodes and $K=2$ depots.

(b) An example of a bipartite graph $G_b=(U\cup R, E_b)$ with $K''=2$ tour nodes, $K=2$ depots.

(c) A subgraph $G'_b=(U\cup R, E'_b)$ of G_b with edge weights no more than $w_b(e_g)$ when $g=3$, and matching $M_1=\{(u_1,r_2),(u_2,r_1)\}$.

(d) A subgraph $G'_b=(U\cup R, E'_b)$ of G_b with $E'_b=\{e\in E_b, e\le w_b(e_g)\}$, when $g=2$.

Fig. 3. An illustration of the strategy for finding a matching in G_u such that not only each tour node is matched a depot, but also the maximum tour weight derived from the matching is minimized.

$K''=2$ tour nodes and $K=2$ depots, where the weights $w'(C'_1)$ and $w'(C'_2)$ of the tours C'_1 and C'_2 are $4\frac{1}{3}B$, $4B$, respectively, and the weights $w'(C'_1,r_1)$, $w'(C'_1,r_2)$, $w'(C'_2,r_1)$, $w'(C'_2,r_2)$ are $\frac{B}{2},\frac{B}{4},\frac{B}{2},\frac{B}{4}$, respectively.

Yu *et al.* [36] showed that each tour node u_k (representing rootless tour C'_k) can be matched to a depot r_j in G_u. Then, K'' rooted tours can be obtained from the K'' rootless tours $C'_1, C'_2, \ldots, C'_{K''}$ as follows. For each tour node u_k (i.e., C'_k) with $1\le k\le K''$, assume that u_k is matched to a depot r_j. Let $C'_k=v_1\rightarrow v_2\rightarrow\cdots\rightarrow v_{n_k}\rightarrow v_1$. Assume that node v_{n_k} is the nearest node in C'_k to r_j, where $w'(v_{n_k},r_j)=\min_{\forall v\in C'_k}\{w'(v,r_j)\}$. A rooted tour C^r_k can be obtained by adding r_j to C'_k, i.e., $C^r_k=v_1\rightarrow v_2\rightarrow\cdots\rightarrow v_{n_k}\rightarrow r_j\rightarrow v_1$. It can be seen that the weight $w'(C^r_k)$ of the rooted tour C^r_k is no more than the sum of the weight $w'(C'_k)$ of C'_k and B, i.e., $w'(C^r_{k,j})\le w'(C'_k)+B$, as $w'(C^r_k)\le w'(C'_k)+2\cdot w'(u_k,r_j)$ and $w'(u_k,r_j)\le\frac{B}{2}$. Therefore, the weight $w'(C^r_k)$ of C^r_k is no greater than $5\frac{1}{3}B$, as $w'(C'_k)\le 4\frac{1}{3}B$.

Although the maximum weight of the K'' rooted tours by applying the method in [36] is no greater than $5\frac{1}{3}B$, we observe that there may be multiple matchings in G_u such that, in each of the matchings, each tour node u_k is matched to a depot r_j. The maximum tour weights derived from different matchings may vary significantly. For example, there are two matchings in Fig. 3a, i.e., $M_1=\{(u_1,r_2),(u_2,r_1)\}$ and $M_2=\{(u_1,r_1),(u_2,r_2)\}$. The weights of the two rooted tours derived from the matching M_1 are no greater than $4\frac{1}{3}B+2\cdot\frac{B}{4}=4\frac{5}{6}B$ and $4B+2\cdot\frac{1}{2}B=5B$, respectively. The maximum tour weight derived from M_1 then is $\max\{4\frac{5}{6}B,5B\}=5B$. On the other hand, the weights of the two rooted tours derived from the matching M_2 are no greater than $4\frac{1}{3}B+2\cdot\frac{B}{2}=5\frac{1}{3}B$ and $4B+2\cdot\frac{B}{4}=4\frac{1}{2}B$, respectively. The maximum tour weight derived from M_2 then is $\max\{5\frac{1}{3}B,\ 4\frac{1}{2}B\}=5\frac{1}{3}B$, which is larger than that derived from M_1, i.e., $5\frac{1}{3}B>5B$. An important problem thus is how to find a matching in G_u, such that not only each tour node u_k is matched to a depot r_j in the matching, but also the maximum tour weight derived from the matching is minimized. In the following, we propose a novel strategy for the problem.

Denote by $C^r_{k,j}$ the rooted tour by adding a depot r_j to a rootless tour C'_k, where $1\le k\le K''$ and $1\le j\le K$. A bipartite graph $G_b=(U\cup R,E_b)$ is constructed from the K'' rootless tours and the K depots, where each node u_k in set U represents a rootless tour C'_k, and $E_b=E_u$ (i.e., there is an edge (u_k,r_j) in E_b when the weight $w'(u_k,r_j)$ between u_k (representing rootless tour C'_k) and r_j is no more than $\frac{B}{2}$). The weight $w_b(u_k,r_j)$ of the edge (u_k,r_j) in E_b is the weight $w'(C^r_{k,j})$ of the rooted tour $C^r_{k,j}$, i.e., $w_b(u_k,r_j)=w'(C^r_{k,j})$, see Fig. 3b. We now find a matching, such that not only each tour node u_k is matched to a depot r_j in the matching, but also the maximum tour weight derived from the matching is minimized.

Algorithm 2. Approximation Algorithm for the Multi-Rooted Monitoring Time Minimization Problem (`ApproAlg`)

Input: A network $G_r=(V\cup R,E_r)$, the monitoring time $h(v_i)$ of each PoI v_i in V, the flying speed s of a UAV.
Output:a set $\mathcal{C}^r$ of K rooted tours $C^r_1, C^r_2, \ldots, C^r_K$.

```
Construct an auxiliary graph G'_r = (V ∪ R, E_r; w' : E_r → Z^{≥0}) from G_r;
Let B_l = 1, B_u = w'(C^r), where w'(C^r) is the weight of the tour C^r that visits all nodes in G'_r, by invoking Christofides' algorithm; /*B_l and B_u are the lower and upper bounds on OPT^r, respectively. */
while B_l + 1 < B_u do
  B ← ⌊(B_l + B_u)/2⌋; /* B is a guess of OPT^r.*/
  Find K' rootless tours C_1, C_2, ..., C_{K'} with their maximum tour time 4 1/3 B, by invoking the procedure from lines 7 to 15 in Algorithm 1;
  if K' > K then
    B_l ← B; /* the guess B is smaller than OPT^r */
    Skip to the next while loop;
  end if
  /* The rest is the case that K' ≤ K */
   Transform the K' tours C_1, C_2, ..., C_{K'} into K'' rootless tours C'_1, C'_2, ..., C'_{K''}, by applying the algorithm in [36];
  Construct a bipartite graph G_u = (U ∪ R, E_u) from the K'' tours;
  if there are no matchings in G_u so that each node u_k is matched to a depot r_j in R, then
    B_l ← B; /* the guess B is strictly less than OPT^r */
  else
    B_u ← B; /* the guess B is no less than OPT^r*/
    Find a matching M_opt, such that not only each tour node u_k is matched to a depot r_j in M_opt, but also the maximum tour weight derived from the matching is minimized.
    Obtain a set C^r of K'' (≤ K' ≤ K) rooted tours from matching M_opt;
  end if
end while
returnC^r.
```

Denote by M_{opt} the optimal matching in G_b. Let e_{opt} be the edge in M_{opt} with the maximum edge weight, i.e., $w_b(e_{opt}) = \max_{\forall e \in M_{opt}}\{w_b(e)\}$. Denote by m_b the number of edges in E_b, i.e., $m_b = |E_b|$. Let $e_1, e_2, \ldots, e_{m_b}$ be the edges in E_b, and sort the m_b edges by their weights in nondecreasing order, i.e., $w_b(e_1) \le w_b(e_2) \le \cdots \le w_b(e_{m_b})$. Given a guess value e_g of e_{opt}, an unweighted subgraph $G'_b = (U \cup R, E'_b)$ of graph G_b with the edges in G'_b no greater than $w_b(e_g)$ is obtained, i.e., $E'_b = \{e \in E_b, w_b(e) \le w_b(e_g)\}$. We will show that if $w_b(e_g) \ge w_b(e_{opt})$, there is a matching in G'_b, so that each tour node u_k is matched to a depot r_j; Otherwise $(w_b(e_g) < w_b(e_{opt}))$, there are no such matchings in G'_b, see Lemma 5. For example, it can be seen from Fig. 3b that the edge e_{opt} with the maximum edge weight in the optimal matching in G_b is e_3, i.e., $opt = 3$. Fig. 3c shows a subgraph $G'_b = (U \cup R, E'_b)$ of G_b with edge weights no more than $w_b(e_g)$, when $g = 3 \ge opt$. And there is a matching $M_1 = \{(u_1, r_2), (u_2, r_1)\}$, such that each tour node is matched to a depot. On the other hand, Fig. 3d shows a subgraph $G'_b = (U \cup R, E'_b)$ of G_b in Fig. 3b with the weights of edges in E'_b no more than e_g, when $g = 2 < opt$. Obviously, there are no such matchings in graph G'_b of Fig. 3d. Then, the edge e_{opt} can be found by a binary search for the m_b different edges in graph G_b. With the obtained edge e_{opt}, the optimal matching M_{opt} in G_b can be found.

The algorithm for the multi-rooted monitoring time minimization problem is presented in `Algorithm 2`.

4.3 Algorithm Analysis

Lemma 4. *Given a guess B of OPT^r with $B \ge OPT^r$, and K' $(\le K)$ rootless tours $C_1, C_2, \ldots, C_{K'}$ with their maximum tour time $4\frac{1}{3}B$, the algorithm in [36] can transform the K' tours into K'' $(\le K')$ rootless tours $C'_1, C'_2, \ldots, C'_{K''}$ with their maximum tour weight $4\frac{1}{3}B$, such that each tour C'_k can be matched to a depot in R with $1 \le k \le K''$.*

Proof. The proof is contained in the supplementary materials file, available online. □

Lemma 5. *Given a guess e_g of e_{opt} and a bipartite graph G_b, construct an unweighted subgraph $G'_b = (U \cup R, E'_b)$ of G_b with the edges in G'_b no greater than $w_b(e_g)$, i.e., $E'_b = \{e \in E_b, w_b(e) \le w_b(e_g)\}$. If $w_b(e_g) \ge w_b(e_{opt})$, then there is a matching in subgraph G'_b such that each tour node u_k in U is matched to a depot r_j in R. Otherwise $(w_b(e_g) < w_b(e_{opt}))$, there are no such matchings.*

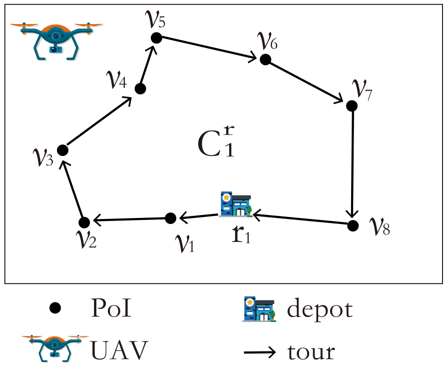
Proof. The proof is contained in the supplementary materials file, available online. □

Theorem 2. *Given a network $G_r = (V \cup R, E_r)$, there is a $5\frac{1}{3}$-approximation algorithm, `Algorithm 2`, for the multi-rooted monitoring time minimization problem, which takes time in $O(n^4)$, where $n = |V|$.*

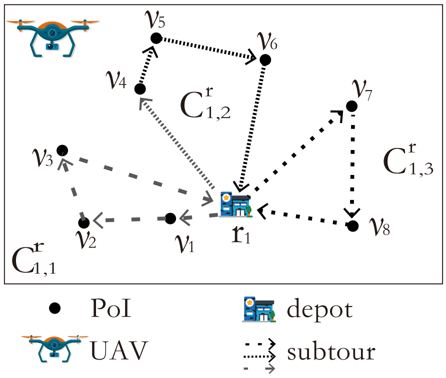
Proof. The proof is contained in the supplementary materials file, available online. □

5 Extension to the Case With Limited Flying Time Constraint on Each UAV

We notice that the maximum flying times of different types of UAVs may vary significantly. For example, the maximum flying time of a HYBRiX 2.1 UAV is as long as four hours [13], while the longest flying time of a DJI Phantom 4 Pro UAV is only about half an hour [22]. Denote by T_{max} the maximum flying time of each UAV adopted in an application of monitoring a disaster area. Then, it is possible that the time spent on some tour by the proposed algorithm in the previous section may be larger than the maximum flying time T_{max} of each UAV, see Fig. 4a. The UAV thus is unable to monitor all PoIs assigned to it.



(a) A tour C^r_1 delivered by `Algorithm 2`.



(b) $p_1 = 3$ subtours $C^r_{1,1}$, $C^r_{1,2}$ and $C^r_{1,3}$ are obtained by splitting the tour C^r_1.

Fig. 4. An illustration of the tour splitting procedure in `Algorithm 3`.

To deal with the limited flying time constraint on each UAV, we assume that the UAV can return back to its depot to replace its battery with a fully-charged battery, when the UAV will run out of its battery energy soon. Then, the UAV can monitor all PoIs assigned to it in multiple tours and the spent time of each tour is no greater than its maximum flying time. For example, Fig. 4b shows that a UAV monitors its assigned PoIs in three tours, which means that it needs to replace its battery twice at its depot. In this case, the spent time of each UAV is the sum of the amounts of spent time in the multiple tours.

In this section, we show how to extend the proposed algorithm in the previous section to the case where the maximum flying time of each UAV is constrained by a value T_{max}.

5.1 Basic Idea

The basic idea behind the proposed algorithm is as follows. We first ignore the limited flying time constraint on each UAV and obtain K rooted tours $C^r_1, C^r_2, \ldots, C^r_K$ for the multi-rooted monitoring time minimization problem by invoking the algorithm in the previous section, where each tour C_k^r contains depot r_k with $1 \le k \le K$. For each tour C_k^r, if its the spent time is larger than the maximum UAV flying time T_{max}, we split C_k^r into multiple r_k-rooted subtours $C_{k,1}^r, C_{k,2}^r, \ldots, C_{k,p_k}^r$ such that the sum of the amounts of spent times in the subtours is minimized, subject to the constraint that the spent time on each of the subtours is no greater than T_{max}, where p_k is the number of split subtours and will be determined later. Since each of the p_k subtours contains depot r_k, the flying trajectory C_k^f of UAV k can be represented as $C_k^f = C_{k,1}^r \rightarrow C_{k,2}^r \rightarrow \cdots \rightarrow C_{k,p_k}^r$ and the total spent time of UAV k in the p_k subtours is $w(C_k^f) = \sum_{j=1}^{p_k} w(C_{k,j}^r)$.

Algorithm 3. Approximation Algorithm for the Multi-Rooted Monitoring Time Minimization Problem With a Maximum Flying Time Constraint on Each UAV (`ApproAlgLimitedFlyTime`)

Input: A UAV network $G_r = (V \cup R, E_r)$ and the maximum UAV flying time T_{max}
Output: a set $\mathcal{C}^f$ of K flying trajectories C_1^f, C_2^f, ..., C_K^f for K UAVs, respectively

```
Find K rooted tours C_1^r, C_2^r, ..., C_K^r, by ignoring the maximum
  UAV flying time constraint and invoking Algorithm 2;
C^f <- ∅; /* the set of obtained K flying trajectories */
for k <- 1 to K do
  /* consider each tour C_k^r */
  if w(C_k^r) <= T_max then
    Let C_k^f = C_k^r;
  else
    /* w(C_k^r) > T_max */
    Split the tour C_k^r into p_k subtours C_{k,1}^r, C_{k,2}^r, ..., C_{k,p_k}^r with
      the maximum UAV flying time T_max.
    Let C_k^f = C_{k,1}^r -> C_{k,2}^r -> ... -> C_{k,p_k}^r;
  end if
  Let C^f <- C^f ∪ {C_k^f};
end for
return C^f.
```

5.2 Algorithm

Given a UAV network $G_r = (V \cup R, E_r)$, we first obtain K rooted tours $C_1^r, C_2^r, \ldots, C_K^r$ by ignoring the maximum UAV flying time T_{max} and invoking `Algorithm 2`, where each tour C_k^r contains depot r_k with $1 \le k \le K$.

We construct the flying trajectory C_k^f of UAV k from tour C_k^r ($1 \le k \le K$), by distinguishing into two cases.

Case (i): the spent time $w(C_k^r)$ of tour C_k^r is no greater than the maximum flying time T_{max} of each UAV, i.e., $w(C_k^r) \le T_{max}$. Then, the flying trajectory C_k^f of UAV k is tour C_k^r, i.e., $C_k^f = C_k^r$.

Case (ii): the spent time $w(C_k^r)$ of tour C_k^r is larger than the maximum UAV flying time T_{max}, i.e., $w(C_k^r) > T_{max}$. For example, Fig. 4a shows such a tour C_1^r and $w(C_1^r) > T_{max}$. In the following, we split tour C_k^r into, say p_k, r_k-rooted subtours $C_{k,1}^r, C_{k,2}^r, \ldots, C_{k,p_k}^r$ such that the sum of the amounts of spent times in the subtours is minimized, subject to the constraint that the spent time in each of the p_k subtours is no greater than T_{max}.

Let tour $C_k^r = r_k \rightarrow v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_{n_k} \rightarrow r_k$, where n_k is the number of PoIs in C_k^r. Assume that we have split j subtours $C_{k,1}^r, C_{k,2}^r, \ldots, C_{k,j}^r$ from C_k^r and PoIs $v_1, v_2, \ldots, v_{n_j}$ are assigned to the j subtours. Initially, $j = 0$. Then, the $(j+1)$th subtour $C_{k,j+1}^r$ is $C_{k,j+1}^r = r_k \rightarrow v_{n_j+1} \rightarrow v_{n_j+2} \rightarrow \cdots \rightarrow v_{n_{j+1}} \rightarrow r_k$ such that the spent time of subtour $C_{k,j+1}^r$ is no greater than T_{max}, while the spent time of subtour $r_k \rightarrow v_{n_j+1} \rightarrow v_{n_j+2} \rightarrow \cdots \rightarrow v_{n_{j+1}} \rightarrow \mathbf{v_{n_{j+1}+1}} \rightarrow r_k$ is larger than T_{max}, i.e.,

$$w(r_k \rightarrow v_{n_j+1} \rightarrow \cdots \rightarrow v_{n_{j+1}} \rightarrow r_k) \le T_{max}, \tag{11}$$

and

$$w(r_k \rightarrow v_{n_j+1} \rightarrow \cdots \rightarrow v_{n_{j+1}} \rightarrow \mathbf{v_{n_{j+1}+1}} \rightarrow r_k) > T_{max}. \tag{12}$$

For example, Fig. 4b shows that the spent time of subtour $r_1 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow r_1$ is no greater than T_{max}, whereas the spent time of subtour $r_1 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow r_1$ is larger than T_{max}. Therefore, the first split subtour $C_{1,1}^r$ is $r_1 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow r_1$.

We continue the tour split procedure until each of the n_k PoIs in C_k^r is assigned to a subtour. Denote by p_k the number of split subtours from C_k^r. For example, Fig. 4b shows that $p_1 = 3$ subtours $C_{1,1}^r, C_{1,2}^r, C_{1,3}^r$ are split from the tour C_1^r in Fig. 4a. Then, the flying trajectory C_k^f of UAV k is $C_k^f = C_{k,1}^r \rightarrow C_{k,2}^r \rightarrow \cdots \rightarrow C_{k,p_k}^r$.

The detailed algorithm is presented in `Algorithm 3`.

5.3 Algorithm Analysis

Denote by OPT^f the value of the optimal solution to the multi-rooted monitoring time minimization problem with the maximum UAV flying time T_{max} constraint. The following lemma estimates an important lower bound on OPT^f, which is the key of the approximation ratio analysis.

Lemma 6. *Given a UAV network $G_r = (V \cup R, E_r)$ and a maximum UAV flying time T_{max}, denote by OPT^r and OPT^f the values of the optimal solutions to the multi-rooted monitoring time minimization problem when* ignoring *and* considering *the maximum UAV flying time T_{max} constraint, respectively. Then, $OPT^r \le OPT^f$.*

Proof. The proof is contained in the supplementary materials file, available online. □

Theorem 3. *Given a UAV network $G_r = (V \cup R, E_r)$ and a maximum UAV flying time T_{max}, there is an α-approximation algorithm, `Algorithm 3`, for the multi-rooted monitoring time minimization problem with a time complexity $O(|V|^4)$ when considering the maximum UAV flying time T_{max}, where the approximation ratio of the algorithm is $\alpha = 5\frac{1}{3} \cdot \frac{1}{1-\frac{A}{T_{max}}}$, A is the maximum time for monitoring only one PoI in V with $A = \max_{v \in V, r \in R}\{2c(v,r) + h(v)\}$, $c(v,r)$ is the UAV flying time between a PoI v and a depot r, and $h(v)$ is the monitoring time of PoI v.*

Proof. The proof is contained in the supplementary materials file, available online.

Remark. The approximation ratio α usually is small due to the following reasoning. On one hand, a UAV is able to monitor many PoIs within its maximum flying time T_{max}, e.g., $T_{max} = 1$ hour.

TABLE 1
Parameters Table

Parameters	Values
monitoring area	10 km $\times$ 10 km $\times$ 500 m
number of PoIs n	100 to 500
PoI monitoring time	$[h_{min}, h_{max}]$ = [10s, 180s]
flying speed of UAVs s	10 m/s
number of UAVs K	1 to 8

On the other hand, the maximum time A for monitoring only one PoI usually is much smaller than the maximum UAV flying time T_{max}. For example, consider a DJI Phantom 4 Pro UAV, its maximum flying speed is 20 m/s (i.e., 72 km/h) [22]. The flying time for a round trip with total $10 \text{ km} \times 2$ is $\frac{20{,}000\ m}{20\ m/s} = 16.7$ minutes.

For example, when $A = \frac{T_{max}}{2}$, the approximation ratio of `Algorithm 3` is $5\frac{1}{3}\frac{1}{1-\frac{A}{T_{max}}} = 5\frac{1}{3}\frac{1}{1-\frac{1}{2}} = 10\frac{2}{3}$. □

6 Performance Evaluation

In this section, we evaluate the performance of the proposed algorithms through extensive experiments. We also study the impact of the network size, the maximum monitoring time, the speed of a UAV, the number of depots.

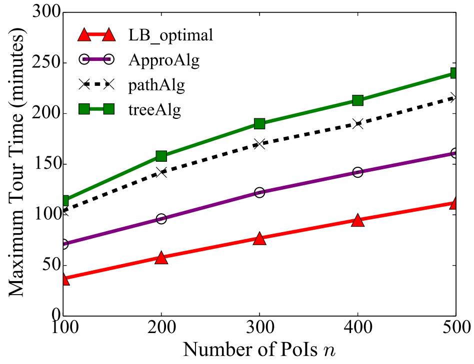
6.1 Simulation Environment

We consider a disaster area in a 10 km×10 km×500 m three-dimensional euclidean space. There are from 100 to 500 PoIs in the area, where the coordinate of each PoI is randomly chosen. Also, the number K of UAVs varies from 1 to 8, and each UAV is initially located at a depot, where the depot is randomly located at the border of the disaster area. The flying speed of each UAV is $s = 10$ m/s [22]. The monitoring time $h(v_i)$ of each PoI v_i is randomly chosen from an interval $[h_{min}, h_{max}]$, where $h_{min} = 10$ s and $h_{max} = 3$ min = 180 s [16]. Table 1 lists the parameters used in the experiments. Unless otherwise specified, those parameters will be used as default settings.

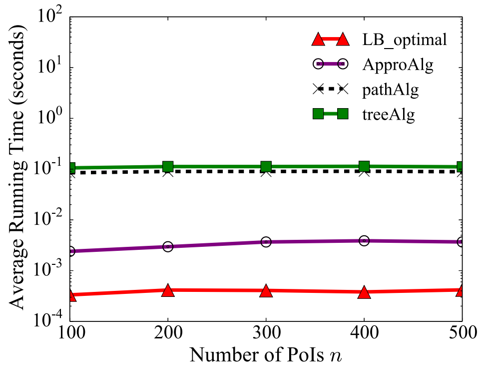
To evaluate the performance of the proposed algorithms `ApproAlg` and `ApproAlgLimitedFlyTime`, we consider three existing benchmarks.

i) Algorithm `treeAlg` [28] delivers a 7-approximation solution to the min-max cycle cover problem, which is to find K rooted tours to cover nodes in an only edge-weighted graph such that the maximum tour length among the K tours is minimized.
ii) Algorithm `pathAlg` [36] finds an improved 6-approximation solution to the min-max cycle cover problem.
iii) Algorithm `LB_optimal` [28] delivers a lower bound on the multi-rooted maximum tour time problem as follows. It first obtains an only edge-weighted graph $G'_r = (V \cup R, E_r; w' : E_r \rightarrow \mathbb{Z}^{\geq 0})$ from the original network G_r by following Step 1 of Algorithm 2. It then constructs an auxiliary graph $G''_r = (V \cup \{r_0\}, E_s; w'' : E_s \rightarrow \mathbb{Z}^{\geq 0})$ from G'_r, by contracting all depots in R into a single node r_0. That is, we first remove all depots in R and their adjacent edges from G'_r, then introduce a new virtual node r_0 and there is an edge between each node $v \in V$ and r_0 with its weight being the minimum edge weight between v and any depot $r_i \in R$, i.e., $w''(v, r_0) = \min_{r_i \in R}\{w'(v, r_i)\}$. Then, in the auxiliary graph G''_r, we can find a minimum spanning tree T. Finally, the lower bound is

$$LB\_optimal = \frac{w''(T)}{K}. \quad (13)$$



(a) Maximum tour time



(b) Algorithm running time

Fig. 5. Performance of algorithms `ApproAlg`, `treeAlg`, `pathAlg`, and `LB_optimal` by varying the number of PoIs n from 100 to 500 with $K = 5$ UAVs.

The simulator was implemented in language C++ and all simulations were run on an Apple server with an Intel(R) Core(TM) i7 CPU (2.5 GHz) and a 16 GB RAM. Each value in figures is the average of the results by applying each mentioned algorithm to 100 different network topologies with the same network size.

6.2 Performance Evaluation With Ignoring the Maximum UAV Flying Time

We first study the performance of different algorithms by varying the number of PoIs n from 100 to 500 with $K = 5$ UAVs. Fig. 5a shows that the maximum tour time among the tours delivered by the proposed algorithm `ApproAlg` is about from 25 to 30 percent shorter than those by existing algorithms `treeAlg` and `pathAlg`. For example, the maximum tour times by algorithms `ApproAlg`, `treeAlg` and `pathAlg` are 160, 216, and 240 min, respectively, when there are $n = 500$ PoIs in the disaster area. On the other hand, Fig. 5a also demonstrates that the empirical approximation ratio of the maximum tour time by algorithm `ApproAlg` to that by algorithm `LB_optimal` is around from 1.44 to 1.92, which is much smaller than its analytical

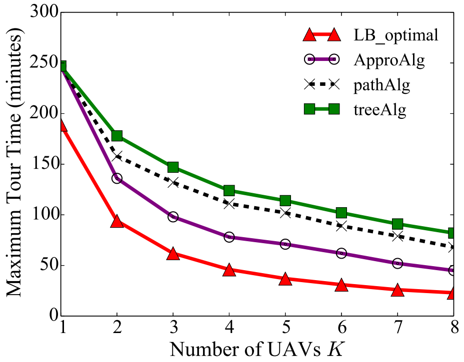


Fig. 6. The performance of different algorithms by increasing the number K of UAVs from 1 to 8, when there are $n = 100$ PoIs.

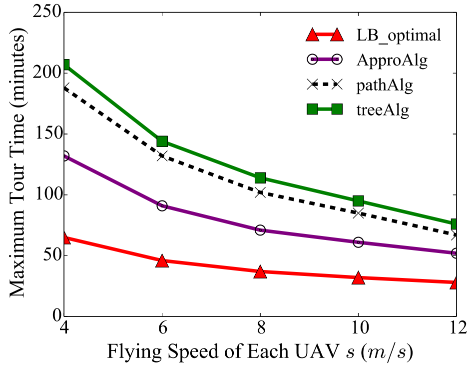


Fig. 7. The performance of different algorithms by increasing the flying speed s of each UAV from 4 m/s to 12 m/s, when there are $n = 100$ PoIs and $K = 5$ UAVs.

approximation ratio $5\frac{1}{3}$ (see Theorem 2 in Section 4.3). This indicates that the estimate on the theoretical approximation ratio $5\frac{1}{3}$ is very conservative. Finally, Fig. 5b plots the running times of different algorithms, from which it can be seen that the running time of the proposed algorithm `ApproAlg` is very short, e.g., no more than 0.01 s when $n = 500$ PoIs. This implies that the proposed strategy to reduce the number of potential combinations in Section 3.3 is very efficient. In the following, we do not compare the running times of the algorithms, since the curves are similar.

We then evaluate the algorithm performance by increasing the number K of UAVs from 1 to 8 when there are $n = 100$ PoIs to be monitored. Fig. 6 plots that the maximum tour times by the three algorithms `ApproAlg`, `treeAlg` and `pathAlg` are almost identical when there is only $K = 1$ UAV. However, the gaps between the maximum tour time by algorithm `ApproAlg` and those by both algorithms `treeAlg` and `pathAlg` become larger when more UAVs are deployed, i.e., K is larger. For example, Fig. 6 shows that the maximum tour time by algorithm `ApproAlg` is 50 min, while the maximum tour times of the algorithms `treeAlg`, `pathAlg` and `LB_optimal` are 80, 75, 25 min, respectively, when there are $K = 8$ UAVs. In addition, Fig. 6 shows that the ratio of the maximum tour time by algorithm `ApproAlg` to that by algorithm `LB_optimal` is from 1.3 to 2.

We further investigate the algorithm performance by increasing the flying speed s of each UAV from 4 m/s to 12 m/s, when there are $n = 100$ PoIs and $K = 5$ UAVs. Fig. 7 demonstrates that the maximum tour time by the four algorithms decrease with a larger flying speed of each UAV, and the maximum tour time of algorithm `ApproAlg` is about from 22 to 30 percent smaller than those by algorithms `treeAlg` and `pathAlg`, while the ratio of the maximum tour time by algorithm `ApproAlg` to that by algorithm `LB_optimal` is around from 1.85 to 2.

We finally study the algorithm performance by increasing the maximum monitoring time of a PoI h_{max} from 10 s to 180 s(=3 min), while fixing the minimum PoI monitoring time h_{min} at 10 s, where the monitoring time $h(v_i)$ of a PoI v_i is randomly chosen from the interval $[h_{min}, h_{max}]$. Fig. 8 shows that the maximum tour time by algorithm `ApproAlg` is about 32 percent shorter than those by algorithms `treeAlg` and `pathAlg`. For example, the maximum tour times by algorithms `ApproAlg`, `treeAlg` and `pathAlg` are 55, 81, and 93 min, respectively, when $h_{min} = 10$ s. Fig. 8 also demonstrates that the empirical ratio of the maximum tour time by algorithm `ApproAlg` to that by algorithm `LB_optimal` varies between 1.45 and 2.1.

6.3 Performance Evaluation With Taking the Maximum UAV Flying Time into Consideration

To evaluate the performance of the proposed algorithm `ApproAlgLimitedFlyTime`, we compare with algorithms `LB_optimal`, `treeAlg` and `pathAlg`. Notice that we modify both algorithms `treeAlg` and `pathAlg` by adopting the similar tour split procedure in Section 5.2, such that the both algorithms take the maximum UAV flying time T_{max} into consideration. On the other hand, algorithm `LB_optimal` calculates a lower bound on the optimal solution still by Eq. (13), since a lower bound on an optimal solution to the problem *without* considering the maximum UAV flying time T_{max} also is a lower bound on the optimal solution to the problem *with* considering T_{max} by Lemma 6.

We study the performance of different algorithms by varying the maximum UAV flying time T_{max} from 20 min to 120 min, when there are $K = 5$ UAVs, and $n = 200$ PoIs. Fig. 9 shows that the maximum spent time among the K UAVs in their flying trajectories delivered by the proposed algorithm `ApproAlgLimitedFlyTime` is about from 30 to 40 percent shorter than those by existing algorithms `treeAlg` and `pathAlg`. For example, the maximum spent time by algorithm `ApproAlgLimitedFlyTime` is 89 min, while the maximum spent times of the three algorithms

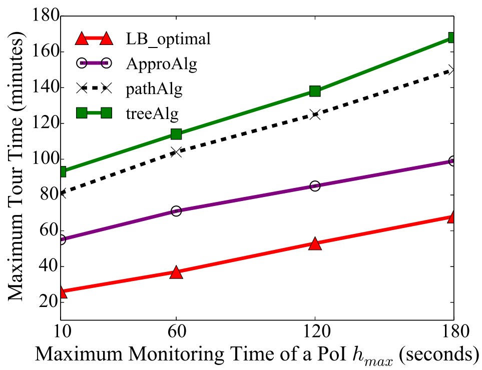


Fig. 8. The performance of different algorithms by increasing the maximum monitoring time h_{max} of a PoI from 10 s to 180 s while fixing h_{min} at 10 s, when there are $n = 100$ PoIs, $K = 5$ UAVs, and the UAV flying speed s is 8 m/s.

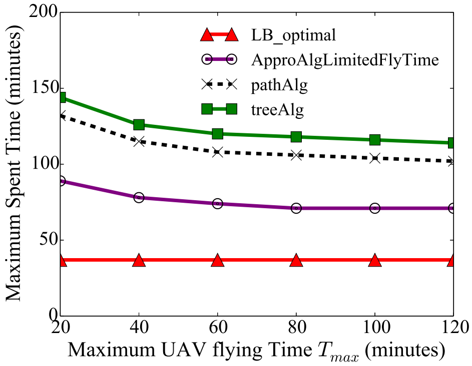


Fig. 9. The performance of different algorithms by increasing the maximum flying time T_{max} of each UAV from 20 min to 120 min, when there are $K = 5$ UAVs and $n = 200$ PoIs.

`treeAlg`, `pathAlg` and `LB_optimal` are 144, 132, and 37 minutes, respectively, when the maximum UAV flying time $T_{max} = 20$ minutes. In addition, Fig. 9 also demonstrates that the empirical ratio of the maximum spent time by algorithm `ApproAlgLimitedFlyTime` to that by algorithm `LB_optimal` is from 1.9 to 2.4, which is much smaller than its analytical approximation ratio $5\frac{1}{3} \cdot \frac{1}{1-\frac{A}{T_{max}}}$ (see Theorem 3 in Section 5.3).

7 Related Work

The employment of UAVs to monitor PoIs has attracted lots of attentions [4], [15], [18], [21], [31], [32], [33], [37]. Some studies focused on the scheduling of UAVs to cover a monitoring area such that the cost consumed by UAVs is minimized [23], [26], [39], [40], where the cost means the number of dispatched UAVs or the energy consumed by UAVs. For example, Torres *et al.* [26] studied a problem of scheduling a UAV to fully cover an area of interest for 3D terrain reconstruction, such that the amount of energy consumed by the UAV is minimized. Zorbas *et al.* [40] investigated a problem of finding hovering locations for multiple UAVs to cover static or mobile targets, under the assumption that the energy consumption of a UAV is less if it hovers at a lower altitude, but the covering area by the UAV becomes also smaller. They proposed two centralized algorithms and a localized algorithm for the problem, such that the total amount of energy consumed by UAVs is minimized. Saeed *et al.* [23] investigated a problem of employing UAVs to uninterruptedly monitor targets and collect accurate information of the targets, so as to minimize the number of deployed UAVs. Zhao *et al.* [39] employed UAVs to provide communication service for users in an area, and studied a problem of deploying the minimum number of UAVs such that all users in the area are served by the UAVs, subject to a connectivity constraint.

On the other hand, there are some studies on balancing the flying distances of different UAVs to monitor PoIs in disaster areas [6], [14], [19], [24], [28], [36], [38]. For example, Scott *et al.* [24] found flying tours of multiple UAVs to cover PoIs in a geometrically complex area, so as to minimize the maximum flying distance among the UAVs. Modares *et al.* [19] recognized that the energy consumed by a UAV is related to not only its flying distance but also the number of turns in its flying tour, and they employed multiple UAVs to monitor PoIs so that the maximum consumed energy of the UAVs in their flying tours is minimized, by proposing a Lin-Kernighan heuristic algorithm. Zhan *et al.* [38] investigated a problem of finding flying tours for multiple UAVs to collect data from sensors so that the maximum time spent by the UAVs is minimized, by devising a genetic algorithm. Kim *et al.* [14] proposed an 8-approximation algorithm for the problem of dispatching UAVs to collect information of PoIs in search and rescue applications, such that the longest flying time among the UAVs is minimized. Xu *et al.* [28] devised a 7-approximation algorithm for the min-max cycle cover problem of finding K rooted tours to cover nodes so that the longest length of the K tours is minimized, while Yu *et al.* [36] recently improved the approximation ratio to 6, by utilizing the algorithm in [35]. In addition, Xu *et al.* [29], [30] considered a multi-node charging model in which a mobile charger can charge multiple sensors simultaneously in the charging range of the charger, and studied a problem of finding tours for multiple mobile chargers to charge sensors such that the longest tour time among the found tours is minimized, subject to that no sensors are charged by multiple chargers at the same time. They proposed a novel approximation algorithm for the problem.

We note that most of the existing algorithms [14], [19], [24], [28], [36], [38] ignored the monitoring time of PoIs and aimed to minimize the longest flying distance/time of the UAVs, and we observe that it takes some time to monitor a PoI. In this paper, we consider a problem, i.e., the multi-rooted monitoring completion time minimization problem, of finding flying tours for multiple UAVs to collaboratively monitor PoIs in a disaster area, such that the maximum spent time of the UAVs in their tours is minimized, where the spent time of a UAV in its tour consists of not only its flying time but also the monitoring time of PoIs. We also propose a $5\frac{1}{3}$-approximation algorithm for the problem, which has a better (i.e., smaller) approximation ratio than the state-of-the-art, i.e., the 6-approximation algorithm in [36].

8 Conclusion

In this paper, we studied a problem of finding flying tours for multiple UAVs such that the maximum time spent by the UAVs in their tours is minimized, where the spent time of a UAV in its tour consists of not only the flying time but also the monitoring time of PoIs. We then proposed a novel $5\frac{1}{3}$-approximation algorithm for the problem in this paper, improving the best result with an approximation ratio of 6 so far in [36]. We also extended the proposed algorithm to the case that the maximum flying time of each UAV is bounded. We finally evaluated the performance of the proposed algorithms via simulation environments, and experimental results showed that the proposed algorithms are very promising. Especially, the maximum spent times by the UAVs in the tours by the proposed algorithms are up to 30 percent shorter than those by existing algorithms. In addition, the empirical approximation ratios of the proposed algorithms are no more than 2.4, which are much less than their analytical approximation ratio that are at least $5\frac{1}{3}$.

Acknowledgments

The authors would like to thank the three anonymous referees for their expertise comments and constructive suggestions, which have helped us improve the quality and presentation of the paper significantly. The work by Wenzheng Xu was supported by the National Natural Science Foundation of China (NSFC) with grant number 61602330, Sichuan Science and Technology Program (Grant Nos. 2018GZDZX0010, 2017GZDZX0003, and 20ZDYF0653). The work by Jian Peng was supported by the National Key Research and Development Program of China (Grant No. 2017YFB0202403) and Sichuan Key Research and Development Program (Grant No. 2018GZ0098). The work by Weifa Liang was supported by Australian Research Council under its Discovery Project Scheme with Grant No. DP200101985. The work by Xiaohua Jia was supported by the Research Grants Council of Hong Kong with Project No. CityU11214316. Finally, the work by Minghui Wang was supported by the Innovation youth fund of ocean telemetry engineering technology research center of State Oceanic Administration (Grant No. 2015001) and Chengdu Science and technology fund (Grant No. 2015-HM02-00080-SF). Qing Guo and Jian Peng contributed equally to this work, i.e., co-primary authors.

REFERENCES

[1] 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, 2006.

[2] D. Avola, G. L. Foresti, N. Martinel, C. Micheloni, D. Pannone, and C. Piciarelli, "Aerial video surveillance system for small-scale UAV environment monitoring," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveillance*, 2017, pp. 1–6.

[3] C. A. Baker, S. Ramchurn, W. T. Teacy, and N. R. Jennings, "Planning search and rescue missions for UAV teams," in *Proc. Eur. Conf. Artif. Intell.*, 2016, pp. 1777–1778.

[4] L. Chen *et al.*, "WideSee: Towards wide-area contactless wireless sensing," in *Proc. 17th Conf. Embedded Netw. Sensor Syst.*, 2019, pp. 258–270.

[5] N. Christofides and Carnegie-Mellon Univ Pittsburgh PA Management Sciences Research Group, "Worst-case analysis of a new heuristic for the traveling salesman problem," Defense Technical Information Center, 1976. [Online]. Available: https://books.google.com.hk/books?id=2A7eygAACAAJ

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

[7] DJI Cinema Color System ZENMUSE X7. 2020. [Online]. Available: https://www.dji.com/zenmuse-x7

[8] 2019. [Online]. Available: http://thedronegirl.com/2019/01/11/state-farm/

[9] M. Erdelj, E. Natalizio, K. R. Natalizio, and I. F. Akyildiz, "Help from the sky: Leveraging UAVs for disaster management," *IEEE Pervasive Comput.*, vol. 16, no. 1, pp. 24–32, Jan–Mar. 2017.

[10] G. N. Frederickson, M. S. Hecht, and C. E. Kim, "Approximation algorithms for some routing problems," in *Proc. 17th Annu. Symp. Found. Comput. Sci.*, 1976, pp. 216–227.

[11] S. Hayat, E. Yanmaz, T. X. Brown, and C. Bettstetter, "Multi-objective UAV path planning for search and rescue," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 5569–5574.

[12] How to help victims of Australia's apocalyptic wildfires. 2020. [Online]. Available: https://www.cbsnews.com/news/australia-fires-how-to-help-victims-evacuees-firefighters-animals-wildfires/

[13] HYBRiX 2.1 is a unique multirotor that provides up to 4 hours of operational flight time. 2020. [Online]. Available: https://www.quaternium.com/uav/hybrix-drone/

[14] D. Kim, L. Xue, D. Li, Y. Zhu, W. Wang, and A. O. Tokuta, "On theoretical trajectory planning of multiple drones to minimize latency in search-and-reconnaissance operations," *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 3156–3166, Nov. 2017.

[15] A. Kleiner and A. Kolling, "Guaranteed search with large teams of unmanned aerial vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 2977–2983.

[16] 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.

[17] L. Lin and M. A. Goodrich, "Hierarchical heuristic search using a Gaussian mixture model for UAV coverage planning," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2532–2544, Dec. 2014.

[18] V. Mersheeva and G. Friedrich, "Multi-UAV monitoring with priorities and limited energy resources," in *Proc. 25th Int. Conf. Autom. Planning Scheduling*, 2015, pp. 327–356.

[19] J. Modares, F. Ghanei, N. Mastronarde, and K. Dantu, "UB-ANC planner: Energy efficient coverage path planning with multiple drones," in *Proc. IEEE Conf. Robot. Autom.*, 2017, pp. 6182–6189.

[20] R. E. Munich, "NatCatSERVICE Loss events worldwide 1980–2015," *Munich Reinsurance, Munich*, 2016. [Online]. Available: https://reliefweb.int/report/world/natcatservice-loss-events-worldwide-1980-2015

[21] L. Paull, C. Thibault, A. Nagaty, M. Seto, and H. Li, "Sensor-driven area coverage for an autonomous fixed-wing unmanned aerial vehicle," *IEEE Trans. Cybern.*, vol. 44, no. 9, pp. 1605–1618, Sep. 2014.

[22] Phantom 4 Specification. 2020. [Online]. Available: https://www.dji.com/phantom-4-adv/info

[23] A. Saeed, A. Abdelkader, M. Khan, A. Neishaboori, K. A. Harras, and A. Mohamed, "On realistic target coverage by autonomous drones," *ACM Trans. Sensor Netw.*, vol. 15, no. 3, 2019, Art. no. 32.

[24] K. Scott, R. Dai, and M. Kumar, "Occlusion-aware coverage for efficient visual sensing in unmanned aerial vehicle networks," in *Proc. IEEE Global Commun. Conf.*, 2016, pp. 1–6.

[25] P. Tokekar, J. V. Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic UAV and UGV system for precision agriculture," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1498–1511, Dec. 2016.

[26] M. Torres, D. A. Pelta, J. L. Verdegay, and J. C. Torres, "Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction," *Expert Syst. Appl.*, vol. 55, no. 15, pp. 441–451, 2015.

[27] A. Wallar, E. Plaku, and D. A. Sofge, "Reactive motion planning for unmanned aerial surveillance of risk-sensitive areas," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 969–989, Jul. 2015.

[28] 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.

[29] W. Xu, W. Liang, H. Kan, Y. Xu, and X. Zhang, "Minimizing the longest charge delay of multiple mobile chargers for wireless rechargeable sensor networks by charging multiple sensors simultaneously," in *Proc. 39th IEEE Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 881–890.

[30] 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.*, early access, Feb. 14, 2020, doi: 10.1109/TMC.2020.2973979.

[31] W. Xu, W. Liang, X. Lin, and G. Mao, "Efficient scheduling of multiple mobile chargers for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7670–7683, Sep. 2016.

[32] W. Xu *et al.*, "Approximation algorithms for the team orienteering problem," in *Proc. 39th IEEE Int. Conf. Comput. Commun.*, 2020, pp. 1389–1398.

[33] W. Xu *et al.*, "Approximation algorithms for the generalized team orienteering problem and its applications," *IEEE/ACM Trans. Netw.*, early access, Oct. 07, 2020, doi: 10.1109/TNET.2020.3027434.

[34] Z. Xu, D. Xu, and W. Zhu, "Approximation results for a min–max location-routing problem," *Discrete Appl. Math.*, vol. 160, no. 3, pp. 306–320, 2012.

[35] 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, 2016.

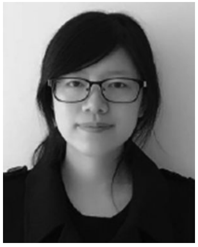
[36] W. Yu and Z. Liu, "Better approximability results for min-max tree/cycle/path cover problems," *J. Combinatorial Optim.*, vol. 37, no. 2, pp. 563–578, 2018.

[37] N. Zhao *et al.*, "UAV-Assisted emergency networks in disasters," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 45–51, Feb. 2019.

[38] C. Zhan and Y. Zeng, "Completion time minimization for multi-UAV-enabled data collection," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4859–4872, Oct. 2019.

[39] H. Zhao, H. Wang, W. Wu and J. Wei, "Deployment algorithms for UAV airborne networks toward on-demand coverage," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2015–2031, Sep. 2018.

[40] D. Zorbas, L. D. P. Pugliese, T. Razafindralambo, and F. Guerriero, "Optimal drone placement and cost-efficient target coverage," *J. Netw. Comput. Appl.*, vol. 75, pp. 16–31, Nov. 2016.

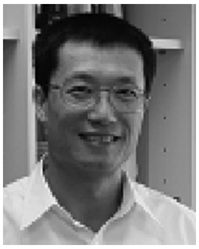


Qing Guo received the BE and ME degrees from Sichuan University, Chengdu, China, in 2014 and 2019, respectively, all in computer science. She currently is working toward the PhD degree at Sichuan University, Chengdu, China. Her research interests include UAV networks, wireless sensor networks, and mobile computing.

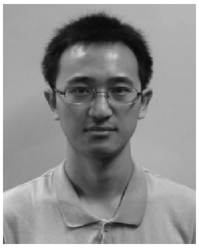
Jian Peng received the BA and PhD degrees from the University of Electronic Science and Technology of China (UESTC), China, in 1992 and 2004, respectively. He is currently a professor at the College of Computer Science, Sichuan University, China. His recent research interests include wireless sensor networks, Big Data, and cloud computing.

Wenzheng Xu (Member, IEEE) received the BSc, ME, and PhD degrees in computer science from Sun Yat-Sen University, Guangzhou, P.R. China, in 2008, 2010, and 2015, respectively. He currently is an associate professor at Sichuan University, China. Also, he was a visitor at both the Australian National University, Australia, and the Chinese University of Hong Kong, Hong Kong. His research interests include wireless ad hoc and sensor networks, mobile computing, approximation algorithms, combinatorial optimization, online social networks, and graph theory.

Weifa Liang (Senior Member, IEEE) received the BSc degree from Wuhan University, China, in 1984, the ME degree from the University of Science and Technology of China, China, in 1989, all in computer science, and the PhD degree from the Australian National University, Australia, in 1998. He is currently a full professor at the Australian National University, Australia. His research interests include design and analysis of energy-efficient routing protocols for wireless ad hoc and sensor networks, mobile edge computing and cloud computing, software-defined networking, online social networks, design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory.



Xiaohua Jia (Fellow, IEEE) received the BSc and MEng degrees from the University of Science and Technology of China, China, in 1984 and 1987, respectively, and the DSc degree in information science from the University of Tokyo, Japan, in 1991. He is currently a chair professor at the Department of Computer Science, City University of Hong Kong, Hong Kong. His research interests include cloud computing and distributed systems, computer networks, wireless sensor networks and mobile wireless networks. He is an editor of the *IEEE Transactions on Parallel and Distributed Systems* (2006-2009), the *Journal of World Wide Web*, the *Wireless Networks*, the *Journal of Combinatorial Optimization*, and so on. He is the general chair of ACM MobiHoc 2008, TPC co-chair of IEEE MASS 2009, area-chair of IEEE INFOCOM 2010, TPC co-chair of IEEE GlobeCom 2010, Ad Hoc and Sensor Networking Symposium, and panel co-chair of IEEE INFOCOM 2011.



Zichuan Xu received the BSc and ME degrees from the Dalian University of Technology, China, in 2008 and 2011, all in computer science, and the PhD degree from the Australian National University, Australia, in 2016. He was a research associate at University College London, U.K. He currently is an associate professor in the School of Software, Dalian University of Technology, China. His research interests include cloud computing, software-defined networking, wireless sensor networks, algorithmic game theory, and optimization problems.

Yanbing Yang received the BE and ME degrees from the University of Electronic Science and Technology of China, China, and the PhD degree in computer science and engineering from Nanyang Technological University, Singapore. He is currently an associate research professor at the College of Computer Science, Sichuan University, China. His research interests include IoTs, visible light communication, and visible light sensing, as well as their applications.

Minghui Wang is currently a professor at the College of Computer Science, Sichuan University, Chengdu, China. His current research interests include information fusion, image processing, intelligent transportation systems, and multiple targets tracking.